**Secure Shell**

From Wikipedia, the free encyclopedia

[Jump to navigation](https://en.wikipedia.org/wiki/Secure_Shell#mw-head) [Jump to search](https://en.wikipedia.org/wiki/Secure_Shell#searchInput)

"SSH" redirects here. For other uses, see [SSH (disambiguation)](https://en.wikipedia.org/wiki/SSH_(disambiguation)).

|  |
| --- |
| [**Internet protocol suite**](https://en.wikipedia.org/wiki/Internet_protocol_suite) |
| [**Application layer**](https://en.wikipedia.org/wiki/Application_layer) |
| * [BGP](https://en.wikipedia.org/wiki/Border_Gateway_Protocol) * [DHCP](https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol) * [DNS](https://en.wikipedia.org/wiki/Domain_Name_System) * [FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol) * [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) * [HTTPS](https://en.wikipedia.org/wiki/HTTPS) * [IMAP](https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol) * [LDAP](https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol) * [MGCP](https://en.wikipedia.org/wiki/Media_Gateway_Control_Protocol) * [MQTT](https://en.wikipedia.org/wiki/MQTT) * [NNTP](https://en.wikipedia.org/wiki/Network_News_Transfer_Protocol) * [NTP](https://en.wikipedia.org/wiki/Network_Time_Protocol) * [POP](https://en.wikipedia.org/wiki/Post_Office_Protocol) * [PTP](https://en.wikipedia.org/wiki/Precision_Time_Protocol) * [ONC/RPC](https://en.wikipedia.org/wiki/Open_Network_Computing_Remote_Procedure_Call) * [RTP](https://en.wikipedia.org/wiki/Real-time_Transport_Protocol) * [RTSP](https://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol) * [RIP](https://en.wikipedia.org/wiki/Routing_Information_Protocol) * [SIP](https://en.wikipedia.org/wiki/Session_Initiation_Protocol) * [SMTP](https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) * [SNMP](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol) * SSH * [Telnet](https://en.wikipedia.org/wiki/Telnet) * [TLS/SSL](https://en.wikipedia.org/wiki/Transport_Layer_Security) * [XMPP](https://en.wikipedia.org/wiki/XMPP) * [*more...*](https://en.wikipedia.org/wiki/Category:Application_layer_protocols) |
| [**Transport layer**](https://en.wikipedia.org/wiki/Transport_layer) |
| * [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) * [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol) * [DCCP](https://en.wikipedia.org/wiki/Datagram_Congestion_Control_Protocol) * [SCTP](https://en.wikipedia.org/wiki/Stream_Control_Transmission_Protocol) * [RSVP](https://en.wikipedia.org/wiki/Resource_Reservation_Protocol) * [*more...*](https://en.wikipedia.org/wiki/Category:Transport_layer_protocols) |
| [**Internet layer**](https://en.wikipedia.org/wiki/Internet_layer) |
| * [IP](https://en.wikipedia.org/wiki/Internet_Protocol)   + [IPv4](https://en.wikipedia.org/wiki/IPv4)   + [IPv6](https://en.wikipedia.org/wiki/IPv6) * [ICMP](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol) * [ICMPv6](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol_for_IPv6) * [ECN](https://en.wikipedia.org/wiki/Explicit_Congestion_Notification) * [IGMP](https://en.wikipedia.org/wiki/Internet_Group_Management_Protocol) * [IPsec](https://en.wikipedia.org/wiki/IPsec) * [*more...*](https://en.wikipedia.org/wiki/Category:Internet_layer_protocols) |
| [**Link layer**](https://en.wikipedia.org/wiki/Link_layer) |
| * [ARP](https://en.wikipedia.org/wiki/Address_Resolution_Protocol) * [NDP](https://en.wikipedia.org/wiki/Neighbor_Discovery_Protocol) * [OSPF](https://en.wikipedia.org/wiki/Open_Shortest_Path_First) * [Tunnels](https://en.wikipedia.org/wiki/Tunneling_protocol)   + [L2TP](https://en.wikipedia.org/wiki/Layer_2_Tunneling_Protocol) * [PPP](https://en.wikipedia.org/wiki/Point-to-Point_Protocol) * [MAC](https://en.wikipedia.org/wiki/Medium_access_control)   + [Ethernet](https://en.wikipedia.org/wiki/Ethernet)   + [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi)   + [DSL](https://en.wikipedia.org/wiki/Digital_subscriber_line)   + [ISDN](https://en.wikipedia.org/wiki/Integrated_Services_Digital_Network)   + [FDDI](https://en.wikipedia.org/wiki/Fiber_Distributed_Data_Interface) * [*more...*](https://en.wikipedia.org/wiki/Category:Link_protocols) |
| * [v](https://en.wikipedia.org/wiki/Template:IPstack) * [t](https://en.wikipedia.org/wiki/Template_talk:IPstack) * [e](https://en.wikipedia.org/w/index.php?title=Template:IPstack&action=edit) |

**Secure Shell** (**SSH**) is a [cryptographic](https://en.wikipedia.org/wiki/Cryptography) [network protocol](https://en.wikipedia.org/wiki/Network_protocol) for operating network services securely over an unsecured network.[[1]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-rfc4251-1) Typical applications include remote [command-line](https://en.wikipedia.org/wiki/Command-line_interface), [login](https://en.wikipedia.org/wiki/Login), and remote command execution, but any [network service](https://en.wikipedia.org/wiki/Network_service) can be secured with SSH.

SSH provides a [secure channel](https://en.wikipedia.org/wiki/Secure_channel) over an unsecured network by using a [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) architecture, connecting an [SSH client](https://en.wikipedia.org/wiki/SSH_client) application with an [SSH server](https://en.wikipedia.org/wiki/SSH_server).[[2]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-rfc4252-2) The protocol specification distinguishes between two major versions, referred to as SSH-1 and SSH-2. The standard TCP port for SSH is 22. SSH is generally used to access [Unix-like](https://en.wikipedia.org/wiki/Unix-like) operating systems, but it can also be used on [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows). [Windows 10](https://en.wikipedia.org/wiki/Windows_10) uses [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH) as its default [SSH client](https://en.wikipedia.org/wiki/SSH_client) and [SSH server](https://en.wikipedia.org/wiki/SSH_server).[[3]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-3)

Despite popular misconception, SSH is not an implementation of [Telnet](https://en.wikipedia.org/wiki/Telnet) with cryptography provided by the [Secure Sockets Layer (SSL)](https://en.wikipedia.org/wiki/Secure_Sockets_Layer).

SSH was designed as a replacement for [Telnet](https://en.wikipedia.org/wiki/Telnet) and for [unsecured](https://en.wikipedia.org/wiki/Computer_security) remote [shell](https://en.wikipedia.org/wiki/Unix_shell) protocols such as the Berkeley [rsh](https://en.wikipedia.org/wiki/Remote_Shell) and the related [rlogin](https://en.wikipedia.org/wiki/Rlogin) and [rexec](https://en.wikipedia.org/wiki/Remote_Process_Execution) protocols. Those protocols send information, notably [passwords](https://en.wikipedia.org/wiki/Password), in [plaintext](https://en.wikipedia.org/wiki/Plaintext), rendering them susceptible to interception and disclosure using [packet analysis](https://en.wikipedia.org/wiki/Packet_analyzer).[[4]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-4) The [encryption](https://en.wikipedia.org/wiki/Encryption) used by SSH is intended to provide confidentiality and integrity of data over an unsecured network, such as the [Internet](https://en.wikipedia.org/wiki/Internet), although files leaked by [Edward Snowden](https://en.wikipedia.org/wiki/Edward_Snowden) indicate that the [National Security Agency](https://en.wikipedia.org/wiki/National_Security_Agency) can sometimes decrypt SSH, allowing them to read, modify and selectively suppress the contents of SSH sessions.[[5]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-Spiegel2014-5)

SSH can also be run using [SCTP](https://en.wikipedia.org/wiki/SCTP) rather than TCP as the connection oriented transport layer protocol.[[6]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-6)

The [IANA](https://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority) has assigned [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) [port](https://en.wikipedia.org/wiki/Port_(computer_networking)) 22, [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol) port 22 and [SCTP](https://en.wikipedia.org/wiki/Stream_Control_Transmission_Protocol) port 22 for this protocol.[[7]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-7)



**Contents**

* [1 Definition](https://en.wikipedia.org/wiki/Secure_Shell#Definition)
* [2 Authentication: OpenSSH key management](https://en.wikipedia.org/wiki/Secure_Shell#Authentication:_OpenSSH_key_management)
* [3 Usage](https://en.wikipedia.org/wiki/Secure_Shell#Usage)
* [4 History and development](https://en.wikipedia.org/wiki/Secure_Shell#History_and_development)
  + [4.1 Version 1.x](https://en.wikipedia.org/wiki/Secure_Shell#Version_1.x)
  + [4.2 Version 2.x](https://en.wikipedia.org/wiki/Secure_Shell#Version_2.x)
  + [4.3 Version 1.99](https://en.wikipedia.org/wiki/Secure_Shell#Version_1.99)
  + [4.4 OpenSSH and OSSH](https://en.wikipedia.org/wiki/Secure_Shell#OpenSSH_and_OSSH)
* [5 Uses](https://en.wikipedia.org/wiki/Secure_Shell#Uses)
  + [5.1 File transfer protocols](https://en.wikipedia.org/wiki/Secure_Shell#File_transfer_protocols)
* [6 Architecture](https://en.wikipedia.org/wiki/Secure_Shell#Architecture)
* [7 Algorithms](https://en.wikipedia.org/wiki/Secure_Shell#Algorithms)
* [8 Vulnerabilities](https://en.wikipedia.org/wiki/Secure_Shell#Vulnerabilities)
  + [8.1 SSH-1](https://en.wikipedia.org/wiki/Secure_Shell#SSH-1)
  + [8.2 CBC plaintext recovery](https://en.wikipedia.org/wiki/Secure_Shell#CBC_plaintext_recovery)
  + [8.3 Possible vulnerabilities](https://en.wikipedia.org/wiki/Secure_Shell#Possible_vulnerabilities)
* [9 Standards documentation](https://en.wikipedia.org/wiki/Secure_Shell#Standards_documentation)
* [10 See also](https://en.wikipedia.org/wiki/Secure_Shell#See_also)
* [11 References](https://en.wikipedia.org/wiki/Secure_Shell#References)
* [12 Further reading](https://en.wikipedia.org/wiki/Secure_Shell#Further_reading)
* [13 External links](https://en.wikipedia.org/wiki/Secure_Shell#External_links)

**Definition**

SSH uses [public-key cryptography](https://en.wikipedia.org/wiki/Public-key_cryptography) to [authenticate](https://en.wikipedia.org/wiki/Authentication) the remote computer and allow it to authenticate the user, if necessary.[[2]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-rfc4252-2) There are several ways to use SSH; one is to use automatically generated public-private key pairs to simply encrypt a network connection, and then use [password](https://en.wikipedia.org/wiki/Password) authentication to log on.

Another is to use a manually generated public-private key pair to perform the authentication, allowing users or programs to log in without having to specify a password. In this scenario, anyone can produce a matching pair of different keys (public and private). The public key is placed on all computers that must allow access to the owner of the matching private key (the owner keeps the private key secret). While authentication is based on the private key, the key itself is never transferred through the network during authentication. SSH only verifies whether the same person offering the public key also owns the matching private key. In all versions of SSH it is important to verify unknown [public keys](https://en.wikipedia.org/wiki/Public_key), i.e. [associate the public keys with identities](https://en.wikipedia.org/wiki/Public-key_cryptography#Associating_public_keys_with_identities), before accepting them as valid. Accepting an attacker's public key without validation will authorize an unauthorized attacker as a valid user.

**Authentication:** [**OpenSSH**](https://en.wikipedia.org/wiki/OpenSSH) **key management**

On [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems, the list of authorized public keys is typically stored in the home directory of the user that is allowed to log in remotely, in the file ~/.ssh/authorized\_keys.[[8]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-8) This file is respected by SSH only if it is not writable by anything apart from the owner and root. When the public key is present on the remote end and the matching private key is present on the local end, typing in the password is no longer required. However, for additional security the private key itself can be locked with a passphrase.

The private key can also be looked for in standard places, and its full path can be specified as a command line setting (the option *-i* for ssh). The [ssh-keygen](https://en.wikipedia.org/wiki/Ssh-keygen) utility produces the public and private keys, always in pairs.

SSH also supports password-based authentication that is encrypted by automatically generated keys. In this case, the attacker could imitate the legitimate server side, ask for the password, and obtain it ([man-in-the-middle attack](https://en.wikipedia.org/wiki/Man-in-the-middle_attack)). However, this is possible only if the two sides have never authenticated before, as SSH remembers the key that the server side previously used. The SSH client raises a warning before accepting the key of a new, previously unknown server. Password authentication can be disabled.

**Usage**

SSH is typically used to log into a remote machine and execute commands, but it also supports [tunneling](https://en.wikipedia.org/wiki/Tunneling_protocol), [forwarding](https://en.wikipedia.org/wiki/Port_forwarding) [TCP ports](https://en.wikipedia.org/wiki/TCP_and_UDP_port) and [X11](https://en.wikipedia.org/wiki/X11) connections; it can transfer files using the associated [SSH file transfer](https://en.wikipedia.org/wiki/SSH_file_transfer_protocol) (SFTP) or [secure copy](https://en.wikipedia.org/wiki/Secure_copy) (SCP) protocols.[[2]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-rfc4252-2) SSH uses the [client-server](https://en.wikipedia.org/wiki/Client-server) model.

The [standard TCP port](https://en.wikipedia.org/wiki/List_of_well-known_ports_(computing)) 22 has been assigned for contacting SSH servers.[[9]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-9)

An SSH [client](https://en.wikipedia.org/wiki/Client_(computing)) program is typically used for establishing connections to an SSH [daemon](https://en.wikipedia.org/wiki/Daemon_(computer_software)) accepting remote connections. Both are commonly present on most modern [operating systems](https://en.wikipedia.org/wiki/Operating_systems), including [macOS](https://en.wikipedia.org/wiki/MacOS), most distributions of [Linux](https://en.wikipedia.org/wiki/Linux), [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD), [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD), [NetBSD](https://en.wikipedia.org/wiki/NetBSD), [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) and [OpenVMS](https://en.wikipedia.org/wiki/OpenVMS). Notably, versions of [Windows](https://en.wikipedia.org/wiki/Windows) prior to Windows 10 version 1709 do not include SSH by default. [Proprietary](https://en.wikipedia.org/wiki/Proprietary_software), [freeware](https://en.wikipedia.org/wiki/Freeware) and [open source](https://en.wikipedia.org/wiki/Open-source_software) (e.g. [PuTTY](https://en.wikipedia.org/wiki/PuTTY),[[10]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-10) and the version of [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH) which is part of [Cygwin](https://en.wikipedia.org/wiki/Cygwin)[[11]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-11)) versions of various levels of complexity and completeness exist. File managers for UNIX-like systems (e.g. [Konqueror](https://en.wikipedia.org/wiki/Konqueror)) can use the [FISH](https://en.wikipedia.org/wiki/Files_transferred_over_shell_protocol) protocol to provide a split-pane GUI with drag-and-drop. The open source Windows program [WinSCP](https://en.wikipedia.org/wiki/WinSCP)[[12]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-12) provides similar file management (synchronization, copy, remote delete) capability using PuTTY as a back-end. Both WinSCP[[13]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-13) and PuTTY[[14]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-14) are available packaged to run directly off a USB drive, without requiring installation on the client machine. Setting up an SSH server in Windows typically involves enabling a feature in Settings app. In [Windows 10 version 1709](https://en.wikipedia.org/wiki/Windows_10_version_history#Version_1709_(Fall_Creators_Update)), an official Win32 port of OpenSSH is available.

SSH is important in [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) to solve connectivity problems, avoiding the security issues of exposing a cloud-based virtual machine directly on the Internet. An SSH tunnel can provide a secure path over the Internet, through a firewall to a virtual machine.[[15]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-15)

**History and development**

**Version 1.x**

In 1995, [Tatu Ylönen](https://en.wikipedia.org/wiki/Tatu_Yl%C3%B6nen), a researcher at [Helsinki University of Technology](https://en.wikipedia.org/wiki/Helsinki_University_of_Technology), Finland, designed the first version of the protocol (now called **SSH-1**) prompted by a password-[sniffing](https://en.wikipedia.org/wiki/Packet_analyzer) attack at his [university network](https://en.wikipedia.org/wiki/University_network).[[16]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-16) The goal of SSH was to replace the earlier [rlogin](https://en.wikipedia.org/wiki/Rlogin), [TELNET](https://en.wikipedia.org/wiki/TELNET), [FTP](https://en.wikipedia.org/wiki/FTP)[[17]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-17) and [rsh](https://en.wikipedia.org/wiki/Remote_Shell) protocols, which did not provide strong authentication nor guarantee confidentiality. Ylönen released his implementation as [freeware](https://en.wikipedia.org/wiki/Freeware) in July 1995, and the tool quickly gained in popularity. Towards the end of 1995, the SSH user base had grown to 20,000 users in fifty countries.

In December 1995, Ylönen founded [SSH Communications Security](https://en.wikipedia.org/wiki/SSH_Communications_Security) to market and develop SSH. The original version of the SSH software used various pieces of [free software](https://en.wikipedia.org/wiki/Free_software), such as [GNU libgmp](https://en.wikipedia.org/wiki/GNU_Multi-Precision_Library), but later versions released by SSH Communications Security evolved into increasingly [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software).

It was estimated that by the year 2000 the number of users had grown to 2 million.[[18]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-Nicholas_Rosasco_and_David_Larochelle-18)

**Version 2.x**

"Secsh" was the official [Internet Engineering Task Force's](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force) (IETF) name for the IETF working group responsible for version 2 of the SSH protocol.[[19]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-19) In 2006, a revised version of the protocol, **SSH-2**, was adopted as a standard. This version is incompatible with SSH-1. SSH-2 features both security and feature improvements over SSH-1. Better security, for example, comes through [Diffie–Hellman key exchange](https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange) and strong [integrity](https://en.wikipedia.org/wiki/Integrity) checking via [message authentication codes](https://en.wikipedia.org/wiki/Message_authentication_code). New features of SSH-2 include the ability to run any number of [shell](https://en.wikipedia.org/wiki/Unix_shell) sessions over a single SSH connection.[[20]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-20) Due to SSH-2's superiority and popularity over SSH-1, some implementations such as libssh (v0.8.0+),[[21]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-21) [Lsh](https://en.wikipedia.org/wiki/Lsh)[[22]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-22) and [Dropbear](https://en.wikipedia.org/wiki/Dropbear_(software))[[23]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-23) support only the SSH-2 protocol.

**Version 1.99**

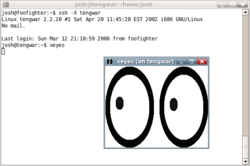
In January 2006, well after version 2.1 was established, [RFC 4253](https://tools.ietf.org/html/rfc4253) specified that an SSH server which supports both 2.0 and prior versions of SSH should identify its protoversion as 1.99.[[24]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-24) This is not an actual version but a method to identify [backward compatibility](https://en.wikipedia.org/wiki/Backward_compatibility).

**OpenSSH and OSSH**

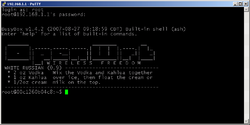
In 1999, developers, wanting a free software version to be available, went back to the older 1.2.12 release of the original SSH program, which was the last released under an [open source license](https://en.wikipedia.org/wiki/Open_source_license). Björn Grönvall's OSSH was subsequently developed from this codebase. Shortly thereafter, [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD) developers [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) Grönvall's code and did extensive work on it, creating [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH), which shipped with the 2.6 release of OpenBSD. From this version, a "portability" branch was formed to port OpenSSH to other operating systems.[[25]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-25)

As of 2005, [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH) was the single most popular SSH implementation, coming by default in a large number of operating systems. OSSH meanwhile has become obsolete.[[26]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-26) OpenSSH continues to be maintained and supports the SSH-2 protocol, having expunged SSH-1 support from the codebase with the OpenSSH 7.6 release.

**Uses**

[](https://en.wikipedia.org/wiki/File:X11_ssh_tunnelling.png)

Example of tunneling an [X11](https://en.wikipedia.org/wiki/X11) application over SSH: the user 'josh' has SSHed from the local machine 'foofighter' to the remote machine 'tengwar' to run [xeyes](https://en.wikipedia.org/wiki/Xeyes).

[](https://en.wikipedia.org/wiki/File:OpenWrtPuTTY.png)

Logging into [OpenWrt](https://en.wikipedia.org/wiki/OpenWrt) via SSH using [PuTTY](https://en.wikipedia.org/wiki/PuTTY) running on [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows).

SSH is a protocol that can be used for many applications across many platforms including most [Unix](https://en.wikipedia.org/wiki/Unix) variants ([Linux](https://en.wikipedia.org/wiki/Linux), the [BSDs](https://en.wikipedia.org/wiki/BSD) including [Apple's](https://en.wikipedia.org/wiki/Apple_Inc) [macOS](https://en.wikipedia.org/wiki/MacOS), and [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system))), as well as [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows). Some of the applications below may require features that are only available or compatible with specific SSH clients or servers. For example, using the SSH protocol to implement a [VPN](https://en.wikipedia.org/wiki/VPN) is possible, but presently only with the [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH) server and client implementation.

* For login to a shell on a remote host (replacing [Telnet](https://en.wikipedia.org/wiki/Telnet) and [rlogin](https://en.wikipedia.org/wiki/Rlogin))
* For executing a single command on a remote host (replacing [rsh](https://en.wikipedia.org/wiki/Remote_shell))
* For setting up automatic (passwordless) login to a remote server (for example, using [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH)[[27]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-27))
* In combination with [rsync](https://en.wikipedia.org/wiki/Rsync) to back up, copy and mirror files efficiently and securely
* For [forwarding](https://en.wikipedia.org/wiki/Port_forwarding) a port
* For [tunneling](https://en.wikipedia.org/wiki/Tunneling_protocol) (not to be confused with a [VPN](https://en.wikipedia.org/wiki/VPN), which [routes](https://en.wikipedia.org/wiki/VPN#Routing) packets between different networks, or [bridges](https://en.wikipedia.org/wiki/VPN#OSI_Layer_1_services) two [broadcast domains](https://en.wikipedia.org/wiki/Broadcast_domain) into one).
* For using as a full-fledged encrypted VPN. Note that only [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH) server and client supports this feature.
* For forwarding [X](https://en.wikipedia.org/wiki/X_Window_System) from a remote [host](https://en.wikipedia.org/wiki/Host_(network)) (possible through multiple intermediate hosts)
* For browsing the web through an encrypted proxy connection with SSH clients that support the [SOCKS protocol](https://en.wikipedia.org/wiki/SOCKS).
* For securely mounting a directory on a remote server as a [filesystem](https://en.wikipedia.org/wiki/File_system) on a local computer using [SSHFS](https://en.wikipedia.org/wiki/SSHFS).
* For automated remote monitoring and management of servers through one or more of the mechanisms discussed above.
* For development on a mobile or embedded device that supports SSH.
* For securing file transfer protocols.

**File transfer protocols**

The Secure Shell protocols are used in several file transfer mechanisms.

* [Secure copy](https://en.wikipedia.org/wiki/Secure_copy) (SCP), which evolved from [RCP](https://en.wikipedia.org/wiki/Rcp_(Unix)) protocol over SSH
* [rsync](https://en.wikipedia.org/wiki/Rsync), intended to be more efficient than SCP. Generally runs over an SSH connection.
* [SSH File Transfer Protocol](https://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol) (SFTP), a secure alternative to [FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol) (not to be confused with [FTP over SSH](https://en.wikipedia.org/wiki/FTP_over_SSH) or [FTPS](https://en.wikipedia.org/wiki/FTPS))
* [Files transferred over shell protocol](https://en.wikipedia.org/wiki/Files_transferred_over_shell_protocol) (a.k.a. FISH), released in 1998, which evolved from [Unix shell](https://en.wikipedia.org/wiki/Unix_shell) commands over SSH
* [Fast and Secure Protocol](https://en.wikipedia.org/wiki/Fast_and_Secure_Protocol) (FASP), aka *Aspera*, uses SSH for control and UDP ports for data transfer.

**Architecture**

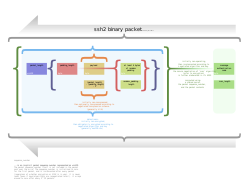
[](https://en.wikipedia.org/wiki/File:Ssh_binary_packet_alt.svg)

Diagram of the SSH-2 binary packet.

The SSH-2 protocol has an internal architecture (defined in [RFC 4251](https://tools.ietf.org/html/rfc4251)) with well-separated layers, namely:

* The *transport* layer ([RFC 4253](https://tools.ietf.org/html/rfc4253)), which typically runs on top of TCP/IP. This layer handles initial key exchange as well as server authentication, and sets up encryption, compression and integrity verification. It exposes to the upper layer an interface for sending and receiving plaintext packets with sizes of up to 32,768 bytes each (more can be allowed by the implementation). The transport layer also arranges for key re-exchange, usually after 1 GB of data has been transferred or after 1 hour has passed, whichever occurs first.
* The *user authentication* layer ([RFC 4252](https://tools.ietf.org/html/rfc4252)). This layer handles client authentication and provides a number of authentication methods. Authentication is *client-driven*: when one is prompted for a password, it may be the SSH client prompting, not the server. The server merely responds to the client's authentication requests. Widely used user-authentication methods include the following:
  + *password*: a method for straightforward password authentication, including a facility allowing a password to be changed. Not all programs implement this method.
  + *publickey*: a method for [public key-based authentication](https://en.wikipedia.org/wiki/Public-key_cryptography), usually supporting at least [DSA](https://en.wikipedia.org/wiki/Digital_Signature_Algorithm), [ECDSA](https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm) or [RSA](https://en.wikipedia.org/wiki/RSA_(algorithm)) keypairs, with other implementations also supporting [X.509](https://en.wikipedia.org/wiki/X.509) certificates.
  + *keyboard-interactive* ([RFC 4256](https://tools.ietf.org/html/rfc4256)): a versatile method where the server sends one or more prompts to enter information and the client displays them and sends back responses keyed-in by the user. Used to provide [one-time password](https://en.wikipedia.org/wiki/One-time_password) authentication such as [S/Key](https://en.wikipedia.org/wiki/S/Key) or [SecurID](https://en.wikipedia.org/wiki/SecurID). Used by some OpenSSH configurations when [PAM](https://en.wikipedia.org/wiki/Pluggable_authentication_modules) is the underlying host-authentication provider to effectively provide password authentication, sometimes leading to inability to log in with a client that supports just the plain *password* authentication method.
  + [GSSAPI](https://en.wikipedia.org/wiki/Generic_Security_Services_Application_Program_Interface) authentication methods which provide an extensible scheme to perform SSH authentication using external mechanisms such as [Kerberos 5](https://en.wikipedia.org/wiki/Kerberos_(protocol)) or [NTLM](https://en.wikipedia.org/wiki/NTLM), providing [single sign-on](https://en.wikipedia.org/wiki/Single_sign-on) capability to SSH sessions. These methods are usually implemented by commercial SSH implementations for use in organizations, though OpenSSH does have a working GSSAPI implementation.
* The *connection* layer ([RFC 4254](https://tools.ietf.org/html/rfc4254)). This layer defines the concept of channels, channel requests and global requests using which SSH services are provided. A single SSH connection can host multiple channels simultaneously, each transferring data in both directions. Channel requests are used to relay out-of-band channel-specific data, such as the changed size of a terminal window or the exit code of a server-side process. Additionally, each channel performs its own flow control using the receive window size. The SSH client requests a server-side port to be forwarded using a global request. Standard channel types include:
  + *shell* for terminal shells, SFTP and exec requests (including SCP transfers)
  + *direct-tcpip* for client-to-server forwarded connections
  + *forwarded-tcpip* for server-to-client forwarded connections
* The [SSHFP](https://en.wikipedia.org/wiki/SSHFP) DNS record ([RFC 4255](https://tools.ietf.org/html/rfc4255)) provides the public host key fingerprints in order to aid in verifying the authenticity of the host.

This open architecture provides considerable flexibility, allowing the use of SSH for a variety of purposes beyond a secure shell. The functionality of the transport layer alone is comparable to [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security) (TLS); the user-authentication layer is highly extensible with custom authentication methods; and the connection layer provides the ability to multiplex many secondary sessions into a single SSH connection, a feature comparable to [BEEP](https://en.wikipedia.org/wiki/BEEP) and not available in TLS.

**Algorithms**

* [EdDSA](https://en.wikipedia.org/wiki/EdDSA)[[28]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-28), [ECDSA](https://en.wikipedia.org/wiki/ECDSA), [RSA](https://en.wikipedia.org/wiki/RSA_(cryptosystem)) and [DSA](https://en.wikipedia.org/wiki/Digital_Signature_Algorithm) for [public key cryptography](https://en.wikipedia.org/wiki/Public_key_cryptography).[[29]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-RFC5656-29)
* [ECDH](https://en.wikipedia.org/wiki/ECDH) and [Diffie–Hellman](https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman) for [key exchange](https://en.wikipedia.org/wiki/Key_exchange).[[29]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-RFC5656-29)
* [UMAC](https://en.wikipedia.org/wiki/UMAC) and [HMAC](https://en.wikipedia.org/wiki/HMAC) for [MAC](https://en.wikipedia.org/wiki/Message_authentication_code).[[30]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-30)
* [AES](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard) for [symmetric encryption](https://en.wikipedia.org/wiki/Symmetric_encryption).
* [AES-GCM](https://en.wikipedia.org/wiki/AES-GCM)[[31]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-31) and [ChaCha20](https://en.wikipedia.org/wiki/ChaCha20)-[Poly1305](https://en.wikipedia.org/wiki/Poly1305) for [AEAD](https://en.wikipedia.org/wiki/AEAD) encryption.
* [SHA](https://en.wikipedia.org/wiki/Secure_Hash_Algorithm) for [key fingerprint](https://en.wikipedia.org/wiki/Key_fingerprint).

**Vulnerabilities**

**SSH-1**

In 1998, a vulnerability was described in SSH 1.5 which allowed the unauthorized insertion of content into an encrypted SSH stream due to insufficient data integrity protection from [CRC-32](https://en.wikipedia.org/wiki/CRC-32) used in this version of the protocol.[[32]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-32)[[33]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-33) A fix known as SSH Compensation Attack Detector[[34]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-34) was introduced into most implementations. Many of these updated implementations contained a new integer overflow vulnerability[[35]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-35) that allowed attackers to execute arbitrary code with the privileges of the SSH daemon, typically root.

In January 2001 a vulnerability was discovered that allows attackers to modify the last block of an [IDEA](https://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm)-encrypted session.[[36]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-36) The same month, another vulnerability was discovered that allowed a malicious server to forward a client authentication to another server.[[37]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-cert1-37)

Since SSH-1 has inherent design flaws which make it vulnerable, it is now generally considered obsolete and should be avoided by explicitly disabling fallback to SSH-1.[[37]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-cert1-37) Most modern servers and clients support SSH-2.[[38]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-38)

**CBC plaintext recovery**

In November 2008, a theoretical vulnerability was discovered for all versions of SSH which allowed recovery of up to 32 bits of plaintext from a block of ciphertext that was encrypted using what was then the standard default encryption mode, [CBC](https://en.wikipedia.org/wiki/Block_cipher_modes_of_operation#Cipher-block_chaining_(CBC)).[[39]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-SSH_CBC_vulnerability-39) The most straightforward solution is to use [CTR](https://en.wikipedia.org/wiki/Block_cipher_modes_of_operation#Counter_(CTR)), counter mode, instead of CBC mode, since this renders SSH resistant to the attack.[[39]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-SSH_CBC_vulnerability-39)

**Possible vulnerabilities**

On December 28, 2014 [Der Spiegel](https://en.wikipedia.org/wiki/Der_Spiegel) published classified information[[5]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-Spiegel2014-5) leaked by whistleblower [Edward Snowden](https://en.wikipedia.org/wiki/Edward_Snowden) which suggests that the [National Security Agency](https://en.wikipedia.org/wiki/National_Security_Agency) may be able to decrypt some SSH traffic. The technical details associated with such a process were not disclosed.

An analysis in 2017 of the hacking tools BothanSpy & Gyrfalcon suggested that the SSH protocol itself was not compromised.[[40]](https://en.wikipedia.org/wiki/Secure_Shell#cite_note-40)

**Standards documentation**

The following [RFC](https://en.wikipedia.org/wiki/Request_for_Comments) publications by the [IETF](https://en.wikipedia.org/wiki/IETF) "secsh" [working group](https://en.wikipedia.org/wiki/Working_group) document SSH-2 as a proposed [Internet standard](https://en.wikipedia.org/wiki/Internet_standard).

* [RFC 4250](https://tools.ietf.org/html/rfc4250) - The Secure Shell (SSH) Protocol Assigned Numbers
* [RFC 4251](https://tools.ietf.org/html/rfc4251) - The Secure Shell (SSH) Protocol Architecture
* [RFC 4252](https://tools.ietf.org/html/rfc4252) - The Secure Shell (SSH) Authentication Protocol
* [RFC 4253](https://tools.ietf.org/html/rfc4253) - The Secure Shell (SSH) Transport Layer Protocol
* [RFC 4254](https://tools.ietf.org/html/rfc4254) - The Secure Shell (SSH) Connection Protocol
* [RFC 4255](https://tools.ietf.org/html/rfc4255) - Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints
* [RFC 4256](https://tools.ietf.org/html/rfc4256) - Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)
* [RFC 4335](https://tools.ietf.org/html/rfc4335) - The Secure Shell (SSH) Session Channel Break Extension
* [RFC 4344](https://tools.ietf.org/html/rfc4344) - The Secure Shell (SSH) Transport Layer Encryption Modes
* [RFC 4345](https://tools.ietf.org/html/rfc4345) - Improved Arcfour Modes for the Secure Shell (SSH) Transport Layer Protocol

It was later modified and expanded by the following publications.

* [RFC 4419](https://tools.ietf.org/html/rfc4419) - Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol (March 2006)
* [RFC 4432](https://tools.ietf.org/html/rfc4432) - RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol (March 2006)
* [RFC 4462](https://tools.ietf.org/html/rfc4462) - Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol (May 2006)
* [RFC 4716](https://tools.ietf.org/html/rfc4716) - The Secure Shell (SSH) Public Key File Format (November 2006)
* [RFC 4819](https://tools.ietf.org/html/rfc4819) - Secure Shell Public Key Subsystem (March 2007)
* [RFC 5647](https://tools.ietf.org/html/rfc5647) - AES Galois Counter Mode for the Secure Shell Transport Layer Protocol (August 2009)
* [RFC 5656](https://tools.ietf.org/html/rfc5656) - Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer (December 2009)
* [RFC 6187](https://tools.ietf.org/html/rfc6187) - X.509v3 Certificates for Secure Shell Authentication (March 2011)
* [RFC 6239](https://tools.ietf.org/html/rfc6239) - Suite B Cryptographic Suites for Secure Shell (SSH) (May 2011)
* [RFC 6594](https://tools.ietf.org/html/rfc6594) - Use of the SHA-256 Algorithm with RSA, Digital Signature Algorithm (DSA), and Elliptic Curve DSA (ECDSA) in SSHFP Resource Records (April 2012)
* [RFC 6668](https://tools.ietf.org/html/rfc6668) - SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol (July 2012)
* [RFC 7479](https://tools.ietf.org/html/rfc7479) - [Ed25519](https://en.wikipedia.org/wiki/Ed25519) SSHFP Resource Records (March 2015)
* [RFC 5592](https://tools.ietf.org/html/rfc5592) - Secure Shell Transport Model for the Simple Network Management Protocol (SNMP) (June 2009)
* [RFC 6242](https://tools.ietf.org/html/rfc6242) - Using the NETCONF Protocol over Secure Shell (SSH) (June 2011)
* draft-gerhards-syslog-transport-ssh-00 - SSH transport mapping for SYSLOG (July 2006)
* draft-ietf-secsh-filexfer-13 - SSH File Transfer Protocol (July 2006)

In addition, the [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH) project includes several vendor protocol specifications/extensions:

* [OpenSSH PROTOCOL overview](http://cvsweb.openbsd.org/cgi-bin/cvsweb/~checkout~/src/usr.bin/ssh/PROTOCOL?content-type=text/plain)
* [OpenSSH certificate/key overview](http://cvsweb.openbsd.org/cgi-bin/cvsweb/~checkout~/src/usr.bin/ssh/PROTOCOL.certkeys?content-type=text/plain)

**See also**

* [Brute-force attack](https://en.wikipedia.org/wiki/Brute-force_attack)
* [Comparison of SSH clients](https://en.wikipedia.org/wiki/Comparison_of_SSH_clients)
* [Comparison of SSH servers](https://en.wikipedia.org/wiki/Comparison_of_SSH_servers)
* [Corkscrew](https://en.wikipedia.org/wiki/Corkscrew_(program))
* [Ident](https://en.wikipedia.org/wiki/Ident_protocol)
* [OpenSSH](https://en.wikipedia.org/wiki/OpenSSH)
* [Secure Shell tunneling](https://en.wikipedia.org/wiki/Secure_Shell_tunneling)
* [Web-based SSH](https://en.wikipedia.org/wiki/Web-based_SSH)

**References**

 *T. Ylonen; C. Lonvick (January 2006).* [*The Secure Shell (SSH) Protocol Architecture*](https://tools.ietf.org/html/rfc4251)*. Network Working Group of the IETF.* [*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.17487/RFC4251*](https://doi.org/10.17487%2FRFC4251)*.* [*RFC*](https://en.wikipedia.org/wiki/RFC_(identifier))[*4251*](https://tools.ietf.org/html/rfc4251)*.*

  Network Working Group of the IETF, January 2006, [RFC 4252](https://tools.ietf.org/html/rfc4252), The Secure Shell (SSH) Authentication Protocol

  [*"OpenSSH in Windows"*](https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_overview)*. Microsoft Docs. 7 January 2019.*

  [*"SSH Hardens the Secure Shell"*](http://www.serverwatch.com/news/print.php/3551081)*. Serverwatch.com.* [*Archived*](https://web.archive.org/web/20081223200033/http:/www.serverwatch.com/news/print.php/3551081) *from the original on 2008-12-23.*

  [*"Prying Eyes: Inside the NSA's War on Internet Security"*](http://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html)*.* [*Spiegel Online*](https://en.wikipedia.org/wiki/Spiegel_Online)*. December 28, 2014.* [*Archived*](https://web.archive.org/web/20150124202809/http:/www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html) *from the original on January 24, 2015.*

  *Seggelmann, R.; Tuxen, M.; Rathgeb, E.P. (18–20 July 2012). "SSH over SCTP — Optimizing a multi-channel protocol by adapting it to SCTP". Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on: 1–6.* [*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1109/CSNDSP.2012.6292659*](https://doi.org/10.1109%2FCSNDSP.2012.6292659)*.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*978-1-4577-1473-3*](https://en.wikipedia.org/wiki/Special:BookSources/978-1-4577-1473-3)*.*

  [*"Service Name and Transport Protocol Port Number Registry"*](https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml)*.*

  [*"How To Set Up Authorized Keys"*](http://wiki.qnap.com/wiki/How_To_Set_Up_Authorized_Keys)*.* [*Archived*](https://web.archive.org/web/20110510111514/http:/wiki.qnap.com/wiki/How_To_Set_Up_Authorized_Keys) *from the original on 2011-05-10.*

  [*"Service Name and Transport Protocol Port Number Registry"*](http://www.iana.org/assignments/port-numbers)*. iana.org.* [*Archived*](https://web.archive.org/web/20010604223215/http:/www.iana.org/assignments/port-numbers) *from the original on 2001-06-04.*

  [*"Download PuTTY - a free SSH and telnet client for Windows"*](http://www.putty.org)*. Putty.org.* [*Archived*](https://web.archive.org/web/20140527122917/http:/www.putty.org/) *from the original on 2014-05-27. Retrieved 2014-04-28.*

  [*"Cygwin Package List"*](https://cygwin.com/packages/package_list.html)*. Retrieved January 5, 2016.*

  [*"WinSCP home page"*](http://winscp.net/eng/index.php)*.* [*Archived*](https://web.archive.org/web/20140217163252/http:/winscp.net/eng/index.php) *from the original on 2014-02-17.*

  [*"WinSCP page for PortableApps.com"*](http://portableapps.com/apps/internet/winscp_portable)*.* [*Archived*](https://web.archive.org/web/20140216120049/http:/portableapps.com/apps/internet/winscp_portable) *from the original on 2014-02-16.*

  [*"PuTTY page for PortableApps.com"*](http://portableapps.com/apps/internet/putty_portable)*.* [*Archived*](https://web.archive.org/web/20140216214310/http:/portableapps.com/apps/internet/putty_portable) *from the original on 2014-02-16.*

  *Amies, A; Wu, C F; Wang, G C; Criveti, M (2012).* [*"Networking on the cloud"*](http://www.ibm.com/developerworks/cloud/library/cl-networkingtools/index.html)*. IBM developerWorks.* [*Archived*](https://web.archive.org/web/20130614123106/http:/www.ibm.com/developerworks/cloud/library/cl-networkingtools/index.html) *from the original on 2013-06-14.*

  *Tatu Ylönen.* [*"The new skeleton key: changing the locks in your network environment"*](https://www.scmagazineuk.com/the-new-skeleton-key-changing-the-locks-in-your-network-environment/article/545848/)*.* [*Archived*](https://web.archive.org/web/20170820162632/https:/www.scmagazineuk.com/the-new-skeleton-key-changing-the-locks-in-your-network-environment/article/545848/) *from the original on 2017-08-20.*

  [*Tatu Ylönen*](https://en.wikipedia.org/wiki/Tatu_Yl%C3%B6nen)*.* [*"SSH Port"*](https://www.ssh.com/ssh/port)*.* [*Archived*](https://web.archive.org/web/20170803235736/https:/www.ssh.com/ssh/port) *from the original on 2017-08-03.*

  *Nicholas Rosasco and David Larochelle.* [*"How and Why More Secure Technologies Succeed in Legacy Markets: Lessons from the Success of SSH"*](http://www.cs.virginia.edu/~drl7x/sshVsTelnetWeb3.pdf) *(PDF). Quoting* [*Barrett*](https://en.wikipedia.org/wiki/Daniel_J._Barrett) *and Silverman, SSH, the Secure Shell: The Definitive Guide, O'Reilly & Associates (2001). Dept. of Computer Science, Univ. of Virginia.* [*Archived*](https://web.archive.org/web/20060625065258/http:/www.cs.virginia.edu/~drl7x/sshVsTelnetWeb3.pdf) *(PDF) from the original on 2006-06-25. Retrieved 2006-05-19.*

  [*"Secsh Protocol Documents"*](http://www.vandyke.com/technology/drafts.html)*. VanDyke Software, Inc.* [*Archived*](https://web.archive.org/web/20100113104155/http:/vandyke.com/technology/drafts.html) *from the original on 2010-01-13.*

  [*"SSH Frequently Asked Questions"*](http://www.snailbook.com/faq/ssh-1-vs-2.auto.html)*.* [*Archived*](https://web.archive.org/web/20041010035705/http:/www.snailbook.com/faq/ssh-1-vs-2.auto.html) *from the original on 2004-10-10.*

  [*"libssh"*](https://www.libssh.org/2018/08/10/libssh-0-8-0/)*.*

  [*"A GNU implementation of the Secure Shell protocols"*](http://www.lysator.liu.se/~nisse/lsh/)*.* [*Archived*](https://web.archive.org/web/20120204035753/http:/www.lysator.liu.se/~nisse/lsh/) *from the original on 2012-02-04.*

  [*"Dropbear SSH"*](https://matt.ucc.asn.au/dropbear/dropbear.html)*.* [*Archived*](https://web.archive.org/web/20111014091250/http:/matt.ucc.asn.au/dropbear/dropbear.html) *from the original on 2011-10-14.*

  [*"RFC 4253"*](http://tools.ietf.org/html/rfc4253#section-5.1)*. Section 5. Compatibility With Old SSH Versions.* [*Archived*](https://web.archive.org/web/20100704150937/http:/tools.ietf.org/html/rfc4253#section-5.1) *from the original on 2010-07-04.*, IETF

  [*"OpenSSH: Project History and Credits"*](http://www.openssh.com/history.html)*. openssh.com. 2004-12-22.* [*Archived*](https://web.archive.org/web/20131224105341/http:/openssh.com/history.html) *from the original on 2013-12-24. Retrieved 2014-04-27.*

  [*"OSSH Information for VU#419241"*](https://www.kb.cert.org/vuls/id/MIMG-6L4LBL)*.* [*Archived*](https://web.archive.org/web/20070927231942/https:/www.kb.cert.org/vuls/id/MIMG-6L4LBL) *from the original on 2007-09-27.*

  *Sobell, Mark (2012). A Practical Guide to Linux Commands, Editors, and Shell Programming (3rd Edition). Upper Saddle River, NJ: Prentice Hall. pp. 702–704.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*978-0133085044*](https://en.wikipedia.org/wiki/Special:BookSources/978-0133085044)*.*

  [RFC 8709](https://tools.ietf.org/html/rfc8709)

  *Stebila, D.; Green J. (December 2009).* [*"RFC5656 - Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer"*](https://tools.ietf.org/html/rfc5656)*.* [*Archived*](https://web.archive.org/web/20120719131525/http:/tools.ietf.org/html/rfc5656) *from the original on 19 July 2012. Retrieved 12 November 2012.*

  *Miller, D.; Valchev, P. (September 3, 2007).* [*"The use of UMAC in the SSH Transport Layer Protocol / draft-miller-secsh-umac-00.txt"*](https://tools.ietf.org/html/draft-miller-secsh-umac-01)*.* [*Archived*](https://web.archive.org/web/20140819125608/https:/tools.ietf.org/html/draft-miller-secsh-umac-01) *from the original on 19 August 2014. Retrieved 12 November 2012.*

  [RFC 5647](https://tools.ietf.org/html/rfc5647)

  [*"SSH Insertion Attack"*](http://www.coresecurity.com/content/ssh-insertion-attack)*.* [*Core Security Technologies*](https://en.wikipedia.org/wiki/Core_Security_Technologies)*.* [*Archived*](https://web.archive.org/web/20110708192336/http:/www.coresecurity.com/content/ssh-insertion-attack) *from the original on 2011-07-08.*

  [*"Vulnerability Note VU#13877 - Weak CRC allows packet injection into SSH sessions encrypted with block ciphers"*](http://www.kb.cert.org/vuls/id/13877)*.* [*US CERT*](https://en.wikipedia.org/wiki/US_CERT)*.* [*Archived*](https://web.archive.org/web/20100710040357/http:/www.kb.cert.org/vuls/id/13877) *from the original on 2010-07-10.*

  [*"SSH CRC-32 Compensation Attack Detector Vulnerability"*](http://www.securityfocus.com/bid/2347/discuss)*.* [*SecurityFocus*](https://en.wikipedia.org/wiki/SecurityFocus)*.* [*Archived*](https://web.archive.org/web/20080725110345/http:/www.securityfocus.com/bid/2347/discuss) *from the original on 2008-07-25.*

  [*"Vulnerability Note VU#945216 - SSH CRC32 attack detection code contains remote integer overflow"*](http://www.kb.cert.org/vuls/id/945216)*. US CERT.* [*Archived*](https://web.archive.org/web/20051013074750/http:/www.kb.cert.org/vuls/id/945216) *from the original on 2005-10-13.*

  [*"Vulnerability Note VU#315308 - Weak CRC allows last block of IDEA-encrypted SSH packet to be changed without notice"*](http://www.kb.cert.org/vuls/id/315308)*. US CERT.* [*Archived*](https://web.archive.org/web/20100711103528/http:/www.kb.cert.org/vuls/id/315308) *from the original on 2010-07-11.*

  [*"Vulnerability Note VU#684820 - SSH-1 allows client authentication to be forwarded by a malicious server to another server"*](http://www.kb.cert.org/vuls/id/684820)*. US CERT.* [*Archived*](https://web.archive.org/web/20090901012536/http:/www.kb.cert.org/vuls/id/684820) *from the original on 2009-09-01.*

  [*"How to use SSH keys for authentication"*](https://upcloud.com/community/tutorials/use-ssh-keys-authentication/)*. Up Cloud. Retrieved 29 November 2019.*

  [*"Vulnerability Note VU#958563 - SSH CBC vulnerability"*](http://www.kb.cert.org/vuls/id/958563)*. US CERT.* [*Archived*](https://web.archive.org/web/20110622005639/http:/www.kb.cert.org/vuls/id/958563) *from the original on 2011-06-22.*

* 1.  Tatu Ylonen. "[BothanSpy & Gyrfalcon - Analysis of CIA hacking tools for SSH](https://www.ssh.com/ssh/cia-bothanspy-gyrfalcon)", ssh.com, 3 August 2017. Retrieved 15 july 2018.

**Further reading**

* [*Barrett, Daniel J.*](https://en.wikipedia.org/wiki/Daniel_J._Barrett)*; Silverman, Richard E.; Byrnes, Robert G. (2005). SSH: The Secure Shell (The Definitive Guide) (2nd ed.). O'Reilly.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*0-596-00895-3*](https://en.wikipedia.org/wiki/Special:BookSources/0-596-00895-3)*.*
* *Stahnke, Michael (2005). Pro OpenSSH. Apress.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*1-59059-476-2*](https://en.wikipedia.org/wiki/Special:BookSources/1-59059-476-2)*.*
* *Tatu Ylönen (12 July 1995).* [*"Announcement: Ssh (Secure Shell) Remote Login Program"*](https://groups.google.com/group/comp.security.unix/msg/67079d812a19f499?dmode=source&hl=en)*. comp.security.unix.* Original announcement of Ssh
* *Dwivedi, Himanshu (2003). Implementing SSH. Wiley.* [*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*978-0-471-45880-7*](https://en.wikipedia.org/wiki/Special:BookSources/978-0-471-45880-7)*.*
* This article is based on material taken from the [*Free On-line Dictionary of Computing*](https://en.wikipedia.org/wiki/Free_On-line_Dictionary_of_Computing) prior to 1 November 2008 and incorporated under the "relicensing" terms of the [GFDL](https://en.wikipedia.org/wiki/GNU_Free_Documentation_License), version 1.3 or later.

**External links**

|  |  |
| --- | --- |
|  | Wikimedia Commons has media related to [***SSH***](https://commons.wikimedia.org/wiki/Category:SSH). |

|  |  |
| --- | --- |
|  | Wikibooks has a book on the topic of: [***Internet Technologies/SSH***](https://en.wikibooks.org/wiki/Internet_Technologies/SSH) |

* [SSH Protocols](http://www.snailbook.com/protocols.html)
* [RFC7076](https://tools.ietf.org/html/rfc7076)
* [How to establish passwordless login with ssh](http://vfx.engineering/2014/05/11/infrastructure-how-to-establish-passwordless-login-with-ssh/)