Internet Engineering Task Force (IETF)
Request for Comments: 5803
Category: Informational
ISSN: 2070-1721

A. Melnikov
Isode Limited

July 2010

Lightweight Directory Access Protocol (LDAP) Schema for Storing Salted Challenge Response Authentication Mechanism (SCRAM) Secrets

Abstract

This memo describes how the "authPassword" Lightweight Directory Access Protocol (LDAP) attribute can be used for storing secrets used by the Salted Challenge Response Authentication Message (SCRAM) mechanism in the Simple Authentication and Security Layer (SASL) framework.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5803.

Table of Contents

1. Overview

This document describes how the authPassword LDAP attribute [AUTHPASS] can be used for storing secrets used by [SCRAM] Simple Authentication and Security Layer [RFC4422] Mechanisms.

The "scheme" part of the authPassword attribute is the SCRAM mechanism name (always without the "-PLUS" suffix), e.g., "SCRAM-SHA-1". See [SCRAM] for the exact syntax of SCRAM mechanism names.

The "authInfo" part of the authPassword attribute is the iteration count (iter-count in the ABNF below), followed by ":" and base64-encoded [BASE64] salt.

The "authValue" part of the authPassword attribute is the base64-encoded [BASE64] StoredKey [SCRAM], followed by ":" and base64-encoded [BASE64] ServerKey [SCRAM].

Syntax of the attribute can be expressed using ABNF [RFC5234]. Non-terminal references in the following ABNF are defined in either [AUTHPASS], [RFC4422], or [RFC5234].

```
scram-mech     = "SCRAM-SHA-1" / scram-mech-ext
                 ; Complies with ABNF for <scheme>
                 ; defined in [AUTHPASS].

scram-authInfo = iter-count ":" salt
                 ; Complies with ABNF for <authInfo>
                 ; defined in [AUTHPASS].

scram-authValue = stored-key ":" server-key
                 ; Complies with ABNF for <authValue>
                 ; defined in [AUTHPASS].

iter-count   = %x31-39 *DIGIT
                 ; SCRAM iteration count.
                 ; A positive number without leading zeros.

salt         = <base64-encoded value>
```

```
stored-key   = <base64-encoded value>
               ; See definition in [SCRAM].

server-key   = <base64-encoded value>
               ; See definition in [SCRAM].

scram-mech-ext = "SCRAM-" 1*9mech-char
               ; Other SCRAM mechanisms registered
               ; in the IANA registry for SASL
               ; mechanism names.

mech-char    = <Defined in RFC 4422>
```

Note that the authPassword attribute is multivalued. For example, it may contain multiple SCRAM hashes for different hashing algorithms.

2. Conventions Used in This Document

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 [RFC2119].

3. Security Considerations

This document defines how the authPassword attribute can be used to store SCRAM secrets. Therefore, security considerations relevant to [SCRAM] and hash functions used with it are also relevant to this document.

General security considerations related to the authPassword attribute (as specified in [AUTHPASS]) also apply to the use of authPassword as specified in this document. In particular, the values of authPassword SHOULD be protected as if they were cleartext passwords. A read operation on this attribute that is not protected by a privacy layer (such as IPsec or TLS) can expose this attribute to an attacker who a) would be able to use the intercepted value to impersonate the user to all servers providing SCRAM access using the same hash function, password, iteration count, and salt or b) would be able to perform an offline dictionary or brute-force attack in order to recover the user’s password.

Servers MUST validate the format of the authPassword attribute before using it for performing a SCRAM authentication exchange. It is possible that an attacker compromised the LDAP server or got access to the entry containing the attribute in order to exploit a vulnerability in the subsystem performing the SCRAM authentication

exchange. Big iteration counts and invalid base64 encoding are two possible (but not the only) exploits in the format specified in the document.

4. Acknowledgements

The author gratefully acknowledges the feedback provided by Chris Newman, Kurt Zeilenga, Chris Lonvick, Peter Saint-Andre, Barry Leiba, and Chris Ridd.

5. Normative References

[AUTHPASS] Zeilenga, K., "LDAP Authentication Password Schema", RFC 3112, May 2001.

[BASE64] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

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

[RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[SCRAM] Menon-Sen, A., Newman, C., Melnikov, A., and N. Williams, "Salted Challenge Response Authentication Message (SCRAM) SASL Mechanisms", RFC 5802, July 2010.

Author’s Address

Alexey Melnikov
Isode Limited
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

EMail: alexey.melnikov@isode.com
URI: http://www.melnikov.ca/