Network Working Group
Internet-Draft
Expires: April 24, 2005

C. Lonvick, Ed.
Cisco Systems, Inc

October 24, 2004

SSH Protocol Assigned Numbers
draft-ietf-secsh-assignednumbers-07.txt

Abstract

This document defines the instructions to the IANA and the initial state of the IANA assigned numbers for the SSH protocol. It is intended only for the initialization of the IANA registries referenced in the documents.

Table of Contents

1. Editor's Note

The references in this document are statically defined. However, the locations of the referenced materials are dynamic and are changing with the whims of the Working Group. Please do not comment to the editor or the Working Group about inaccuracies along those lines in this document at this time. (This paragraph will be removed before this document is submitted to the RFC Editor.)

2. Introduction

This document does not define any new protocols. It is intended only to create the initial state of the IANA databases for the SSH protocol and also contains instructions for future assignments. Except for one HISTORIC algorithm generally regarded as obsolete, this document does not define any new protocols or any number ranges not already defined in: [SSH-ARCH], [SSH-TRANS], [SSH-USERAUTH], [SSH-CONNECT].

3. Conventions Used in This Document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", and "MAY" that appear in this document are to be interpreted as described in [RFC2119].

The keywords "PRIVATE USE", "HIERARCHICAL ALLOCATION", "FIRST COME FIRST SERVED", "EXPERT REVIEW", "SPECIFICATION REQUIRED", "IESG APPROVAL", "IETF CONSENSUS", and "STANDARDS ACTION" that appear in this document when used to describe namespace allocation are to be interpreted as described in [RFC2434]. These designations are repeated in this document for clarity.

PRIVATE USE - For private or local use only, with the type and purpose defined by the local site. No attempt is made to prevent multiple sites from using the same value in different (and incompatible) ways. There is no need for IANA to review such assignments and assignments are not generally useful for interoperability.

HIERARCHICAL ALLOCATION - Delegated managers can assign values provided they have been given control over that part of the name space. IANA controls the higher levels of the namespace according to one of the other policies.

FIRST COME FIRST SERVED - Anyone can obtain an assigned number, so long as they provide a point of contact and a brief description of what the value would be used for. For numbers, the exact value is generally assigned by the IANA; with names, specific names are

usually requested.

EXPERT REVIEW - approval by a Designated Expert is required.

SPECIFICATION REQUIRED - Values and their meaning must be documented in an RFC or other permanent and readily available reference, in sufficient detail so that interoperability between independent implementations is possible.

IESG APPROVAL - New assignments must be approved by the IESG, but there is no requirement that the request be documented in an RFC (though the IESG has discretion to request documents or other supporting materials on a case-by-case basis).

IETF CONSENSUS - New values are assigned through the IETF consensus process. Specifically, new assignments are made via RFCs approved by the IESG. Typically, the IESG will seek input on prospective assignments from appropriate persons (e.g., a relevant Working Group if one exists).

STANDARDS ACTION - Values are assigned only for Standards Track RFCs approved by the IESG.

4. IANA Considerations

This entire document is the IANA considerations for the SSH protocol as is defined in [SSH-ARCH], [SSH-TRANS], [SSH-USERAUTH], [SSH-CONNECT]. This section contains conventions used in naming the namespaces, the initial state of the registry, and instructions for future assignments.

4.1 Message Numbers

The Message Number is an 8-bit value, which describes the payload of a packet.

4.1.1 Conventions

Protocol packets have message numbers in the range 1 to 255. These numbers are allocated as follows:

Transport layer protocol:

1 to 19	Transport layer generic (e.g. disconnect, ignore, debug, etc.)
20 to 29	Algorithm negotiation
30 to 49	Key exchange method specific (numbers can be reused

for different authentication methods)

User authentication protocol:

50 to 59 User authentication generic
60 to 79 User authentication method specific (numbers can be reused for different authentication methods)

Connection protocol:

80 to 89 Connection protocol generic
90 to 127 Channel related messages

Reserved for client protocols:

128 to 191 Reserved

Local extensions:

192 to 255 Local extensions

4.1.2 Initial Assignments

Message ID	Value	Reference
SSH_MSG_DISCONNECT	1	[SSH-TRANS]
SSH_MSG_IGNORE	2	[SSH-TRANS]
SSH_MSG_UNIMPLEMENTED	3	[SSH-TRANS]
SSH_MSG_DEBUG	4	[SSH-TRANS]
SSH_MSG_SERVICE_REQUEST	5	[SSH-TRANS]
SSH_MSG_SERVICE_ACCEPT	6	[SSH-TRANS]
SSH_MSG_KEXINIT	20	[SSH-TRANS]
SSH_MSG_NEWKEYS	21	[SSH-TRANS]
SSH_MSG_KEXDH_INIT	30	[SSH-TRANS]
SSH_MSG_KEXDH_REPLY	31	[SSH-TRANS]
SSH_MSG_USERAUTH_REQUEST	50	[SSH-USERAUTH]
SSH_MSG_USERAUTH_FAILURE	51	[SSH-USERAUTH]
SSH_MSG_USERAUTH_SUCCESS	52	[SSH-USERAUTH]
SSH_MSG_USERAUTH_BANNER	53	[SSH-USERAUTH]
SSH_MSG_USERAUTH_PK_OK	60	[SSH-USERAUTH]
SSH_MSG_GLOBAL_REQUEST	80	[SSH-CONNECT]
SSH_MSG_REQUEST_SUCCESS	81	[SSH-CONNECT]
SSH_MSG_REQUEST_FAILURE	82	[SSH-CONNECT]
SSH_MSG_CHANNEL_OPEN	90	[SSH-CONNECT]
SSH_MSG_CHANNEL_OPEN_CONFIRMATION	91	[SSH-CONNECT]
SSH_MSG_CHANNEL_OPEN_FAILURE	92	[SSH-CONNECT]

SSH_MSG_CHANNEL_WINDOW_ADJUST	93	[SSH-CONNECT]
SSH_MSG_CHANNEL_DATA	94	[SSH-CONNECT]
SSH_MSG_CHANNEL_EXTENDED_DATA	95	[SSH-CONNECT]
SSH_MSG_CHANNEL_EOF	96	[SSH-CONNECT]
SSH_MSG_CHANNEL_CLOSE	97	[SSH-CONNECT]
SSH_MSG_CHANNEL_REQUEST	98	[SSH-CONNECT]
SSH_MSG_CHANNEL_SUCCESS	99	[SSH-CONNECT]
SSH_MSG_CHANNEL_FAILURE	100	[SSH-CONNECT]

4.1.3 Future Assignments

Requests for assignments of new message numbers in the range of 1 to 127 MUST be done through the STANDARDS ACTION method as described in [RFC2434].

Requests for assigments of new message numbers in the range of 128 to 191 MUST be done through the IETF CONSENSUS method as described in [RFC2434].

The IANA will not control the message numbers range of 192 through 255. This range will be left for PRIVATE USE.

4.2 Disconnection Messages Reason Codes and Descriptions

The Disconnection Message 'reason code' is a uint32 value. The associated Disconnection Message 'description string' is a human-readable message which describes the disconnect reason.

4.2.1 Conventions

Protocol packets containing the SSH_MSG_DISCONNECT message MUST have Disconnection Message 'reason code' values in the range of 0x00000001 to 0xFFFFFFFF.

4.2.2 Initial Assignments

description string	reason code	Reference
SSH_DISCONNECT_HOST_NOT_ALLOWED_TO_CONNECT	1	[SSH-TRANS]
SSH_DISCONNECT_PROTOCOL_ERROR	2	[SSH-TRANS]
SSH_DISCONNECT_KEY_EXCHANGE_FAILED	3	[SSH-TRANS]
SSH_DISCONNECT_RESERVED	4	[SSH-TRANS]
SSH_DISCONNECT_MAC_ERROR	5	[SSH-TRANS]
SSH_DISCONNECT_COMPRESSION_ERROR	6	[SSH-TRANS]
SSH_DISCONNECT_SERVICE_NOT_AVAILABLE	7	[SSH-TRANS]
SSH_DISCONNECT_PROTOCOL_VERSION_NOT_SUPPORTED	8	[SSH-TRANS]

SSH_DISCONNECT_HOST_KEY_NOT_VERIFIABLE	9	[SSH-TRANS]
SSH_DISCONNECT_CONNECTION_LOST	10	[SSH-TRANS]
SSH_DISCONNECT_BY_APPLICATION	11	[SSH-TRANS]
SSH_DISCONNECT_TOO_MANY_CONNECTIONS	12	[SSH-TRANS]
SSH_DISCONNECT_AUTH_CANCELLED_BY_USER	13	[SSH-TRANS]
SSH_DISCONNECT_NO_MORE_AUTH_METHODS_AVAILABLE	14	[SSH-TRANS]
SSH_DISCONNECT_ILLEGAL_USER_NAME	15	[SSH-TRANS]

4.2.3 Future Assignments

Disconnection Message 'reason code' values MUST be assigned sequentially. Requests for assignments of new Disconnection Message 'reason codes', and their associated Disconnection Message 'description string', in the range of 0x00000010 through 0xFFFFFFE9 MUST be done through the IETF CONSENSUS method as described in [RFC2434]. The IANA will not assign Disconnection Message 'reason codes' in the range of 0xFFFFFFF0 through 0xFFFFFFFF. Disconnection Message 'reason code' values in that range are left for PRIVATE USE as described in [RFC2434].

4.3 Channel Connection Failure Reason Codes and Descriptions

The Channel Connection Failure 'reason code' is a uint32 value. The associated Channel Connection Failure 'description string' is a human-readable message which describes the channel connection failure reason.

4.3.1 Conventions

Protocol packets containing the SSH_MSG_CHANNEL_OPEN_FAILURE message MUST have Channel Connection Failure 'reason code' values in the range of 0x00000001 to 0xFFFFFFFF.

4.3.2 Initial Assignments

description string	reason code	Reference
SSH_OPEN_ADMINISTRATIVELY_PROHIBITED	1	[SSH-CONNECT]
SSH_OPEN_CONNECT_FAILED	2	[SSH-CONNECT]
SSH_OPEN_UNKNOWN_CHANNEL_TYPE	3	[SSH-CONNECT]
SSH_OPEN_RESOURCE_SHORTAGE	4	[SSH-CONNECT]

4.3.3 Future Assignments

Channel Connection Failure 'reason code' values MUST be assigned

sequentially. Requests for assignments of new Channel Connection Failure 'reason code' values, and their associated Channel Connection Failure 'description string', in the range of 0x00000005 to 0xFFFFFFE9 MUST be done through the IETF CONSENSUS method as described in [RFC2434]. The IANA will not assign Channel Connection Failure 'reason code' values in the range of 0xFFFFFFF0 to 0xFFFFFFFF. Channel Connection Failure 'reason code' values in that range are left for PRIVATE USE as described in [RFC2434].

4.4 Extended Channel Data Transfer data_type_code and Data

The Extended Channel Data Transfer 'data_type_code' is an uint23 value. The associated Extended Channel Data Transfer 'data' is a human-readable message which describes the type of data allowed to be transferred in the channel.

4.4.1 Conventions

Protocol packets containing the SSH_MSG_CHANNEL_EXTENDED_DATA message MUST have Extended Channel Data Transfer 'data_type_code' values in the range of 0x00000001 to 0xFFFFFFFF.

4.4.2 Initial Assignments

data	data_type_code	Reference
SSH_EXTENDED_DATA_STDERR	1	[SSH-CONNECT]

4.4.3 Future Assignments

Extended Channel Data Transfer 'data_type_code' values MUST be assigned sequentially. Requests for assignments of new Extended Channel Data Transfer 'data_type_code' values, and their associated Extended Channel Data Transfer 'data' strings, in the range of 0x00000002 to 0xFFFFFFE9 MUST be done through the IETF CONSENSUS method as described in [RFC2434]. The IANA will not assign Extended Channel Data Transfer 'data_type_code' values in the range of 0xFFFFFFF0 to 0xFFFFFFFF. Extended Channel Data Transfer 'data_type_code' values in that range are left for PRIVATE USE as described in [RFC2434].

4.5 Pseudo-Terminal Encoded Terminal Modes

SSH_MSG_CHANNEL_REQUEST messages with a "pty-req" string MUST contain "encoded terminal modes". These "encoded terminal modes" are opcode-argument pairs consisting of an opcode and an argument.

4.5.1 Conventions

Protocol packets containing the SSH_MSG_CHANNEL_REQUEST message with a "pty-req" string MUST contain "encoded terminal modes" with an opcode of 1 byte. The opcode values are in the range of 1 to 255. Opcodes 1 to 159 have a single uint32 argument. Opcodes 160 to 255 are not yet defined.

4.5.2 Initial Assignments

opcode	argument	description
0	TTY_OP_END	Indicates end of options.
1	VINTR	Interrupt character; 255 if none. Similarly for the other characters. Not all of these characters are supported on all systems.
2	VQUIT	The quit character (sends SIGQUIT signal on POSIX systems).
3	VERASE	Erase the character to left of the cursor.
4	VKILL	Kill the current input line.
5	VEOF	End-of-file character (sends EOF from the terminal).
6	VEOL	End-of-line character in addition to carriage return and/or linefeed.
7	VEOL2	Additional end-of-line character.
8	VSTART	Continues paused output (normally control-Q).
9	VSTOP	Pauses output (normally control-S).
10	VSUSP	Suspends the current program.
11	VDSUSP	Another suspend character.
12	VREPRINT	Reprints the current input line.
13	VWERASE	Erases a word left of cursor.
14	VLNEXT	Enter the next character typed literally, even if it is a special character
15	VFLUSH	Character to flush output.
16	VSWTCH	Switch to a different shell layer.
17	VSTATUS	Prints system status line (load, command, pid, etc).
18	VDISCARD	Toggles the flushing of terminal output.
30	IGNPAR	The ignore parity flag. The parameter SHOULD be 0 if this flag is FALSE set, and 1 if it is TRUE.
31	PARMRK	Mark parity and framing errors.
32	INPCK	Enable checking of parity errors.
33	ISTRIP	Strip 8th bit off characters.
34	INLCR	Map NL into CR on input.
35	IGNCR	Ignore CR on input.
36	ICRNL	Map CR to NL on input.

37	IUCLC	Translate uppercase characters to lowercase.
38	IXON	Enable output flow control.
39	IXANY	Any char will restart after stop.
40	IXOFF	Enable input flow control.
41	IMAXBEL	Ring bell on input queue full.
50	ISIG	Enable signals INTR, QUIT, [D]SUSP.
51	ICANON	Canonicalize input lines.
52	XCASE	Enable input and output of uppercase characters by preceding their lowercase equivalents with "\".
53	ECHO	Enable echoing.
54	ECHOE	Visually erase chars.
55	ECHOK	Kill character discards current line.
56	ECHONL	Echo NL even if ECHO is off.
57	NOFLSH	Don't flush after interrupt.
58	TOSTOP	Stop background jobs from output.
59	IEXTEN	Enable extensions.
60	ECHOCTL	Echo control characters as ^(Char).
61	ECHOKE	Visual erase for line kill.
62	PENDIN	Retype pending input.
70	OPOST	Enable output processing.
71	OLCUC	Convert lowercase to uppercase.
72	ONLCR	Map NL to CR-NL.
73	OCRNL	Translate carriage return to newline (output).
74	ONOCR	Translate newline to carriage return-newline (output).
75	ONLRET	Newline performs a carriage return (output).
90	CS7	7 bit mode.
91	CS8	8 bit mode.
92	PARENB	Parity enable.
93	PARODD	Odd parity, else even.
128	TTY_OP_ISPEED	Specifies the input baud rate in bits per second.
129	TTY_OP_OSPEED	Specifies the output baud rate in bits per second.

4.5.3 Future Assignments

Requests for assignments of new opcodes and their associated arguments MUST be done through the IETF CONSENSUS method as described in [RFC2434].

4.6 Names

In the following sections, the values for the name spaces are textual. The conventions and instructions to the IANA for future assignments are given in this section. The initial assignments are given in their respective sections.

4.6.1 Conventions for Names

All names registered by the IANA in the following sections MUST be printable US-ASCII strings, and MUST NOT contain the characters at-sign ("@"), comma (","), or whitespace or control characters (ASCII codes 32 or less). Names are case-sensitive, and MUST NOT be longer than 64 characters.

A provision is made here for locally extensible names. The IANA will not register, and will not control names with the at-sign ("@") in them. Names with the at-sign in them will have the format of "name@domainname" (without the double quotes) where the part preceeding the at-sign is the name. The format of the part preceding the at sign is not specified, however these names MUST be printable US-ASCII strings, and MUST NOT contain the comma character (","), or whitespace, or control characters (ASCII codes 32 or less). The part following the at-sign MUST be a valid, fully qualified internet domain name [RFC1034] controlled by the person or organization defining the name. Names are case-sensitive, and MUST NOT be longer than 64 characters. It is up to each domain how it manages its local namespace. It has been noted that these names resemble [RFC0822] email addresses. This is purely coincidental and actually has nothing to do with [RFC0822]. An example of a locally defined name is "ourcipher-cbc@example.com" (without the double quotes).

4.6.2 Future Assignments of Names

Requests for assignments of new Names MUST be done through the IETF CONSENSUS method as described in [RFC2434].

4.7 Service Names

The Service Name is used to describe a protocol layer.

Service name	Reference
ssh-userauth	[SSH-USERAUTH]
ssh-connection	[SSH-CONNECT]

4.8 Authentication Method Names

The Authentication Method Name is used to describe an authentication method for the "ssh-userauth" service [SSH-USERAUTH].

Method name	Reference
publickey	[SSH-USERAUTH, Section 4]
password	[SSH-USERAUTH, Section 5]
hostbased	[SSH-USERAUTH, Section 6]
none	[SSH-USERAUTH, Section 2.3]

4.9 Connection Protocol Assigned Names

The following are the Connection Protocol Type and Request names.

4.9.1 Connection Protocol Channel Types

Channel type	Reference
session	[SSH-CONNECT, Section 4.1]
x11	[SSH-CONNECT, Section 4.3.2]
forwarded-tcpip	[SSH-CONNECT, Section 5.2]
direct-tcpip	[SSH-CONNECT, Section 5.2]

4.9.2 Connection Protocol Global Request Names

Request type	Reference
tcpip-forward	[SSH-CONNECT, Section 5.1]
cancel-tcpip-forward	[SSH-CONNECT, Section 5.1]

4.9.3 Connection Protocol Channel Request Names

Request type	Reference
pty-req	[SSH-CONNECT, Section 4.2]
x11-req	[SSH-CONNECT, Section 4.3.1]
env	[SSH-CONNECT, Section 4.4]
shell	[SSH-CONNECT, Section 4.5]
exec	[SSH-CONNECT, Section 4.5]
subsystem	[SSH-CONNECT, Section 4.5]
window-change	[SSH-CONNECT, Section 4.7]
xon-xoff	[SSH-CONNECT, Section 4.8]
signal	[SSH-CONNECT, Section 4.9]

exit-status	[SSH-CONNECT, Section 4.10]
exit-signal	[SSH-CONNECT, Section 4.10]

4.9.4 Initial Assignment of Signal Names

Signal	Reference
ABRT	[SSH-CONNECT]
ALRM	[SSH-CONNECT]
FPE	[SSH-CONNECT]
HUP	[SSH-CONNECT]
ILL	[SSH-CONNECT]
INT	[SSH-CONNECT]
KILL	[SSH-CONNECT]
PIPE	[SSH-CONNECT]
QUIT	[SSH-CONNECT]
SEGV	[SSH-CONNECT]
TERM	[SSH-CONNECT]
USR1	[SSH-CONNECT]
USR2	[SSH-CONNECT]

4.10 Key Exchange Method Names

The Key Exchange Method Name describes a key-exchange method for the protocol [SSH-TRANS]. Note that, for historical reasons, the name "diffie-hellman-group1-sha1" is used for a key exchange method using Oakley Group 2. This is considered an aberration and should not be repeated. Any future specifications of Diffie Hellman key exchange using Oakley groups defined in [RFC2412] or its successors should be named using the group numbers assigned by IANA, and names of the form "diffie-hellman-groupN-sha1" should be reserved for this purpose.

Editor's Note: diffie-hellman-group14-sha1 is controversial at the moment. It is being discussed on the mailing list.

Method name	Reference
diffie-hellman-group1-sha1	[SSH-TRANS, Section 8.1]
diffie-hellman-group14-sha1	[SSH-TRANS, Section 8.2]

4.11 Assigned Algorithm Names

The following names identify the Encryption Algorithm Names.

4.11.1 Encryption Algorithm Names

Cipher name	Reference
3des-cbc	[SSH-TRANS, Section 4.3]
blowfish-cbc	[SSH-TRANS, Section 4.3]
twofish256-cbc	[SSH-TRANS, Section 4.3]
twofish-cbc	[SSH-TRANS, Section 4.3]
twofish192-cbc	[SSH-TRANS, Section 4.3]
twofish128-cbc	[SSH-TRANS, Section 4.3]
aes256-cbc	[SSH-TRANS, Section 4.3]
aes192-cbc	[SSH-TRANS, Section 4.3]
aes128-cbc	[SSH-TRANS, Section 4.3]
serpent256-cbc	[SSH-TRANS, Section 4.3]
serpent192-cbc	[SSH-TRANS, Section 4.3]
serpent128-cbc	[SSH-TRANS, Section 4.3]
arcfour	[SSH-TRANS, Section 4.3]
idea-cbc	[SSH-TRANS, Section 4.3]
cast128-cbc	[SSH-TRANS, Section 4.3]
none	[SSH-TRANS, Section 4.3]
des-cbc	[FIPS-46-3] HISTORIC; See page 4 of [FIPS 46-3]

4.11.2 MAC Algorithm Names

The following names identify the MAC Algorithm Names.

MAC name	Reference
hmac-sha1	[SSH-TRANS, Section 4.4]
hmac-sha1-96	[SSH-TRANS, Section 4.4]
hmac-md5	[SSH-TRANS, Section 4.4]
hmac-md5-96	[SSH-TRANS, Section 4.4]
none	[SSH-TRANS, Section 4.4]

4.11.3 Public Key Algorithm Names

This table identifies the Public Key Algorithm names.

Algorithm name	Reference
ssh-dss	[SSH-TRANS, Section 4.6]
ssh-rsa	[SSH-TRANS, Section 4.6]
x509v3-sign-rsa	[SSH-TRANS, Section 4.6]
x509v3-sign-dss	[SSH-TRANS, Section 4.6]

spki-sign-rsa	[SSH-TRANS, Section 4.6]
spki-sign-dss	[SSH-TRANS, Section 4.6]
pgp-sign-rsa	[SSH-TRANS, Section 4.6]
pgp-sign-dss	[SSH-TRANS, Section 4.6]

4.11.4 Compression Algorithm Names

The following names identify the Compression Algorithm names.

Algorithm name	Reference
none	[SSH-TRANS, Section 4.2]
zlib	[SSH-TRANS, Section 4.2]

5. References

5.1 Normative References

[SSH-ARCH] Ylonen, T. and C. Lonvick, "SSH Protocol Architecture", I-D draft-ietf-architecture-17.txt, October 2004.

[SSH-TRANS] Ylonen, T. and C. Lonvick, "SSH Transport Layer Protocol", I-D draft-ietf-transport-19.txt, October 2004.

[SSH-USERAUTH] Ylonen, T. and C. Lonvick, "SSH Authentication Protocol", I-D draft-ietf-userauth-22.txt, October 2004.

[SSH-CONNECT] Ylonen, T. and C. Lonvick, "SSH Connection Protocol", I-D draft-ietf-connect-20.txt, October 2004.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2412] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

5.2 Informative References

[RFC0822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[FIPS-46-3] U.S. Dept. of Commerce, "FIPS PUB 46-3, Data Encryption Standard (DES)", October 1999.

Author's Address

Chris Lonvick (editor)
Cisco Systems, Inc
12515 Research Blvd.
Austin 78759
USA

EMail: clonvick@cisco.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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