SSH Academy

PuTTY Home - Free Downloads, Tutorials, and How-Tos

PuTTY is a versatile terminal program for Windows. It is the world's most popular free SSH client. It supports SSH, telnet, and raw socket connections with good terminal emulation. It supports public key authentication and Kerberos single-sign-on. It also includes command-line SFTP and SCP implementations.

Contents

PuTTY is most commonly used on Windows. It is also available on Linux.

- Download PuTTY for Windows
- PuTTY on Linux
- PuTTY on Mac

Alternative SSH clients

There many SSH clients that are more modern. A major shortcoming of PuTTY is that it does not have integrated file transfers in the client itself. Instead, file transfers have to be done via the command line. This is too complicated for most users. Tectia SSH has had them since 2000. PuTTY also does not include an SSH server.

- Other SSH clients

How to get an SSH server

PuTTY does not come with an SSH server. It can be used with Linux OpenSSH. For Windows and IBM z/OS mainframes, we recommend the Tectia SSH server.

Tutorials, how-tos, and user manual

- Installing PuTTY on Windows

- **Setting up SSH keys on Linux with PuTTYgen**
- **PuTTY user manual**

Features

- Windows client. Mac and Linux ports exist. No server included.
- Supports both 32-bit and 64-bit Windows. An MSI installer has been available since 2016.
- Supports SSH client, **telnet** client, **SFTP** client (command line only), and **rlogin** client. Both SSH2 and SSH1 protocols are supported. Note that use of SSH1 is not recommended for security reasons. Practically all devices support SSH2 these days.
- Supports **public key authentication** and **Active Directory**/Kerberos authentication.
- File transfers only using a separate command-line programs. No integrated file transfer support.
- No scripting support, but can be used together with **WinSCP**.

Terminal window

The main feature of the product is the terminal window. It has good terminal emulation, good configurability, and good support for different cryptographic algorithms. SSH, telnet, and plain TCP/IP protocols are supported.

The PuTTY terminal is pretty good and handles terminal emulation well.

Transferring files

The user interface does not include an integrated file transfer client. However, command-line tools called PSFTP and PSCP are provided. These can be used for file transfers. However, most non-technical users are not willing to use a command line. **Tectia SSH**, for example, has offered fully integrated file transfer capability since 2000.

The **WinSCP** and **FileZilla** clients can also be used for file transfers in conjunction with PuTTY. Having two software packages, switching between them to do operations, and managing profiles and logins for both is extra trouble. WinSCP can now import PuTTY profiles, but separate login is still required for each.

Public key authentication

PuTTY uses its own file format for SSH keys. The keys are stored in `.ppk` files. The **PuTTYgen** tool can be used for generating new keys and converting between `.ppk` files and other key

It is common for hackers and malware to collect SSH keys when penetrating an organization. This happened, for example, in the infamous **Sony breach**. Recently, Wikileaks obtained **CIA hacking tools** designed to steal SSH keys and their passphrases.

Managing SSH keys properly is important. **Universal SSH Key Manager** a popular SSH key management solution and the only one at the time of this writing that supports `.ppk` files.

Telnet support

PuTTY grew out of a **telnet** client. It still supports the `telnet` protocol. However, very few devices use `telnet` these days. Its use is not recommended for security reasons.

`Telnet` sends all user names and passwords in the clear. It is very easy to listen to network traffic and steal user names and passwords from `telnet` traffic. By mid-1990s, such password sniffing attacks had become the largest security problem on the Internet. That was the very problem SSH was designed to solve. Compromised routers, switches, or ARP proofing attacks can also be used to inject arbitrary commands into telnet sessions.

There is a separate version of the software, called `PuTTYtel`, for countries that do not allow any use of encryption. However, SSH is now used in all countries, officially or unofficially. Most systems can no longer be managed without encryption. Even the most oppressive countries need to secure their systems somehow. There cannot be cybersecurity in a networked environment without encryption.

PuTTY also supports connecting to serial ports and raw sockets. These can sometimes be useful for debugging purposes and for working with some legacy devices. For example, in kernel development access via a serial port is still sometimes the best way to debug a panic that causes an immediate reboot, as it provides a way to see the boot messages.

Known security vulnerabilities

Version 0.66 and earlier are known to contain security vulnerabilities. Upgrading to the latest version is recommended.

- **Buffer overflow in SCP**. This a potential stack overflow and remote code execution vulnerability. A corrupt server could execute code on the client when any file is downloaded. It could also be exploited by **man-in-the-middle attacks**.
- **Integer overflow in terminal escape sequence handling**. This is a memory corruption and possible remote code execution vulnerability. It involves sending an escape sequence to the terminal. For example, a compromised switch could inject the attack into a session. It can also be exploited by a corrupt server to execute code on the client, or using man-in-the-middle attacks.

Lack of proper key management can expose servers to risk and allow attackers to spread server-to-server or jump through desktops/laptops containing SSH keys. More information on SSH key management can be found **here**.

History and maintenance status

PuTTY is one of the oldest SSH clients for Windows. It was first released by Simon Tatham in 1998. SSH support was added in 2000.

After 19 years, the software is still a beta version. Development has been slow, but it is still being maintained. A recent version added support for elliptic curve cryptography. The user interface or features have not changed much in 15 years.

A Frequently Asked Questions document (FAQ) can be found **here**.

Where to find the source code

Source code is available on Simon Tatham's **home page**. Installation packages can be downloaded securely **here**.

Extensions, branches, and integrations

The product is open source. Several projects have branched off and build on its source code.

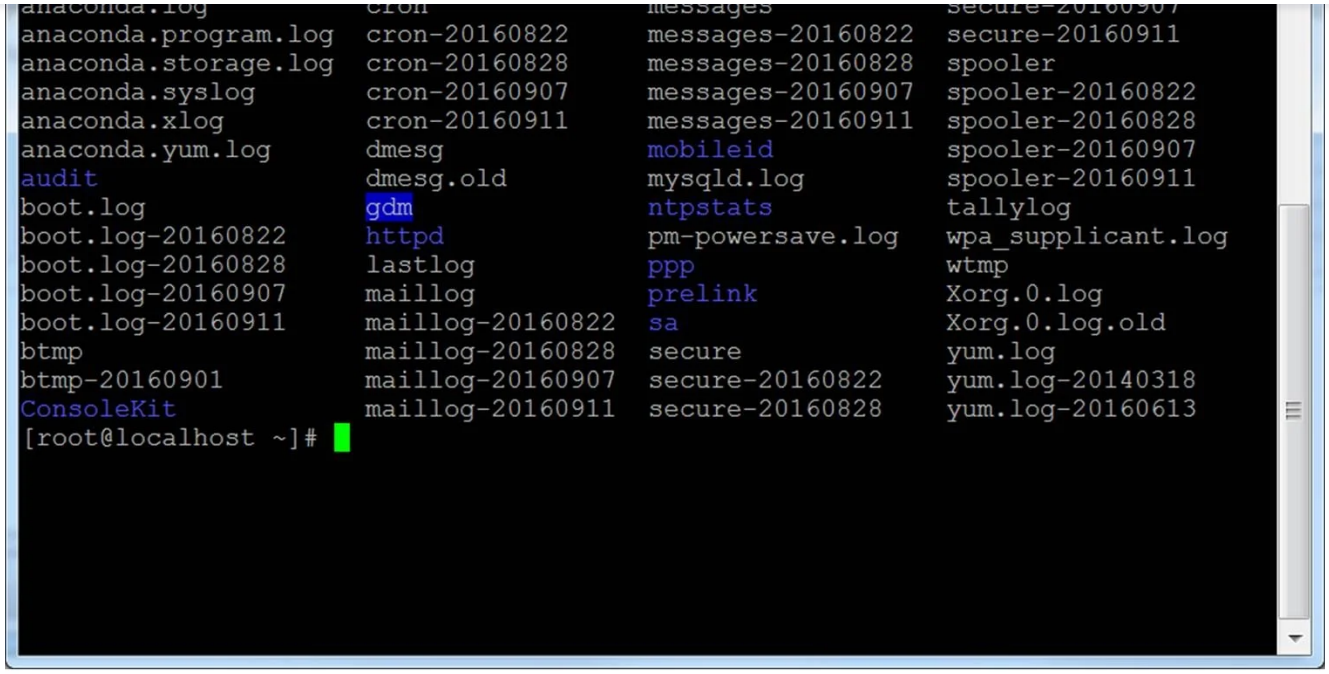
- **PuttyManager** is a tabbed user interface, but development appears to have stopped years ago.
- **ExtraPuTTY** is a fork that has various extensions, such as Lua programming language integration.
- **MTPuTTY** is a version of with a user interface that supports multiple tabs (i.e., a tab control where each tab is a terminal window).
- **WinSCP** has some level of integration for file transfer functionality.

Videos and screenshows

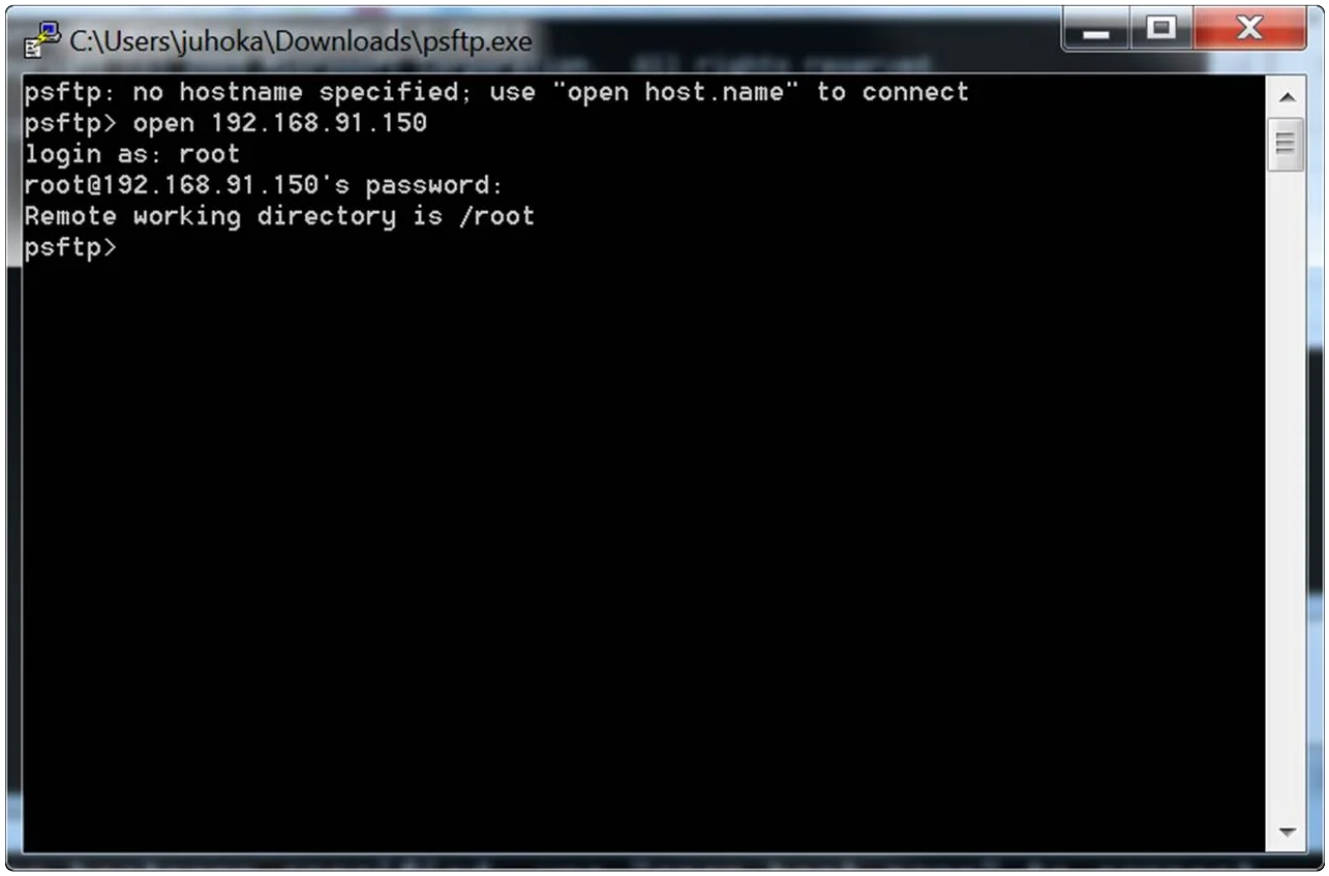
Tutorial video

PuTTY tutorial

Terminal window


anaconda.program.log cron-20160822 messages-20160822 secure-20160911
anaconda.storage.log cron-20160828 messages-20160828 spooler
anaconda.syslog cron-20160907 messages-20160907 spooler-20160822
anaconda.xlog cron-20160911 messages-20160911 spooler-20160828
anaconda.yum.log dmesg mobileid spooler-20160907
audit dmesg.old mysqld.log spooler-20160911
boot.log gdm ntpstats tallylog
boot.log-20160822 httpd pm-powersave.log wpa_supplicant.log
boot.log-20160828 lastlog ppp wtmp
boot.log-20160907 maillog prelink Xorg.0.log
boot.log-20160911 maillog-20160822 sa Xorg.0.log.old
btmp maillog-20160828 secure yum.log
btmp-20160901 maillog-20160907 secure-20160822 yum.log-20140318
ConsoleKit maillog-20160911 secure-20160828 yum.log-20160613
[root@localhost ~]#

SFTP client


C:\Users\juhoka\Downloads\psftp.exe
psftp: no hostname specified; use "open host.name" to connect
psftp> open 192.168.91.150
login as: root
root@192.168.91.150's password:
Remote working directory is /root
psftp>

We at SSH secure communications between systems, automated applications, and people. We strive to build future-proof and safe communications for businesses and organizations to grow safely in the digital world.

Solutions

Zero Trust Suite

Zero Trust Suite & Entra ID Integration

Quantum-Safe Cryptography (QSC)

Secure Collaboration 2024

Security Risk Mitigation

OT security

MSP Security

Just-in-Time Access

Secure vendor access

Hybrid cloud security

Credentials & Secrets Management

IT Audits & Compliance

Products

PrivX™ Hybrid PAM

UKM Zero Trust™

Tectia SSH Client/Server™

Tectia™ z/OS

SSH Secure Collaboration 2024

Secure Mail 2024

Secure Sign

NQX™ Quantum-Safe

Services

Support

Resources

Careers

References

Downloads

Manuals

Events & Webinars

Blog

Company

About us

Contact

Investors

Partners

Press

Stay on top of the latest in cybersecurity

Be the first to know about SSH's new solutions, product updates, new features, and other SSH news!