HIP and Strong Password Authentication of Users
draft-varjonen-hip-srp-00

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.

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/1id-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 September 1, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

This document specifies how to use Secure Remote Password (SRP) protocol in conjunction with Host Identity Protocol (HIP). In order to conceive this conjunction this document specifies three new parameters to be used with HIP control packets. These parameters are used to transport values related to the SRP protocol. This document also specifies how peers should act when these SRP parameters are found from HIP control packets and how this affects middleboxes.

Requirements Language

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].

Table of Contents

1. Glossary ......................................................... 4
2. Introduction ................................................... 5
3. SRP Parameters .................................................. 7
   3.1. SRP_CV parameter ........................................... 7
   3.2. SRP_CV_M parameter ......................................... 8
   3.3. SRP_SV Parameter ........................................... 8
   3.4. SRP_SV_M Parameter ......................................... 9
   3.5. SRP_E Parameter ............................................. 10
   3.6. SRP_E_M Parameter .......................................... 10
4. Handling of SRP Parameters ................................. 10
5. SRP and middleboxes ........................................... 12
6. Extension to HIP Native API ............................... 13
7. Design Alternatives ........................................... 14
8. IANA Considerations ........................................... 15
9. Security Considerations ..................................... 15
10. Acknowledgements ............................................ 16
11. References .................................................. 16
   11.1. Normative References ...................................... 16
   11.2. Informative References .................................... 17
1. Glossary

Terminology and notation used in this document is in most parts borrowed from [RFC5054], [SRP-6] and from the original paper describing SRP ([Wu.SRP]).

USER: Is the entity that is being authenticated in the SRP protocol.

HOST: Denotes the hardware and is identified by its Host Identity.

CLIENT: Host in a role of a client that initiate the connection.

SERVER: Host in a role of a server that offers service that is protected with username-password tuple.

USERNAME, C: String that identifies the USER uniquely to the SERVER and is used in similar fashion as a public key (aka "identity").

PASSWORD, P: String that is user memorable and is used in a similar fashion as private key.

VERIFIER: Is computed from PASSWORD in such fashion that it is hard to derive the PASSWORD from the VERIFIER.

VERIFIER-BASED: SERVER stores verifiers instead of plain-text passwords.

ZERO-KNOWLEDGE PROOF: PASSWORD is not sent on wire between CLIENT and SERVER. SERVER computes VERIFIERs from the values given by the CLIENT.

N, g: group parameters (prime and generator), defined in [RFC5054] appendix A.

s: salt

B, b: server’s public and private values

A, a: client’s public and private values

v: verifier, calculated from P.

k: SRP-6 multiplier

K: Session key

M: evidence, calculated from public values and K.
The | symbol indicates string concatenation, the ^ operator is the exponentiation operation, and the % operator is the integer remainder operation. H() denotes hash function like SHA-1 (length 160 bits). Interleaved SHA is used in some parts of the protocol (Incorporates two SHA-1 hashes one for even bits and one for odd bits (length 320 bits)). Interleaved SHA is used to create the session key, see [RFC2945] for details.

Conversion between integers and byte-strings assumes that the resulting byte-string is in network byte order.

2. Introduction

Host Identity Protocol (HIP) [RFC4423] offers a cryptographic namespace and a way to negotiate a creation of Security Association (SA) between hosts. By default SAs are created by contacting the Responder. With HIP firewalls administrators can access control the connection attempts. In some cases access control based only on HITs is not enough. Organizations can have mobile employees that need access to the organizations network from outside the network. HIP can be used to authenticate the host but by default anyone possessing the machine and keys can create the tunnel. By introducing Secure Remote Password protocol (SRP, [Wu.SRP]) to HIP we can extend the authentication of the host to include the authentication of the user.

SRP [RFC2945] offers a strong zero-knowledge proof authentication of a user/client to a server without any need for trusted third party. The SRP protocol can be run in one-way or in two-way mode, in one-way mode only the client is authenticated and in two-way mode client and the server are authenticated. Along with the authentication, peers run key negotiation and after the authentication peers share a session key. Performance wise SRP is slower than Diffie-Hellman [RFC2631] but SRP’s performance has been proven as adequate [Performance.SRP]. SRP standardization [RFC2945] defines the SRP in an abstract way leaving most of the details to the hands of the implementors. This is one reason why this document borrows some parts from the other standardized solutions, like [RFC5054]

SRP needs some preliminary values stored on the server requiring the authentication. In SRP it is a triplet consisting of username, password verifier and salt. Upon initialization of SRP, user gives a username, a password and SRP implementation takes a random salt. Verifier is calculated from these values (v = g^(SHA(salt|SHA(username)|":"|plain password) % N) and stored on the server requiring the authentication. The prime (N) and generator (g) can be taken from Appendix A of [RFC5054]
SRP authentication consists of four messages (see Figure 1, in the first of the messages user tells his/her identity in the manner of supplying a username. The user also provides a public value that is calculated by raising the generator to the power of a random value a and taking the integer remainder of N from the result ($A = g^a \mod N$).

For the second message the server finds the salt for the user and send its own public value calculated by raising generator to the power of random b, adding it to the verifier of the user and taking the integer remainder of N from the result ($B = (v + g^b) \mod N$). By now the peers have enough information on for creation of a session key, for details see [RFC2945]. The third message is the evidence (M1), if verified, it tells if the user knew the password. This is calculated by using the values sent in earlier messages and by using the newly formed session key, in other words the peers are verifying that they know the same key. For the calculation of the evidence M see [RFC2945].

If the negotiation is left to the three messages only the client/Initiator is authenticated. If the server/responder needs to be authenticated, the server sends a fourth message containing its evidence (M2)

The authentication does not need to be run endpoint-to-endpoint. There are cases where the middlebox along the way would be more interested in authenticating the user, for example a firewall on the edge of a corporate network could want to authenticate the user along side the equipment (identified by HIT) before letting the ESP flow through (see Section 5).
While here, we talk of messages, in case of HIP these messages are converted to mean individual parameters transported in HIP control packets.

3. SRP Parameters

This section defines six parameters to be used when using SRP with HIP. These packets are SRP_CV, SRP_SV, SRP_E and their middlebox counterparts. The first parameter, named SRP_CV, is the initiating parameter for the SRP negotiation. It contains the username of the user to be authenticated and its public value. SRP_SV stands for server values which contains the salt and servers public value and SRP_E is a parameter to transport the evidence. Parameters ending with _M are identical to the ones that end without _M, only difference is the type number which tells the receiver that _M parameters are not protected by signatures. Middleboxes are the most probable users of _M parameters, with the exception of pre-creation of R1s discussed in section Section 4.

3.1. SRP_CV parameter

The SRP_CV parameter is used to convey the users’ username and the public value of the client to the server. SRP_CV parameter is used in I1 control packet. Servers can use SRP_CV parameter for access control by checking that the HIT in I1 control packet is valid and also that the user using it is also valid (see section Section 9 for further discussion about SRP_CV). SRP_CV parameter is non-critical.
and so can be safely ignored if unknown.

When mutual authentication is needed, a client can inform the server if it wants to authenticate the user (Administrator in practice) on server/responder by using field R. If field R is set client requires the server to authenticate by default the field R must not be set.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Length C</td>
<td>Length A</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
|   |   | /
|    / | A |
|     / | Padding |
| ++++++++++++++++++++++++++++++++

Type          TBD-IANA (998)
Length        Length in octets, excluding Type, Length, and Padding
Length C      Length of field C in octets
Length A      Length of field A in octets
R             Server authentication required if set
Reserved      Not in use, must be zeroed
C             Username, length from 8 to 2^8-1 bytes (255)
A             Public value of client, length from 1 to 2^16-1 bytes (65535)
Padding       Any Padding, if necessary, to make the TLV a multiple of 8 bytes.

3.2. SRP_CV_M parameter

On-path middleboxes append this parameter to the control packets. This parameter is identical with SRP_CV differing only by its type number (TBD-IANA (632997)).

3.3. SRP_SV Parameter

The SRP_SV parameter is used to convey the group and server values to the client. SRP_SV is used in R1 control packet. The group values consist of a prime (N) and a primitive root modulo of N, called generator (g). The server values are the public key of server and a salt found from servers passwd file. Public key of server (B) is a
service or application key and not the hosts HI. See Section 4 for further discussion about the usage.

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------+---------------+---------------</td>
</tr>
</tbody>
</table>

Type          TBD-IANA (998)
Length        Length in octets, excluding Type, Length, and Padding
Length N      Length of field N in octets
Length g      Length of field g in octets
Length s      Length of field s in octets
Length B      Length of field B in octets
N             Prime, length from 1 to 2^16-1 bytes (65535)
g            Generator, length from 1 to 2^16-1 bytes (65535)
s            Salt, length from 1 to 2^8-1 bytes (255)
B             Public value of server, length from 1 to 2^16-1 bytes (65535)
Padding       Any Padding, if necessary, to make the TLV a multiple of 8 bytes.

3.4. SRP_SV_M Parameter

On-path middleboxes append this parameter to the control packets. This parameter is identical with SRP_SV differing only by its type number (TBD-IANA (62998)). See Section 4 for further discussion about the usage.
3.5. SRP_E Parameter

SRP_E parameter is used to convey the evidence between peers. If this parameter is in I2 it is the clients evidence and if this parameter is in R2 it is the server’s evidence. Evidence is produced with interleaved SHA as described earlier and in [RFC2945].

3.6. SRP_E_M Parameter

On-path middleboxes append this parameter to the control packets. This parameter is identical with SRP_SV differing only by its type number TBD-IANA (62999).

4. Handling of SRP Parameters

This section defines how hosts should behave when SRP related parameters are present in HIP control packets. In Figure 2 we present a BEX with one-way SRP parameters (Only relevant parameters are depicted).
In the following we present a BEX with two-way SRP parameters. The only difference to the previous one-way authentication is the SRP_E parameter in the R2 control packet.

HIP Responders pre-create R1s in order to minimize the expensive cryptographic operations before Responder has established state. Using SRP_SV will prevent this. In order to use pre-created R1s,
Responders can append SRP_SV_M parameter into the unsigned part of the message.

In cases where server requires authentication receives I1 control packets that do not contain SRP_CV parameter, server MAY drop the I1 packet or OPTIONALLY answer with NOTIFY control packet containing NOTIFICATION parameter with one of the following status types.

<table>
<thead>
<tr>
<th>NOTIFY MESSAGES - ERROR TYPES</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>BLOCKED_BY_POLICY</td>
<td>42</td>
</tr>
</tbody>
</table>

The Responder is unwilling to set up an association for some policy reason (e.g., received HIT is NULL and policy does not allow opportunistic mode).

The Responder is not willing to set up an association because required authentication data is not present.

When endpoint receives the SRP_CV_M parameter it can notice that a middlebox on the way is performing authentication, but it may decide not to react to this parameter. More about the middleboxes in section Section 5

5. SRP and middleboxes

In the following we depict the situation where a middlebox needs to authenticate the user. The middle appends its parameters to the BEX packets traversing through it.
Middleboxes can degrade or restrict service such as bandwidth limitation up to refusing connections and reporting access violations when it does not find SRP_E_M parameter from I1 control packet.

The freshness of the authentication can be forced by using extensions defined in [HIP.Midauth]

6. Extension to HIP Native API

In this section we describe extensions to the HIP Native API defined in [Native.API]. The target readers of this section are application programmers.

As this section specifies sockets API extensions, it is written so that the syntax and semantics are in line with the POSIX standard [POSIX] as much as possible. The API specified in this section defines how to use socket options to set the used username and password so that the HIP layer can calculate the needed values and negotiate keys. The definition of the API is presented in C language and data types follow the POSIX format; intN_t means a signed integer of exactly N bits (e.g. int16_t) and uintN_t means an unsigned integer of exactly N bits (e.g. uint32_t).
<table>
<thead>
<tr>
<th>optname</th>
<th>get</th>
<th>set</th>
<th>description</th>
<th>dtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIP_USERINFO</td>
<td>o</td>
<td></td>
<td>Set the username and the password corresponding to it.</td>
<td>*</td>
</tr>
</tbody>
</table>

*: pointer to hip_userinfo data structure, defined below.

```c
struct hip_userinfo {
    uint8_t username[256]; /* username null terminated */
    uint8_t password[256]; /* password null terminated */
};
```

For example, a username and password can be specified as follows.

```c
struct hip_userinfo userinfo;
char usern[] = "username";
char passwd[] = "password";
memset(&userinfo, 0, sizeof(userinfo));
/* fill hip_userinfo data structure */
memcpy(&userinfo.username, &usern, sizeof(usern));
memcpy(&userinfo.password, &passwd, sizeof(passwd));
setsockopt(fd, IPPROTO_HIP, HIP_USERINFO, &userinfo, sizeof(userinfo));
```

7. Design Alternatives

Most of the devices (phones, PDAs, laptops) are operated by single user so the main approach described in this document can be used. Problems arise when clients have multiple users. For example, in case where machine A has users Aa and Ab. When Aa connects to server B, he creates a tunnel between the HITs of A and B. The problem in this, related to the SRP, is that now user Ab can also communicate using the same tunnel. Usage of Simultaneous Multi-Access extension [Hip.Sima] solves this multi-user to multi-user problem, because it allows HIP to use flow binding to identify and separate multiple ongoing upper layer data flows.
8. IANA Considerations

This document defines the SRP related parameters for the Host Identity Protocol [RFC5201]. These parameter are defined in Section 3. A new status type for NOTIFICATION parameter is defined in section Section 4

9. Security Considerations

Parameter SRP_CV that is sent in I1 control packet can be used for access control but it also advertises the user that currently is using the machine. This can be avoided with similar approach as in [BLIND]. In this approach the username is concatenated to the salt, hashed and so made unreadable to parties not knowing the salt (in [BLIND] nonce). This approach has the disadvantage, that the user has to know his/her salt. Better way would be to concatenate the HIT of the server with the username and hashing the result \( \text{UH} = \text{H(HIT-of-server | username)} \). For this to work efficiently, the server needs to index its passwd by the UH and the identification of user should be done based on the UH. While the attacker knows the HIT of the server it has to guess the username with brute force. Still this has a drawback that the hash (UH) stays the same. This could be avoided by sending a nonce with the username, but performance issues arise when the server needs to hash every username it knows with the nonce in order to identify the user.

Another way to protect the privacy of the user is to encrypt the SRP_CV with servers public key by using ENCRYPTED parameter, defined in [RFC5201]. This poses a problem of getting the public key of the server. The server can use DNS [RFC5205] or DHT [Opendht.Interface] in order to publish its public key. While the public key is available it poses delays when querying the key.

The above ways could be improved by using physical tokens that contain the required information, like the public key or the salt. With physical tokens One Time Passwords (OTP) could be introduced and so making the required authentication tuple into triple containing username, password verifier and OTP.

Parameters that are added by the middlebox are not signed. This should not pose any additional security problems. Attacker could replace or alter the SRP_*_M parameters and so letting the authentication fail. This can be done even without SRP parameters just by flipping random bits and so making the HMACs and signatures fail.
10. Acknowledgements

The authors would like to thank M. Komu J. Koskela, T. Heer and J. Heikkila of fruitful conversations on the subject.

11. References

11.1. Normative References

[HIP.Midauth]

[Hip.Sima]

[Native.API]

[Opendht.Interface]
Ahrenholz, J., "HIP DHT Interface", <draft-ahrenholz-hiprg-dht-03>.


11.2. Informative References


Author’s Address

Samu Varjonen
Helsinki Institute for Information Technology
Metsnneidonkuja 4
Helsinki
Finland

Fax: +35896949768
Email: samu.varjonen@hiit.fi
URI: http://www.hiit.fi