The SSH protocol

When discussing computer networks, the word “shell” refers to a program that provides an interface for accessing another operating system. With all the effort you put into keeping your own machine secure, you certainly want security when it’s connected to another machine. The Secure Shell network protocol, usually shorthanded to “SSH,” allows secure access to a computer over an unsecured network.

What is a protocol?

A protocol is a set of rules for how two things should communicate with each other. You may have heard the phrase “military protocol,” which refers to the strict guidelines that govern communications between members of the armed forces in all situations.

In the case of computer protocols, these are usually published as open standards so that any given protocol can be implemented in various products. Having these protocols readily available to everyone means that any machine or network that implements a given protocol should be able to communicate seamlessly with anything else that supports the same protocol.

For a deeper dive into Secure Shell, see SSH protocol.

The SSH protocol

So how does SSH secure the network? It works on the principle of public-key encryption. The client and the server each generate a strong encryption key for any data that is passed between them. Then, that key gets split in half, with the client retaining one portion and the server keeping the other. It’s a complex version of a simple idea, really; it’s not hard to imagine two people making up an encryption code and then tearing it in half for extra secrecy.

In SSH, the keys are split between a public key, the public half of the server’s encryption key, and the private key, which is stored only on the server. This way, a user’s machine can encrypt a message using the public key, but only the connected server can decode it because only the server’s private key will successfully decrypt the message. This way, if someone did intercept the network traffic, they still couldn’t read it because they don’t have the server’s private key. Using SSH, your keystrokes and the server’s responses are completely secure.

For more on these keys, see Public – private key pairs & how they work and A Deep Dive on End-to-End Encryption.

Using the SSH protocol

The SSH protocol is commonly used for logging in to servers remotely. While it is primarily used for logging in to Linux and Unix servers, it is also used to encrypt file transfers and to log in to some network equipment, like routers.

Of course, your private key should never be transmitted to anyone else or shared anywhere. Most SSH clients will not connect if your private key is not protected from other users. Because your private key is unique to you, it can serve as both authentication and encryption, so the server doesn’t need to ask you for a password.

Besides providing a secure login shell on a remote server, SSH can be used for a number of other functions, including:

- Transferring files between client and server with SCP (Secure Copy Protocol) or SFTP (Secure File Transfer Protocol); for more about these types of file transfers, see the Difference between SFTP and SCP.
- Forwarding network ports from server to client, or “tunneling”; for more on port forwarding, see How to Use SSH Port Forwarding.
- Relaying your login to yet another server behind a firewall, sometimes referred to as a “jump box” or “bastion host”; for more on this relaying method, see How to Set Up an SSH Jump Server.
- Running graphical user interface (GUI) applications on a server but displaying them on a local client; for more on this, see Use X forwarding on a personal computer.

Mark as completed

Like **Dislike** **Report an issue**