Network Working Group D. Taylor
Internet-Draft Forge Research Pty Ltd
Expires: August 6, 2001 February 5, 2001

Using SRP for TLS Authentication
draft-taylor-tls-srp-00.txt

Status of this Memo

Copyright Notice

Abstract

This memo presents a technique for using the SRP (Secure Remote Password) protocol as an authentication method for the TLS (Transport Layer Security) protocol.

1. Introduction

At the time of writing, TLS[1] uses public key certificiates with RSA/DSA digital signatures, or Kerberos, for authentication.

These authentication methods do not seem well suited to the applications now being adapted to use TLS (IMAP[3], FTP[4], or TELNET[5], for example). Given these protocols (and others like them) are designed to use the user name and password method of authentication, being able to use user names and passwords to authenticate the TLS connection seems to be a useful feature.

SRP[2] is an authentication method that allows the use of user names and passwords in a safe manner.

This document describes the use of the SRP authentication method for TLS.

2. SRP Authentication in TLS

2.1 Modifications to the TLS Handshake Sequence

The SRP protocol can not be implemented using the sequence of handshake messages defined in [1] due to the sequence in which the SRP messages must be sent.

This document proposes a new sequence of handshake messages for handshakes using the SRP authentication method.

2.1.1 Message Sequence

Handshake Message Flow for SRP Authentication

```
       Client                                     Server
         |                                          |
    Client Hello (U) ------------------------->    |
         |  <---------------------------- Server Hello
         |  <---------------------------- Server Key Exchange (g, N, s)
    Client Key Exchange (A) ------------------>    |
         |  <---------------------------- Server Key Exchange (B)
         |  <---------------------------- Server Hello Done
    change cipher spec                              |
    Finished --------------------------------->    |
         |                                change cipher spec
         |  <---------------------------- Finished
         |                                          |
```

The identifiers given after each message name refer to variables defined in [2] that are sent in that message.

This new handshake sequence has a number of differences from the standard TLS handshake sequence:

- o The client hello message has the user name appended to the message. This is allowable as stated in section 7.4.1.2 of [1].

- o The client cannot generate its its public key (A) until after it has received the (g) and (N) paramters from the server, and the client must send its public key before it receives the servers public key (B) (as stated in section 3 of [2]). This means the client must wait for a server key exchange message containing (g) and (N), send a client key exchange message containing (A), and then wait for another server key exchange message containing (B).

- o There is no server identification in this version of a TLS handshake. If an attacker gets the SRP password file, they can masquerade as the real system.

2.2 Changes to the Handshake Message Contents

This section describes the changes to the TLS handshake message contents when SRP is being used for authentication. The details of the on-the-wire changes are given in Section 2.5.

2.2.1 The Client Hello Message

The user name is appended to the standard client hello message. The extra data is included in the handshake message hashes.

2.2.2 The First Server Key Exchange Message

The server key exchange message in the first round contains the generator (g), the prime (N), and the salt value (s) read from the SRP password file.

2.2.3 The Client Key Exchange Message

The client key exchange message carries the clients public key (A), which is calculated using both information known locally, and information received in the first server key exchange message. This message MUST be sent between the first and second server key exchange messages.

2.2.4 The Second Server Key Exchange Message

The server key exchange message in the second round contains the servers public key (B).

2.3 Calculating the Pre-master Secret

The shared secret resulting from the SRP calculations (S) is used as the pre-master secret.

The finished messages perform the same function as the client and server evidence messages specified in [2]. If either the client or the server calculate an incorrect value, the finished messages will not be understood, and the connection will be dropped as specified in [1].

2.4 Cipher Suite Definitions

The following cipher suites are added by this draft. The numbers have been left blank until a suitable range has been selected.

```
CipherSuite     TLS_SRP_WITH_3DES_EDE_CBC_SHA       = { ?,? };

CipherSuite     TLS_SRP_WITH_RC4_128_SHA            = { ?,? };
```



```
CipherSuite     TLS_SRP_WITH_IDEA_CBC_SHA           = { ?,? };

CipherSuite     TLS_SRP_WITH_3DES_EDE_CBC_MD5       = { ?,? };

CipherSuite     TLS_SRP_WITH_RC4_128_MD5            = { ?,? };

CipherSuite     TLS_SRP_WITH_IDEA_CBC_MD5           = { ?,? };
```

2.5 New Message Structures

This section shows the structure of the messages passed during a handshake that uses SRP for authentication. The representation language used is that used in [1].

```
opaque Username<1..2^8-1>;

enum { non_srp, srp } CipherSuiteType;

struct {
   ProtocolVersion client_version;
   Random random;
```

```
      SessionID session_id;
      CipherSuite cipher_suites<2..2^16-1>;

      /* Need a better way to show the optional user_name field */
      select (CipherSuiteType) {
         case non_srp:
            CompressionMethod compression_methods<1..2^8-1>;
         case srp:
            CompressionMethod compression_methods<1..2^8-1>;
            Username user_name;   /* new entry */
      };
   } ClientHello;

   enum { rsa, diffie_hellman, srp } KeyExchangeAlgorithm;

   enum { first, second } KeyExchangeRound;

   struct {
      select (KeyExchangeRound) {
         case first:
            opaque srp_s<1..2^8-1>
            opaque srp_N<1..2^16-1>;
            opaque srp_g<1..2^16-1>;
         case second:
            opaque srp_B<1..2^16-1>;
      };
   } ServerSRPParams;     /* SRP parameters */
```



```
   struct {
      select (KeyExchangeAlgorithm) {
         case diffie_hellman:
            ServerDHParams params;
            Signature signed_params;
         case rsa:
            ServerRSAParams params;
            Signature signed_params;
         case srp:
            ServerSRPParams params;   /* new entry */
      };
   } ServerKeyExchange;

   struct {
```

```
      opaque srp_A<1..2^16-1>;
   } SRPClientEphemeralPublic;

   struct {
      select (KeyExchangeAlgorithm) {
         case rsa: EncryptedPreMasterSecret;
         case diffie_hellman: ClientDiffieHellmanPublic;
         case srp: SRPClientEphemeralPublic;   /* new entry */
      } exchange_keys;
   } ClientKeyExchange;
```


3. Security Considerations

There is no server identification in this version of a TLS handshake. If an attacker gets the SRP password file, they can masquerade as the real system.

What are the security issues of this new handshake sequence? Are the

SRP parameters passed in a safe order? Is it a problem having the username appended to the client hello message?

References

[1] Dierks, T. and C. Allen, "The TLS Protocol", RFC 2246, January 1999.

[2] Wu, T., "The SRP Authentication and Key Exchange System", RFC 2945, September 2000.

[3] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.

[4] Ford-Hutchinson, P., Carpenter, M., Hudson, T., Murray, E. and V. Wiegand, "Securing FTP with TLS", draft-murray-auth-ftp-ssl-06 (work in progress), September 2000.

[5] Boe, M. and J. Altman, "TLS-based Telnet Security", draft-ietf-tn3270e-telnet-tls-05 (work in progress), October 2000.

Author's Address

David Taylor
Forge Research Pty Ltd

EMail: DavidTaylor@forge.com.au
URI: http://www.protekt.com/

Full Copyright Statement

Acknowledgement

Funding for the RFC editor function is currently provided by the Internet Society.