J. Michener & D. Fritch
Novell, Inc.
13 March 1998

Multi-Authentication Framework Method for SOCKS V5

Status of this Memo

This document is an Internet-Draft. 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.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

SOCKS V5 [RFC 1928] provides a means to select one from among a number of authentication methods, but does not provide any means for utilizing multiple authentication methods to obtain desired authentication properties. This document specifies the Multi-Authentication Framework Method (MAF) which is a method extension to SOCKS Version 5 to support the efficient management of composite authentication protocols composed of more than one authentication methods. MAF is a client-initiated but server managed framework. MAF relies upon a trusted Authentication Management Server (AMS) to select the authentication methods to be invoked, order them as appropriate, and assign integrity grades to the final authentication after all methods invoked have been completed.

MAF SOCKS V5 Identifier

During initial SOCKS V5 negotiation, the client and server negotiate the authentication method. The METHOD ID to invoke the MAF shall be X'??'.

Subnegotiation

Subnegotiation begins after the client has selected MAF. Subnegotiation is conducted under the control of the server.

The client sends an initial version identifier/method selection message:

INSTR	VER	FLAGS	NMETHODS	METHODS
1	1	2	4	4 to 1020

The INSTR field is an octet that specifies the operation being performed. Defined values at this time are:

X'FF'	failure
X'00'	success
X'01'	MAF sub-methods supported
X'02'	request additional MAF sub-methods supported
X'03'	do
X'04'	what next?
X'05'	process
X 06'	Acknowledge

To start the subnegotiation the INSTR field is set to MAF sub-methods supported", X'01'.

The VER field is an octet and is set to the version of the MAF framework. At this time VER is set to X'00'.

The FLAGS field is a uint16 value. At this time it is set to X'0000'. It provides future tunability for higher versions and serves to word align the data.

The NMETHODS field is an octet that contains the number of MAF sub-method identifiers that appear in the METHODS field (1 to 255). If the client has another block of potential sub-methods that it can send to the server, it includes the MORE_METHODS_AVAILABLE method ID as the last method in the list to notify the server to request the next block of methods, if necessary.

The METHOD identifiers are 32 bit unsigned int values in network byte order. MAF methods are fixed and inalterable after they have been registered. Consequentially, MAF methods do not have version identification and version incompatibilities are avoided. If a method is found to be inadequate, the revised method should be registered and a new MAF method ID should be issued.

The server may select one of the MAF sub-methods given in METHODS (if none of the methods meets the AMS requirements and the client did not note that it had more methods available, the method selected would be FAILURE) and send a DO METHOD command:

INSTR	VER	FLAGS	METHOD	LEN	DATA
1	1	2	4	4	LEN

The INSTR field is set to "do", X'03'. As above, the VER field is set to version of the MAF parent method. At this time VER is set to X'00' and the FLAGS field is set to X'0000'. The MAF sub-method ID being performed is entered into the METHOD field. Data being transported between the client and server modules is sent in the DATA section as a (void) array, with the length of the array specified in the LEN field.

If the server instructs the client to send more methods, via the X'02' request additional MAF sub-methods supported command , the server will use the FLAGS field to specify a relative method download (instructing the cleint to send only methods that have not already been sent) or an absolute method download (instructing the client to start again with the method list). The FLAGS field will be X 0000' for a relative method download and X 0001' for an absolute method download.

The client and the server then call the appropriate modules to execute the specified function. The exchange between the selected client and server modules will utilize the following data structure.

INSTR	VER	FLAGS	METHOD	LEN	DATA
1	1	2	4	4	LEN

The INSTR field is set to "process", X'05'. At this time VER is set to X'00'and the FLAGS field is set to X'0000'. The MAF sub-method ID is specified by the METHOD variable. If any data or parameters are to be sent to the method, they are sent in the DATA section as a (void) array, with the length of the array specified in the LEN field.

The client and server method modules return success or failure to both the client and server, respectively. In all cases, the client sends the following message to the server.

INSTR	VER	FLAGS
1	1	2

The INSTR field is set to "what next?", X'04'. At this time VER is set to X'00' and the FLAGS field is set to X'0000'.

In event of failure of an authentication method or of the authentication process, the server may instruct the client to close the connection. If the sub-method was successful or if the sub-method failed and the server did not instruct the client to close the connection, the server may instruct the client to execute another MAF sub-method module.

At the end of the process, as determined by the server, the server will send back:

INSTR	VER	FLAGS	METHOD	LEN	DATA
1	1	2	4	4	LEN

If the authentication process succeeded, the INSTR field will be set to success , X'00' and the method ID is set to SUCCESS, X'00000000'. If the authentication process failed, the INSTR field will be set to failure , X'FF' and the method ID is set to FAILURE, X'FFFFFFFF'. In either case, at this time VER is set to X'00' and the FLAGS field is set to X'0000'. If any data or parameters are to be sent to the process to be run upon successful authentication, they are sent in the DATA section as a (void) array, with the length of the array specified in the LEN field.

Current MAF Sub-Methods:

X'FFFFFFFF'	FAILURE
X'00000000'	SUCCESS
X'00000001'	Internal Test Method
X'00000002'	MORE_METHODS_AVAILABLE
X'00000003' To X 0000FFFF	Reserved for proprietary methods
X'00010000'up	MAF general authentication method numbers

Acknowledgements

Thanks to Russel Weiser for assistance with this document.

References

[RFC 1928] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., & Jones, L., "SOCKS Protocol V5," April 1996.

Author's Address

John Michener
Novell, Inc.
122 East 1700 South
Provo Utah, 84606-6194

Phone: +1 801 861-5478
Fax: +1 801 861-2522
Email: jmichener@novell.com

Dan Fritch
Novell, Inc.
122 East 1700 South
Provo Utah, 84606-6194

Phone: +1 801 861-5136
Fax: +1 801 861-2522
Email: dfritch@novell.com