SIPCORE Working Group
Internet-Draft
Intended Status: Standards Track
Expires: April 13, 2015

R. Shekh-Yusef
Avaya

October 10, 2014

Key-Derivation Authentication Scheme
draft-yusef-sipcore-key-derivation-00

Abstract

This document defines a Key-Derivation Authentication Scheme, based on the PBKDF2 Key Derivation Function (KDF), that could be used with the challenge-response authentication framework used by SIP to authenticate the user.

The scheme allows two parties to establish a mutually authenticated communication channel based on a shared password, without ever sending the password on the wire.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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/1id-abstracts.html

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

Copyright and License Notice

Table of Contents

1 Introduction

SIP [RFC3261] uses the Digest Authentication schemes with the general framework for access control and authentication, which is used by a server to challenge a client request and by a client to provide authentication information.

The challenge-response framework relies on passwords chosen by users which usually have low entropy and weak randomness, and as a result cannot be used as cryptographic keys.

While cannot be used directly as cryptographic keys, the passwords can still be used to derive cryptographic keys, by using Key Derivation Function (KDF).

This document defines a new scheme, Key-Derivation Authentication Scheme, to replace the Digest scheme, that could be used with the challenge-response authentication framework used by SIP to authenticate the user.

The Key-Derivation scheme ensures that the password is never sent on the wire, and allows for a better secure storage of passwords, as it significantly increases the amount of computation needed to derive a key from a password in a dictionary attack.

The Key-Derivation scheme creates a master-key, that is derived from the password, which has a much better entropy than the password, to calculate a proof-of-possession for the shared password.

1.1 Terminology

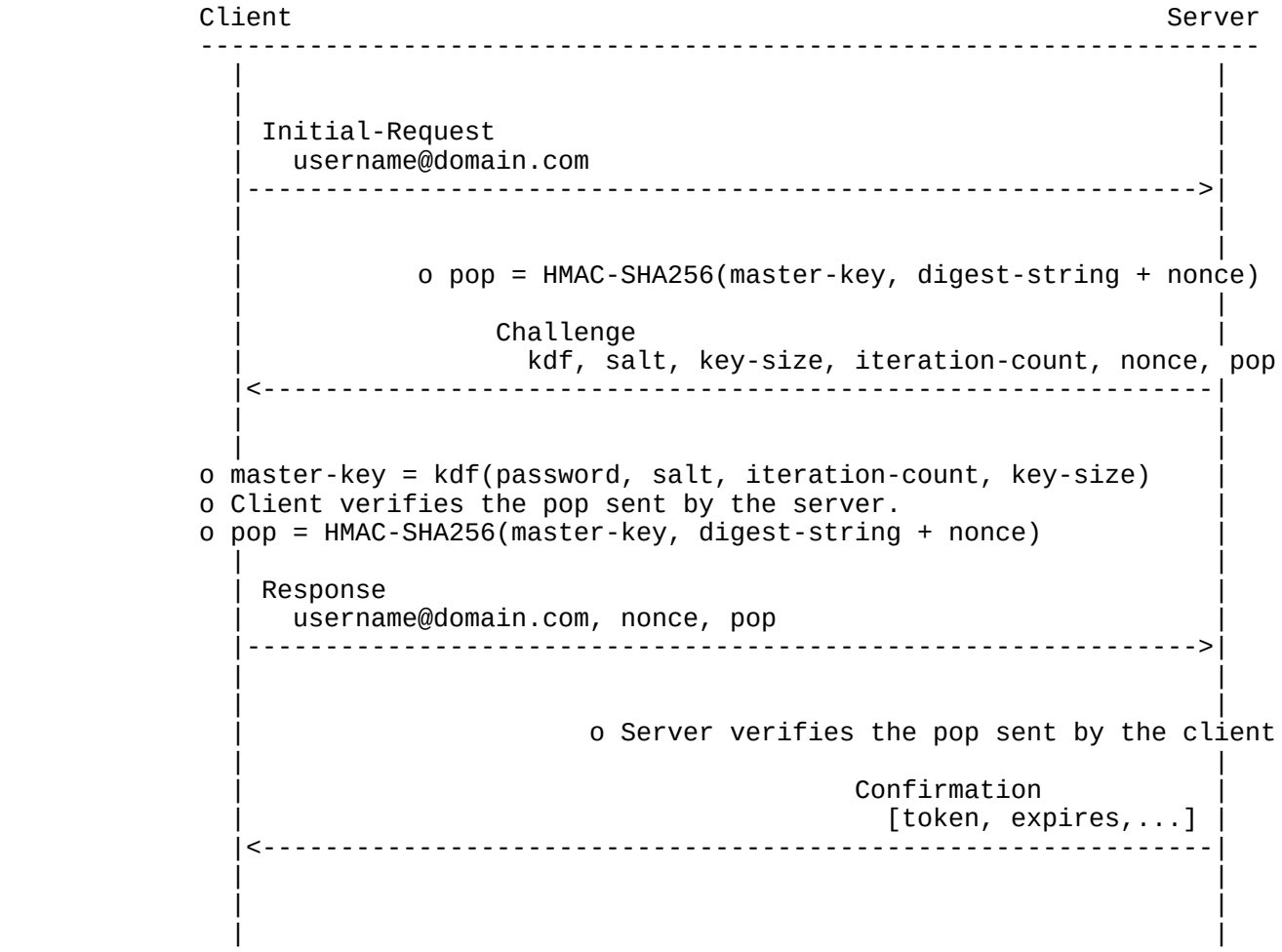
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2 Operations Overview

When an account is created, the server uses a KDF, a salt, a key length, and an iteration count to create a master-key based on the user’s password, as defined in [NIST-KD]. The server then stores the following information in the database:

- username
- iteration count
- salt
- master-key

The following flow describes at a high-level the flow of messages based on the challenge-response framework:



With the challenge-response framework the initial request from the client is sent without providing any credentials.

When the server receives the initial request from the client, the server fetches the master-key associated with the username provided in the request. The server then uses the master-key to create a proof-of-possession (pop) using an HMAC-Hash function with the digest-string and nonce from the challenge.

The digest-string, as defined in Section 9 of [RFC4474], is a list of SIP headers that must be hashed to create the proof-of-possession defined in this document.

The server then challenges the request and includes the Key-Derivation scheme with a kdf, a salt, a key-size, an iteration-count, a nonce, and pop.

To be able to provide credentials to the server the client must create the master-key as was done by the server when the account was initially created as described above using the parameters provided by the server in the challenge. The client will then verify the pop sent by the server using its master-key, the digest-string of the incoming request, and the nonce provided in the challenge.

The client then creates a proof-of-possession (pop) using an HMAC-Hash function and the master-key using the digest-string from the response concatenated with the nonce to be sent to the server. A valid response from the client will contain the Key-Derivation scheme, a nonce, and the pop parameter.

When the server receives the response, it verifies the pop, and if that is valid, it sends a confirmation.

At the end of the above process, the client and the server would have established a communication channel after completing a mutual authentication using the same master-key on both sides.

Subsequent requests will be able to use the master-key to create pop to prove possession of the credentials.

3 The Challenge

When a server receives a request from a client, and an acceptable authorization is not sent, the server challenges the originator to provide credentials by rejecting the request and include the Key-Derivation scheme.

The challenge should include the following parameters:

KDF (REQUIRED)

A deterministic algorithm used to derive cryptographic keys from a shared secret like a password. A good example of such a function is HMAC-SHA2-256.

Iterations (OPTIONAL)
The number of iterations that the KDF will be applied on the salt and password. The default value for this parameter is 1000.

Salt (REQUIRED)

A random value that is used to make sure that the same password will always be hashed differently. The salt MUST be generated using an approved Random Number Generator.

Key-Size (REQUIRED)
The size of the derived key in bits.

nonce (REQUIRED)
A server-specified value that should be uniquely generated each time a challenge is made.

pop (REQUIRED)

The pop is derived from applying the HMAC-SHA256 on digest-string and a nonce using the master-key, as follows:

```
pop = HMAC-SHA256(master-key, digest-string + nonce)
```

4 The Response

The client first creates the master-key based on the parameters provided by the server in the challenge.

The client then uses the master-key to verify the pop sent by the server; if that is successful, the client then uses the master-key to create a pop for the response to be sent to the server.

The client is expected to retry the request, passing the nonce and pop with the Key-Derivation scheme.

nonce (REQUIRED)
A client-specified value that should be uniquely generated each time a response is made.

pop (REQUIRED)

The pop is derived from applying the HMAC-SHA256 on digest-string and a nonce using the master-key, as follows:

pop = HMAC-SHA256(master-key, digest-string + nonce)

5 The Confirmation

The server verifies the proof-of-possession sent by the client. If the verification is successful, the server sends a confirmation to the client; otherwise, the server declines the request.

6 Security Considerations

<Security considerations text>

7 IANA Considerations

<IANA considerations text>

8. Acknowledgments

<Acknowledgments text>

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.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[NIST-KD] "NIST Special Publication 800-132 - Recommendations for Password-Based Key Derivations", December 2010.

http://csrc.nist.gov/publications/nistpubs/800-132/nist-sp800-132.pdf

9.2 Informative References

Authors' Addresses

Rifaat Shekh-Yusef
Avaya
250 Sydney Street
Belleville, Ontario
Canada

Phone: +1-613-967-5267
Email: rifaat.ietf@gmail.com