

Network Working Group  
Internet-Draft  
Updates: [3261](#) (if approved)  
Expires: August 28, 2006

R. Sparks, Ed.  
Estacado Systems  
S. Lawrence  
Pingtel Corp.  
A. Hawrylyshen  
Ditech Communications Corp.  
February 24, 2006

Addressing an Amplification Vulnerability in Forking Proxies  
draft-ietf-sip-fork-loop-fix-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 28, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

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.

Internet-Draft

fork-loop-fix

February 2006

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

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

---

Internet-Draft

fork-loop-fix

February 2006

## 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):

Internet-Draft

fork-loop-fix

February 2006

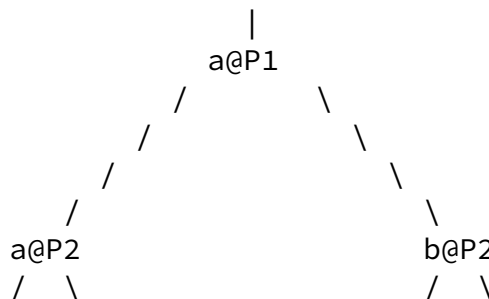
```
REGISTER sip:P1 SIP/2.0
To: <sip:a@P1>
Contact: <sip:a@P2>, <sip:b@P2>
```

```
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:



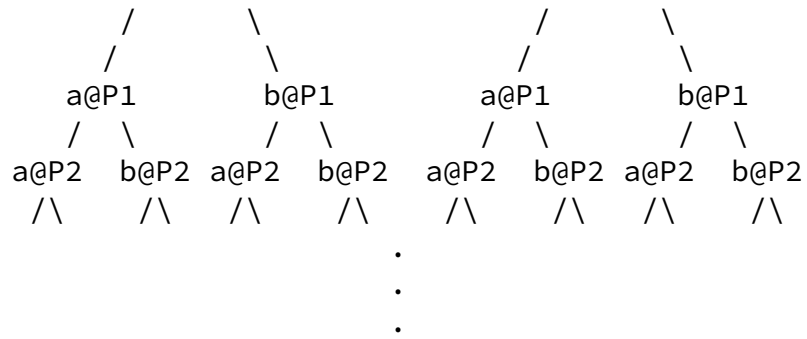


Figure 2

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 scenerio 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 in the wild, 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](#). Other loop-detection mechanisms, such as an adaptation of Knuth's algorithm, or the technique described in [[I-D.campen-sipping-stack-loop-detect](#)] MAY be used.

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. Optimizations such as that described in [[I-D.campen-sipping-stack-loop-detect](#)] could reduce this to work that is linear with  $m$  and independent of  $n$ .

## [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 thier code to this scenario and helped analyze the results at SIPIT 17.

Dave Oran brought the idea of using a Knuth's-algorithm-like alternative to the loop detection mechanisms in [RFC 3261](#) to the SIP working group's attention many times in the development of the SIP protocol.

## [9.](#) References

Sparks, et al.

Expires August 28, 2006

[Page 6]

---

Internet-Draft

fork-loop-fix

February 2006

### [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.campen-sipping-stack-loop-detect]

Campen, B., "An Efficient Loop Detection Algorithm for SIP Proxies", [draft-campen-sipping-stack-loop-detect-00](#) (work in progress), February 2006.

Sparks, et al.

Expires August 28, 2006

[Page 7]

---

Internet-Draft

fork-loop-fix

February 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

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

## Acknowledgment

Funding for the RFC Editor function is currently provided by the

Internet Society.

Sparks, et al.

Expires August 28, 2006

[Page 9]