Testing Dialog-Verification of SIP Phones with Single-Message Denial-of-Service Attacks

Jan Seedorf, Kristian Beckers, Felipe Huici

NEC Laboratories Europe
Kurfuerstenanlage 36, 69115 Heidelberg
{firstname.lastname}@nw.neclab.eu

Abstract. The Session Initiation Protocol (SIP) is widely used for signaling in multimedia communications. However, many SIP implementations are still in their infancy and vulnerable to malicious messages. We investigate flaws in the SIP implementations of eight phones, showing that the deficient verification of *SIP dialogs* further aggravates the problem by making it easier for attacks to succeed. Our results show that the majority of the phones we tested are susceptible to these attacks.

1 Introduction

The Session Initiation Protocol (SIP) [2] is a protocol for setting up, managing, and tearing down multimedia sessions. Much of its success and popularity are due to its use in Voice-over-IP (VoIP) devices. VoIP is a rapidly growing market with high potential for the future, and so competition is fierce. This leads to fast and immature development of SIP products; indeed, it is no secret that quite a few SIP implementations are vulnerable to malformed messages [1]. In this paper we investigate the robustness of SIP phones against two specific, single-message Denial-of-Service (DoS) attacks: *Cancel* and *Bye* attacks.

2 Description of Attacks

To establish a VoIP session with SIP, user *Alice* first sends an INVITE request to its proxy, which forwards it to the proxy of *Bob*'s domain (Figure 1). This proxy knows his current location (IP address) and can forward him the INVITE request. The proxies and *Bob*'s SIP user agent signal back to *Alice*'s user agent that the call is being established (`100-trying/180-ringing`). At this point an attacker can carry out a **Cancel attack** by sending a CANCEL request to *Bob* before the session is completely set up, resulting in DoS (step 6 in the figure). A normal connection would continue by *Bob* sending a 200 OK message (steps 10-12) and *Alice* replying with an ACK message (not shown in the figure for simplicity). After the connection is established, an attacker can perform a **Bye attack**, in which he sends a BYE message to *Bob*, prematurely ending his conversation with *Alice* (step 13).

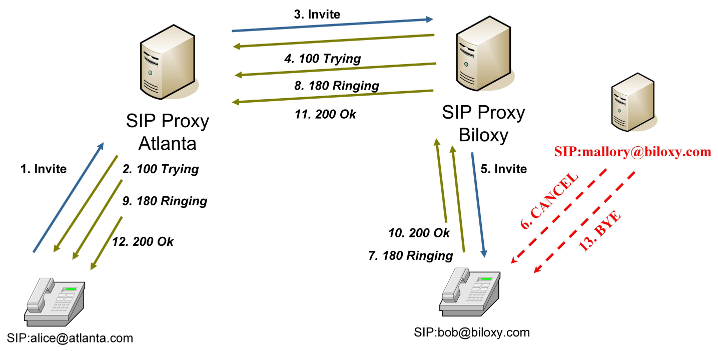

Fig. 1. SIP call establishment showing *Cancel* and *Bye* attacks.

In general, SIP messages are of different types (e.g., INVITE, CANCEL, BYE), contain various header fields and a body, and are sent either as requests or responses. To distinguish between different sessions, SIP uses so-called *dialog identifiers* [2]. A SIP *dialog ID* is composed of the `Call-ID`, the `From-tag` (contained in the From-header) and the `To-tag` (contained in the To-header). Thus, any SIP entity can verify that a message belongs to a dialog (and therefore in principle prevent *Cancel* and *Bye* attacks) by checking that:

1. The Call-ID is the same as that of previous messages within the dialog
2. The tag in the From header matches that of previous messages within the dialog
3. The tag in the To header matches that of previous messages within the dialog

Figure 2 shows the establishment of a SIP session including the dialog-ID components sent between the entities. The `Call-ID` and the `From-tag` are set by the caller and the `To-tag` by the callee. These tags remain in every SIP message throughout the dialog, and if a message does not match an existing dialog it should be discarded. While tags are optional, if a message contains one or both these tags, all messages in the dialog must also contain them. The figure also shows how attacker *Mallory* could, in principle, carry out *Cancel* and *Bye* attacks. However, in order to do so, she would have to know several components of the corresponding dialog in order to succeed. We will show that with quite a few SIP implementations such attacks can be carried out with less knowledge.

3 Testing SIP Dialog Verification

We were interested in the robustness of currently-available User Agent SIP implementations against forged dialog-IDs. We consider a dialog-ID to be forged if one component that comprises the dialog-ID differs from the component originally chosen by either the caller or the callee. For our testing we sent both CANCEL and BYE requests, forging either the `Call-id`, the `From-tag`, or the `To-tag`. Note that even though a CANCEL can be sent without a `To-tag` at all, a CANCEL with an unknown `To-tag` should be ignored. Our goal was to

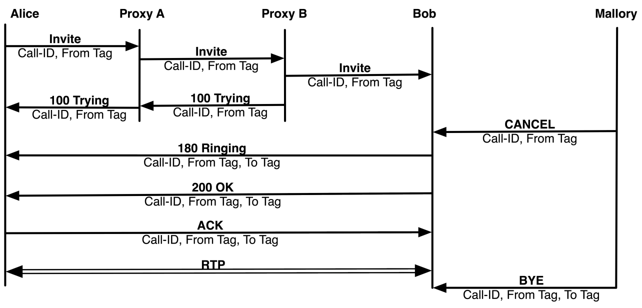

Fig. 2. Detailed view of SIP *Cancel* and *Bye* attacks.

see in which of these cases we would be successful in carrying out a *Cancel* or *Bye* attack. To do so we needed a testing tool that was both *stateful* and *reactive*: the former in order to trigger a forged BYE or CANCEL message after previous messages had been exchanged; and the latter to parse the `To-tag` in SIP responses from the user agent under test (this was needed to execute tests where the `To-tag` differs slightly from the one in the current dialog). Most SIP testing tools are neither stateful nor reactive (e.g., Protos [4]). Therefore, we implemented the test cases ourselves using SIPp [3] as a message generator and parser. We tested four SIP hardphones and four SIP softphones against messages with forged SIP dialog-IDs. Table 1 summarises the results. The right-most column shows which phones accepted a particular forged request, with the labels representing anonymised phone names (softphones: S1-S4; hardphones: H1-H4). The hardphones were tested *out-of-the-box* as well as after a firmware update. As none of the phones we tested was susceptible against an attack with a forged `Call-id`, these results are left out in the table. Our results demonstrate that a majority of SIP softphones is vulnerable to these kind of attacks as well as two hardphones with their unpatched, out-of-the box firmware. Only three SIP phones we tested (S3, H2, H4) ignored messages in all our test-cases with forged dialog-components.

Request	**Forged Header Field**	**Vulnerable Phones**
Cancel	From Tag	S1, S2, S4, H1*, H3*
	To Tag	S1, S2, S4, H1*, H3*
Bye	From Tag	S1, S2, S4, H1*, H3*
	To Tag	S1, S2, S4, H1*, H3*

Table 1. Results of vulnerabilities to forged SIP header field attacks for four softphones (S1-S4) and four hardphones (H1-H4). Starred entries denote vulnerabilities that were only found with the unpatched firmware version of the particular phone.

4 Discussion of Results

In the previous section we presented results showing several vulnerabilities in the eight phones tested. The goal of the attacker is to cause DoS either by preventing call establishment with a *Cancel* attack or by terminating an existing call with a *Bye* attack; we will now discuss the extent to which these flaws enable the attacker to be successful. We assume that *Mallory* is able to sniff the outgoing INVITE message, but not the reply message from *Bob* containing the To-tag; this would be the case if *Mallory* could see messages going to *Alice*'s proxy, but not those going directly from *Bob* to *Alice*[1]. In this scenario, *Mallory* could very easily deny communication by sending CANCEL or BYE messages with the values obtained from the sniffed INVITE message and any value for the `To-tag`, an attack that would succeed on five of the phones we tested. While admittedly this attack puts constraints on the scenarios in which it would be successful, it is nonetheless of concern. More importantly though, these results shed some light on the weakness of current SIP implementations, and hint at perhaps even more serious flaws to be discovered. Worse, despite manufacturers releasing fixes, not all phones are kept up to date but instead are used continuously with unpatched firmware versions. Our results show that while hardphones seem better protected against *Cancel* or *Bye* attacks than softphones, two of the hardphones we tested were not protected against these attacks with an unpatched firmware version. Motivated by these results, we intend to further investigate similar attacks with other forged SIP headers. For instance, at present we are working on more sophisticated test-cases to help in finding related flaws, e.g., with forged transaction components like the `via-branch`.

5 Conclusion

We tested several SIP implementations against simple yet effective DoS *Cancel* and *Bye* attacks using forged dialog IDs. Our results are worrying and show that a majority of the softphones we tested as well as two hardphones with unpatched firmware are vulnerable, allowing attackers to prevent or prematurely end VoIP sessions. The results further illustrate the weakness of current SIP implementations, and we are continuing to investigate these and related vulnerabilities. As part of these efforts, we are currently developing a tool to enable discovery of perhaps even more serious flaws quickly.

Acknowledgement: The authors would like to thank Stephan Sutardi for his help in implementing test-cases and testing phones.

References

1. CERT Advisory CA-2003-06, *Multiple vulnerabilities in implementations of the Session Initiation Protocol (SIP)*, http://www.cert.org/advisories/CA-2003-06.html
2. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E., *SIP: Session Initiation Protocol*, RFC 3261, 2002
3. SIPp, *Welcome to sipp*, http://sipp.sourceforge.net/
4. Wieser, C., Laakso, M., Schulzrinne, H., *SIP Robustness Testing for Large-Scale Use*, 1st Int. Workshop on Software Quality (SOQUA 2004), Erfurt, Germany, 2004

[1] SIP as specified in [2] allows the callee to send the *180-ringing* and *200-ok* directly to the caller and not via the proxies (unlike the message flow shown in figure 2).