Network Working Group
Internet Draft: Plaintext Transition
Document: draft-newman-sasl-plaintrans-04.txt
November 1997
Expires in six months

C. Newman
Innosoft

Plaintext Password SASL Mechanism for Transitioning

Status of this memo

Abstract

Unencrypted plaintext passwords are the biggest single risk to Internet application protocol security. Unfortunately, they are widely deployed, often tightly integrated into operating system services and very difficult to replace in an interoperable fashion.

This specification discusses some methods which can be used to eliminate unencrypted plaintext passwords. It also defines a SASL mechanism [SASL] which may be used by newer protocols such as ACAP [ACAP] to transition away from a legacy authentication database.

1. Conventions Used in this Document

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as defined in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS]. However, it is important to understand that this is not an IETF standards track document and therefore the key words only apply to conformance with this specification independent of any standards body.

2. Security Impact of Unencrypted Plaintext Passwords

Use of unencrypted plaintext passwords over the Internet is a severe security risk. In particular, a passive observer can get the password with any packet sniffer. This requires no technical expertise, as one can simply plug a consumer level computer into the network and run widely available network snoop programs. Such attacks are difficult or impossible to detect, and can only be prevented by complete physical and virtual security of the network between the client and server -- something which is usually impossible to achieve.

Unfortunately, most modern servers use legacy authentication databases, often tightly integrated with the server's operating system. These databases usually apply a one-way function to the user's password so that server break-ins only expose the users to dictionary attacks (testing likely passwords) and trojan horse server attacks (e.g., replacing the server with one which records user passwords). The result is that plaintext passwords are the only authentication technology today which will work with the vast majority of deployed authentication databases.

3. Transition Strategies

There are several techniques which a site may use to transition from unencrypted plaintext passwords. None of these are easy, but sites are STRONGLY ENCOURAGED to make the effort before a security breach occurs.

3.1. Deploying Encryption

One way to eliminate unencrypted plaintext passwords is to deploy encryption services for all protocols which use plaintext passwords. This is probably the most viable technique for sites which use a legacy authentication database, run servers from different vendors, and are unable to modify all password changing services.

There are several drawbacks to this. First, several common public key protocols are very expensive to deploy both in terms of administrative retraining and in order to purchase licenses,

software and certification. Second, for many simple protocols the encryption and public key services will be many times more complicated than the base protocol they protect. Third, it is illegal to use or export sufficiently strong encryption in many countries. Fourth, some currently deployed software using the non-standard SSL protocol is export crippled to 40-bit keys which is only marginally better than plaintext passwords. And even worse, some of this export-crippled software misleads the user into believing it is secure. Finally, the only current standards-track protocol suitable to encrypt TCP based protocols carrying passwords is [IPESP] which is difficult to deploy due to the need for support in the TCP/IP stack.

The TLS protocol, a work in progress, may address the last two problems. The secure shell protocol, another work in progress, may also address the first two problems.

3.2. One Time Upgrade to New Authentication System

Sites with sufficient control over their infrastructure may be able to deploy a new authentication system. This requires support from all clients, servers, remote login services and password changing services at the site.

There are several drawbacks to this approach. First, it requires all users to change their password or enter a new password. Second, it is very difficult to get support for the same mechanism in all the necessary components. Third, this is likely to require a single-vendor server solution as the only standards track option for interoperable server authentication is RADIUS [RADIUS] and it is designed solely for use by network access servers and protocol support is only available in PPP.

3.3. Gradual Transition on Password Change

A gradual transition can be achieved my modifying all password change services to set the password in both the old an new authentication systems. Components can be individually updated to use the new authentication system once both verifiers are available. This requires support from all password changing services at the site.

There are several drawbacks to this approach. First, it is likely to require parallel databases for a long time as it will be difficult to phase out the old system due to the need to upgrade all services and users (especially those who rarely change their

password). Second, this is likely to require a single-vendor server solution as the only standards track option for interoperable server authentication is RADIUS [RADIUS] and it is designed solely for use by network access servers and protocol support is only available in PPP.

3.4. Gradual Transition on Plaintext Mechanism

A gradual transition can be achieved by permitting use of a plaintext mechanism to authenticate to the old authentication service and create an entry in the new service. This also requires modifying all password change services.

There are several drawbacks to this approach. First, it is likely to require parallel databases until all services have been upgraded although it is a faster transition than that described in section 3.3. Second, it requires some support in protocols. Third, this is likely to require a single-vendor server solution as the only standards track option for interoperable server authentication is RADIUS [RADIUS] and it is designed solely for use by network access servers and protocol support is only available in PPP.

4. Error Codes For Transition

A number of error codes are defined in ACAP [ACAP] which may be used by ACAP and similar protocols to assist transition. This further explains those error codes and adds an additional error code "EXPIRED-PASS." These error codes are also suitable for use with IMAP [IMAP4].

EXPIRED-PASS
This indicates the user's password or passphrase has expired and needs to be changed. This is useful both for transition strategy 3.3, and to force users to change their password or passphrase more frequently.

TRANSITION-NEEDED
This occurs after a client attempts to authenticate using a mechanism other than plaintext. It indicates that the server has an entry for the specified user in a legacy authentication database but does not yet have credentials to offer the requested mechanism. A client which receives this error code may do a one-time login using the PLAIN mechanism (or another plaintext mechanism) after asking the user for permission to

activate the transition. Alternatively, the client could inform the user that they must change their password to transition. This is useful for transition strategy 3.4.

AUTH-TOO-WEAK
This indicates that the authentication mechanism is too weak for that user according to site security policy and that a stronger mechanism must be used instead. A client which receives this error code should try a stronger mechanism if available and stop using the weaker mechanism for that user.

ENCRYPT-NEEDED
This indicates that external strong encryption is needed in order to use the requested authentication mechanism. This is primarily intended for use with the PLAIN mechanism. A client which receives this may activate an encryption layer or try a stronger mechanism if available.

5. Plaintext Password SASL mechanism

Newer protocols, such as ACAP [ACAP], require a plaintext mechanism in order to implement transition strategy 3.4. This defines a mechanism suitable for that purpose. If this mechanism is implemented, it is important that it can be disabled by configuration.

The SASL [SASL] mechanism name is "PLAIN".

The mechanism consists of a single message from the client to the server. The client sends the authorization identity (identity to login as), followed by a US-ASCII NUL character, followed by the authentication identity (identity whose password will be used), followed by a US-ASCII NUL character, followed by the plaintext password. The client may leave the authorization identity empty to indicate that it is the same as the authentication identity.

The server will verify the authentication identity and password with the system authentication database and verify that the authentication credentials permit the client to login as the authorization identity. If both steps succeed, the user is logged in.

When used as a transition mechanism, the password will be stored in a new authentication database capable of supporting stronger authentication mechanisms. Once this is completed, the server MAY

refuse future use of the PLAIN mechanism by that authentication identity.

Non-US-ASCII characters are permitted as long as they can be represented in UTF-8 [UTF8]. Use of non-visible characters or characters which a user may be unable to enter on some keyboards is discouraged.

The formal grammar for the client message using Augmented BNF [ABNF] follows.

```
message         = [authorize-id] NUL authenticate-id NUL password

NUL             = %x00

US-ASCII-SAFE   = %x01-09 / %x0B-0C / %x0E-7F
                 ;; US-ASCII except CR, LF, NUL

UTF8-SAFE       = US-ASCII-SAFE / UTF8-2 / UTF8-3 / UTF8-4
                     / UTF8-5 / UTF8-6

UTF8-1          = %x80-BF

UTF8-2          = %xC0-DF UTF8-1

UTF8-3          = %xE0-EF 2UTF8-1

UTF8-4          = %xF0-F7 3UTF8-1

UTF8-5          = %xF8-FB 4UTF8-1

UTF8-6          = %xFC-FD 5UTF8-1

authenticate-id = 1*255UTF8-SAFE

authorize-id    = 1*255UTF8-SAFE

password        = 1*255UTF8-SAFE
```

6. Gradual Transition on PLAIN Example

Here is a sample transition exchange between an IMAP client and server. In this example, "C:" and "S:" indicate lines sent by the client and server respectively. If such lines are wrapped without a new "C:" or "S:" label, then the wrapping is for editorial clarity and is not part of the command.

Note that this example uses the IMAP profile [IMAP4] of SASL. The base64 encoding of challenges and responses, as well as the "+ " preceding the responses are part of the IMAP4 profile, not part of SASL itself. Newer profiles of SASL will include the initial client PLAIN message with the AUTHENTICATE command itself so the extra round trip below (the server response with an empty "+ ") can be eliminated.

In this example, the user's authentication identifier is "tim", his authorization identifier is the same, and his password is "tanstaaftanstaaf".

```
S: * OK IMAP4 server ready
C: A001 CAPABILITY
S: * CAPABILITY IMAP4 IMAP4rev1 AUTH=CRAM-MD5 AUTH=PLAIN
S: A001 OK done
C: A002 AUTHENTICATE CRAM-MD5
S: + PDE4OTYuNjk3MTcwOTUyQHBvc3RvZmZpY2UucmVzdG9uLm1jaS5uZXQ+
C: dGltIGI5MTNhNjAyYzdlZGE3YTQ5NWI0ZTZlNzMzNGQzODkw
S: A002 NO [TRANSITION-NEEDED] You can't login securely until
        you've changed your password on the server
<client gets permission from user to transition>
C: A003 AUTHENTICATE PLAIN
S: +
C: AHRpbQB0YW5zdGFhZnRhbnN0YWFm
S: A003 OK You can now login securely in the future.
C: A004 SELECT INBOX
   ...
```

7. Security Considerations

Security considerations are discussed throughout this document.

A man in the middle or a spoof server may be able to aquire the user's password by removing the announcement of available strong authentication mechanisms. Clients SHOULD record the available of strong authentication mechanisms on a given server and/or allow explicit configuration to prevent use of the PLAIN mechanism.

Some authentication mechanisms are susceptible to passive dictionary attacks. Password change agents should check new passwords against a dictionary and reject matches in order to reduce the effectiveness of this attack.

As there have been successful amateur attacks on 40-bit and 56-bit keys these are not deemed adequate security for passwords. The PLAIN mechanism SHOULD be used in combination with an external encryption layer using a key of sufficient strength to prevent

attack.

8. References

[ABNF] Crocker, D., "Augmented BNF for Syntax Specifications: ABNF", Work in progress: draft-ietf-drums-abnf-xx.txt

[ACAP] Newman, Myers, "ACAP -- Application Configuration Access Protocol", work in progress.

[CRAM-MD5] Klensin, Catoe, Krumviede, "IMAP/POP AUTHorize Extension for Simple Challenge/Response", RFC 2195, MCI, September 1997.

<ftp://ds.internic.net/rfc/rfc2195.txt>

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, University of Washington, December 1996.

<ftp://ds.internic.net/rfc/rfc2060.txt>

[IPESP] Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 1827, Naval Research Laboratory, August 1995.

<ftp://ds.internic.net/rfc/rfc1827.txt>

[KERBEROS-GSS] Linn, "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, OpenVision Technologies, June 1996.

<ftp://ds.internic.net/rfc/rfc1964.txt>

[KEYWORDS] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997.

<ftp://ds.internic.net/rfc/rfc2119.txt>

[MIME-SEC] Galvin, Murphy, Crocker, Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, Trusted Information Systems, CyberCash, Innosoft International, October 1995.

<ftp://ds.internic.net/rfc/rfc1847.txt>

[POP3] Myers, J., Rose, M., "Post Office Protocol - Version 3", RFC 1939, Carnegie Mellon, Dover Beach Consulting, Inc., May 1996.

<ftp://ds.internic.net/rfc/rfc1939.txt>

[POP-AUTH] Myers, "POP3 AUTHentication command", RFC 1734, Carnegie Mellon, December 1994.

<ftp://ds.internic.net/rfc/rfc1734.txt>

[RADIUS] Rigney, Rubens, Simpson, Willens, "Remote Authentication Dial In User Service (RADIUS)", RFC 2138, Livingston, Merit, Daydreamer, April 1997.

<ftp://ds.internic.net/rfc/rfc2138.txt>

[SASL] Myers, "Simple Authentication and Security Layer (SASL)", RFC 2222, Netscape Communications, October 1997.

<ftp://ds.internic.net/rfc/rfc2222.txt>

[UTF8] Yergeau, F. "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, Alis Technologies, October 1996.

<ftp://ds.internic.net/rfc/rfc2044.txt>

9. Acknowledgements

Thanks to John Myers, Larry Osterman, Ned Freed and Kevin Carosso for feedback on this proposal.

10. Author's Address

Chris Newman
Innosoft International, Inc.
1050 Lakes Drive
West Covina, CA 91790 USA

Email: chris.newman@innosoft.com