Abstract

This document examines the proposal for a SIP extension to allow authentication of a user-agent through a proxy to a RADIUS server using CHAP put forward by Byerly and Williams [BYERLY]. An alternative method is offered which does not require any changes to the SIP protocol as defined in [SIP]. It also will not require any modification to the way RADIUS clients communicate with RADIUS Servers [RADIUS], although it will change the way some of the fields are interpreted by the RADIUS server.

1. Introduction

One widely implemented platform for authenticating users and authorizing them for selected services is the RADIUS protocol [RADIUS]. RADIUS allows the secure transfer of information between a Radius Client (often called the NAS - Network Access Server) and a Radius Server (which has access to the user’s account...
information typically stored in a separate database). RADIUS does this and guarantees security of: (1) the Radius Client’s authenticity, (2) the RADIUS Server’s authenticity, (3) confidentiality of the user’s password, and (4) integrity of any other account information passed between the Radius Server and Radius Client. This is accomplished by an out-of-band negotiation between the Radius Server and Radius Client of a "shared secret" that should be known only to the two entities.

SIP currently supports two authentication methods, Basic and Digest (section 3.2.2.2 of [SIP]). In Basic Authentication the user password is passed from the user agent to the proxy in clear text, and is therefore undesirable. Byerly and Williams [BYERLY] pointed out that if Digest Authentication is used, a mapping problem exists between the hashing algorithms used in Digest Authentication and that used between the Radius Client and Server. They see two possibilities for dealing with this, "One is to extend RADIUS to support HTTP-Digest; the other is to extend the SIP list of authentication schemes to support a CHAP-Password." (Section 1 of [Byerly]) They propose an extension to SIP in order to make use of CHAP.

This draft will examine the possibility of the former alternative and argue that extending RADIUS to deal with Digest Authentication is more desirable in certain circumstances. This scheme can also be more easily implemented since it requires some minor modifications only to the RADIUS Server (of which there are very few), as opposed to altering the end-user SIP stacks (of which there are very many).

2. Definitions

Nonce
A uniquely generated number or string that is generated each time a request is made. In RADIUS, this is called the Request Authenticator.

Sequence-number
An integer that is incremented by one for each request in order to facilitate transaction handling and allow detection of duplicate transmissions.

Shared-secret
A string that is known only to the RADIUS Server and RADIUS Client. The negotiation of this string between the two is outside the scope of the RADIUS protocol.
3. Review of Authentication Algorithms

A short review of the algorithms used by the authentication methods is presented here for clarity.

3.1 RADIUS Authorization

The RADIUS Server sends an Authorization Request to the Server with the relevant fields Request Authenticator (RA) and User-Password. RA is a 16 octet random number decided on by the RADIUS Client. The User-Password is calculated as

\[ \text{User-Password} = \text{MD5Hash(Shared Secret, RA)} \text{ XOR password} \]

Since the RADIUS Server knows the Shared Secret and receives RA, it can reconstruct the user’s password and test it against the one it has stored in its database.

3.2 Digest Authentication

There are slight variations in the string that is hashed using this method depending on the security level required (i.e. just authorization, requiring integrity of the message as well, whether a session key is to be established). The following algorithm should be suitable for our purposes. The authentication value passed back by the user agent is calculated as:

\[ \text{MD5Hash(H(A1), nonce-value, H(A2))} \]

Where

\[ A1 = \text{User-Name} + \text{":"} + \text{realm value} + \text{":"} + \text{password} \]

A2 contains the modified URI values as detailed in section 14.3 of [SIP] and may also add a signature to assure integrity.

3.3 CHAP Authentication

In CHAP, the RADIUS Server issues a challenge to the RADIUS Client, and typically passes back a nonce value. The server expects to receive back a response from the client of the form

\[ \text{MD5Hash(seqnum, password, nonce)} \]
Byerly and Williams have proposed that the RADIUS Client pass this nonce on to the user-agent and have the user-agent calculate the hash value to be returned (via the proxy) to the RADIUS Server for authentication.

4. Problems and limitations of using CHAP

A number of limitations and problems arise when considering implementing CHAP as the authentication method in SIP/RADIUS.

1. The requirement for user-agents to implement MD5 hashing increases the processing power needed by the UA and also increases the memory footprint and complexity of the SIP stack. This may be critical if SIP is implemented on "dumb" hardware and embedded systems where external constraints may necessitate considerable hardware and memory limitations.

2. Adding extensions to SIP effects many end-user agents and will not allow current implementations to work with proxies authenticating with RADIUS.

3. CHAP has inherent security problems in that it requires user passwords on the Server database to be stored in clear text. Some service providers will refuse to take the responsibility for maintaining their passwords in such an insecure state.

4. The method proposed by Byerly will not work with non-standard implementations of RADIUS such as Ericsson IPTS. In that implementation, the proxy (RADIUS Client) must have the password available to it in order to encrypt it according to another (proprietary) scheme.

5. CHAP verifies that the user has the correct password and in this way authenticity of the user is guaranteed, but it does not give the user any information relating to the proxy, (i.e. the proxy can be spoofed and the user ends up giving its password to it). In other words, it does not provide for authenticity of the proxy.

6. CHAP can not be extended to non-password fields. If the header messages or body require confidentiality, then CHAP is not sufficient.
5. Alternative Proposals

5.1 Best Solution from a Security Standpoint

The solution which best solves the above security related problems is to initiate a TLS session between the user-agent and the proxy ([TLS]). All subsequent SIP messages can be passed in a secure environment. This solution provides for authenticity of the proxy. If authenticity of the client is also required (to a greater level than just knowing the user password), it can be provided through a client certificate. Scalability issues regarding this scheme need to be investigated further.

5.2 Next-Best Solution

TLS is the best solution in terms of security, but it requires overhead and complexity, necessitates changes to the user agent, and is not backwards compatible to existing SIP user-agent implementations.

Another option is to allow user-agents to authenticate themselves with the proxy using Digest Authentication. In order for the RADIUS server to utilize this information a small modification to the server is necessary. RADIUS allows for the passing of any information through the use of attributes, and specifically, Vendor Specific Attributes (Section 5 and 5.26 of [RADIUS]). This information is passed in clear text between the RADIUS Client and Server and can be implemented as the vendor sees fit. The user-name, realm, nonce, and any other information necessary to reconstruct the Digest Authentication hash value - except the password, of course - can be sent from the proxy to the RADIUS Server as Vendor Specific Attributes. The RADIUS Server can then retrieve the last unknown from its database - the password - and compute the hash value to be compared. The Digest Authentication value can either be passed as the RADIUS user-password, or separately as a Vendor Specific Attribute.

This scheme will allow backwards-compatibility with existing user-agents that can implement Digest Authentication. It does not increase the complexity of the UA SIP stack. Furthermore, since the Digest Authentication algorithm hashes the user-name, realm, and password first, and then hashes that result together with the ever-changing nonce, the RADIUS server can store the hash of the user-name, realm and password instead of the clear text password.
6. Acknowledgements

The author would like to thank Jonathan Rosenberg for his comments and insights in discussions relating to the topic presented in this draft.

7. References


Author’s Address

Baruch Sterman
Deltathree
75 Broad St.
New York, NY 10004
USA
Email: baruch@deltathree.com
8. Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.