Network Working Group
Internet-Draft
Updates: 3261 (if approved)
Expires: October 2, 2006

R. Sparks, Ed.
Estacado Systems
S. Lawrence
Pingtel Corp.
A. Hawrylyshen
Ditech Communications Corp.

March 31, 2006

Addressing an Amplification Vulnerability in Forking Proxies
draft-ietf-sip-fork-loop-fix-01

Abstract

This document normatively updates RFC 3261, the Session Initiation Protocol (SIP), to address a security vulnerability identified in SIP proxy behavior. This vulnerability enables an attack against SIP networks where a small number of legitimate, even authorized, SIP requests can stimulate massive amounts of proxy-to-proxy traffic.

This document strengthens loop-detection requirements on SIP proxies when they fork requests (that is, forward a request to more than one destination).

Table of Contents

1. Conventions and Definitions 3
2. Introduction 3
3. Vulnerability: Leveraging Forking to Flood a Network 3
4. Normative changes to RFC 3261 5
5. Impact on overall network performance 6
6. IANA Considerations 6
7. Security Considerations 6
8. Acknowledgements 6
9. References 6
9.1. Normative References 6
9.2. Informative References 7
Authors' Addresses 8
Intellectual Property and Copyright Statements 9

1. Conventions and Definitions

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].

2. Introduction

Interoperability testing uncovered a vulnerability in the behavior of forking SIP proxies as defined in [RFC3261]. This vulnerability can be leveraged to cause a small number of valid SIP requests to generate an extremely large number of proxy-to-proxy messages. A version of this attack demonstrates fewer than ten messages stimulating potentially 2^70 messages.

This document specifies normative changes to the SIP protocol to address this vulnerability. According to this update, when a SIP proxy forks a request to more than one destination, it is required to ensure it is not participating in a request loop.

3. Vulnerability: Leveraging Forking to Flood a Network

This section describes setting up an attack with a simplifying assumption, that two accounts on each of two different RFC 3261 compliant proxy/registrar servers that do not perform loop-detection are available to an attacker. This assumption is not necessary for the attack, but makes representing the scenario simpler. The same attack can be realized with a single account on a single server.

Consider two proxy/registrar services, P1 and P2, and four Addresses of Record, a@P1, b@P1, a@P2, and b@P2. Using normal REGISTER requests, establish bindings to these AoRs as follows (non-essential details elided):

```
           REGISTER sip:P1 SIP/2.0
           To: <sip:a@P1>
           Contact: <sip:a@P2>, <sip:b@P2>

           REGISTER sip:P1 SIP/2.0
           To: <sip:b@P1>
           Contact: <sip:a@P2>, <sip:b@P2>

           REGISTER sip:P2 SIP/2.0
           To: <sip:a@P2>
           Contact: <sip:a@P1>, <sip:b@P1>

           REGISTER sip:P2 SIP/2.0
           To: <sip:b@P2>
           Contact: <sip:a@P1>, <sip:b@P1>
```

With these bindings in place, introduce an INVITE to any of the four AoRs, say a@P1. This request will fork to two requests handled by P2, which will fork to four requests handled by P1, which will fork to eight messages handled by P2, and so on:

```
                                          |
                                        a@P1
                                      /       \
                                    /           \
                                  /               \
                                /                   \
                             a@P2                   b@P2
                             /  \                   /  \
                           /      \               /      \
                          /        \             /        \
                        a@P1       b@P1        a@P1       b@P1
                        /  \       /  \        /  \       /  \
                     a@P2  b@P2 a@P2  b@P2  a@P2  b@P2 a@P2  b@P2
                      /\    /\   /\    /\    /\    /\   /\    /\
                                          .
                                          .
                                          .
```

Figure 2

Requests will continue to propagate down this tree until Max-Forwards reaches zero. If the endpoint and two proxies involved follow RFC 3261 recommendations, the tree will be 70 rows deep, representing 2^70 requests. The actual number of messages may be much larger if the time to process the entire tree worth of requests is longer than Timer C at either proxy. In this case, a storm of 408s, and/or a storm of CANCELs will also be propagating through the tree along with

the INVITEs. Remember that there are only two proxies involved in this scenario - each having to hold the state for all the transactions it sees (at least 2^69 simultaneously active transactions near the end of the scenario).

The attack can be simplified to one account at one server if the service can be convinced that contacts with varying attributes (parameters, schemes, embedded headers) are sufficiently distinct, and these parameters are not used as part of AOR comparisons when forwarding a new request. Perhaps:

```
REGISTER sip:P1 SIP/2.0
To: <sip:a@P1>
Contact: <sip:a@P1;unknown-param=whack>,<sip:a@P1;unknown-param=thud>
```

This attack was realized in practice during one of the SIP Interoperability Test (SIPit) sessions. The scenario was extended to include more than two proxies, and the participating proxies all limited Max-Forwards to be no larger than 20. After a handful of messages to construct the attack, the participating proxies began bombarding each other. Extrapolating from the several hours the experiment was allowed to run, the scenario would have completed in just under 10 days. Had the proxies used the RFC 3261 recommended Max-Forwards value of 70, and assuming they performed linearly as the state they held increases, it would have taken 3 trillion years to complete the processing of the single INVITE that initiated the attack. It is interesting to note that a few proxies rebooted during the scenario, and rejoined in the attack when they restarted (as long as they maintained registration state across reboots). This points out that if this attack were launched on the Internet at large, it might require coordination among all the affected elements to stop it.

4. Normative changes to RFC 3261

The following requirements mitigate the risk of a proxy falling victim to the attack described in this document.

When a SIP proxy forks a particular request to more than one destination, it MUST ensure that request is not looping through this proxy. It is RECOMMENDED that proxies meet this requirement by performing the Loop-Detection steps defined as an optional step in Section 16.3 of RFC 3261.

The requirement to use the loop-detection algorithm in RFC 3261 is set at should-strength since it is expected that other mechanisms

that will allow a proxy to determine it is not looping will be standardized in the near future. For example, a proxy forking to destinations established using the sip-outbound mechanism [I-D.ietf-sip-outbound] would know those branches will not loop.

A SIP proxy forwarding a request to only one location MAY perform loop detection but is not required to. When forwarding to only one location, the amplification risk being exploited is not present, and the Max-Forwards mechanism is sufficient to protect the network. A proxy is not required to perform loop detection when forwarding a request to a single location even if it previously forked that request in its progression through the network.

5. Impact on overall network performance

These requirements and the recommendation to use the loop-detection mechanisms from RFC 3261 make the favorable trade of exponential message growth for work that is at worst case order n^2 as a message crosses n proxies. Specifically, this work is order m*n where m is the number of proxies in the path that fork the request to more than one location. In practice, m is expected to be small.

6. IANA Considerations

None.

7. Security Considerations

This document is entirely about addressing a vulnerability in SIP proxies as defined by RFC 3261 that can lead to an exponentially growing message exchange attack.

8. Acknowledgements

Thanks go to the implementors that subjected their code to this scenario and helped analyze the results at SIPit 17.

9. References

9.1. Normative References

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

[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.

9.2. Informative References

[I-D.ietf-sip-outbound]
Jennings, C. and R. Mahy, "Managing Client Initiated Connections in the Session Initiation Protocol (SIP)", draft-ietf-sip-outbound-03 (work in progress), March 2006.

Authors' Addresses

Robert Sparks (editor)
Estacado Systems
17210 Campbell Road
Suite 250
Dallas, Texas 75254-4203
USA

Email: RjS@nostrum.com

Scott Lawrence
Pingtel Corp.
400 West Cummings Park
Suite 2200
Woburn, MA 01801
USA

Phone: +1 781 938 5306
Email: slawrence@pingtel.com

Alan Hawrylyshen
Ditech Communications Corp.
602 - 11 Ave SW
Suite 310
Calgary, Alberta T2R 1J8
Canada

Phone: +1 403 561 7313
Email: ahawrylyshen@ditechcom.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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