

Forums

Products

Download

Purchase

Contact

Home
About us
Products

sshlib

The 'log' utility
WinSSHD and Tunnelier
WinSSHD

Users' guide
Usage FAQ
Activation FAQ
History
Pricing
License

Tunnelier

Portable
FTP bridge
History
Pricing
License

Screenshots**Download**

Notifications

Security

Large scale

Purchase

Reseller list
Reseller policy
Affiliate program
Customers

How-tos and tutorials

About SSH
SSH Features
[How the internet works](#)
Using WinSSHD for SFTP
Port forwarding guide
Tunnel Remote Desktop
Tunnel Windows shares
Tunnel WinVNC
Tunnelier as service

Work at Bitvise
Contact

What you need to know about how the Internet works

Our products can be used in ways that don't require much knowledge about the internet. You can just type in the address of the server you're connecting to, open an SFTP window and start transferring files. However, if you will be using the more advanced features of our products, such as tunneling, you will need to understand the basics of how the Internet is structured. This guide is an attempt at relaying some of that understanding.

This guide is composed of the following sections:

- **IP addresses**
- **DNS names**
- **Subnets**
- **Types of IP addresses and subnets**
- **TCP and UDP**
- **Direction of TCP connections**
- **Ports**
- **Connecting to the internet from office**
- **Connecting to the internet from home**
- **Dynamic IP address issues**
- **Virtual servers - port forwarding at the router**
- **Firewalls**

IP addresses

Every computer connected to the internet has an Internet Protocol or IP address which identifies the computer on the internet. In the currently most widely used version of the Internet Protocol - version 4 - IP addresses are 4 bytes long and are expressed in the form nn.nn.nn.nn. Each nn is a number between 0 and 255.

When you connect to a web server to browse a web page, the DNS name of the web server, e.g. www.bitvise.com, is automatically translated by the software in your machine to an IP address in the nn.nn.nn.nn form. This address is then used to connect to the actual web server.

For example, the IP address of the server hosting fogbugz.bitvise.com at the time of this writing is 70.85.217.69. Our primary website, on the other hand, is hosted on several servers, and their IP addresses are 207.155.248.18, 207.155.248.31, 207.155.248.122 and 207.155.252.47.

In a Windows Command Prompt session, you can discover the IP addresses associated with DNS names using the nslookup command: e.g. 'nslookup www.bitvise.com'.

DNS names

IP addresses are difficult to remember, so the internet provides a translation service which translates memorable names into associated IP addresses. This facility is called the **Domain Name System** or DNS. You use DNS implicitly every time you type in an address such as 'www.bitvise.com' - your browser asks your operating system for translation into an IP address, and the operating system either returns a cached result, or inquires with a DNS server operated by your ISP. This server in turn either returns a cached result or inquires with another DNS server.

Subnets

No computer is directly connected to every other computer on the internet. Instead, each computer is a member of one or more subnets. Subnets, in turn, are connected to each other by machines called routers or gateways, which belong to multiple subnets, forwarding internet traffic from one subnet to the other and reverse.

In order to successfully communicate with other computers throughout the internet, your computer must know what subnet it is part of, so that it knows what IP addresses are outside your local subnet and must be relayed through the gateway. In addition, your computer must of course also know the IP address of the gateway.

Typically, a subnet is a group of consecutive IP addresses, such as all IP addresses from 11.22.33.0 to 11.22.33.255. This is commonly expressed in either of two formats:

- The subnet mask format. Here, the subnet is expressed as 11.22.33.0 with **subnet mask** 255.255.255.0. The subnet mask indicates what bits of the subnet IP address indicate the actual subnet, and what bits are variable, indicating individual computers in the subnet. A byte consists of 8 bits, and 255 is 1111 1111 in binary. Therefore, 255.255.255.0 means that the first 3 bytes of the subnet IP address (11.22.33) indicate the actual subnet, and the last byte can be variable (and indicates computers

What is SSH?

The protocol implemented by our products is SSH Secure Shell version 2, or SSH2. Click [here](#) to learn more about it.

How-Tos and Tutorials

How the internet works
Using WinSSHD for SFTP
Port forwarding guide
Tunnel Remote Desktop
Tunnel Windows shares
Tunnel WinVNC

Try out WinSSHD!

WinSSHD is an advanced **SSH server for Windows** with secure remote access, file transfer and tunneling features. Supports public key authentication, SCP, SFTP. Its advantages include **speed**, **reliability**, **ease of use**, **configurability**, and **bvterm** - the best remote console on the Windows platform. Try it free for up to 30 days!

Try out Tunnelier!

Tunnelier is a friendly and flexible **SSH client for Windows** which includes state of the art **terminal emulation**, graphical as well as command-line **SFTP support**, an **FTP-to-SFTP bridge**, powerful **tunneling** features, and also remote administration for WinSSHD. **Free for individual use!**

in the subnet). If the subnet mask were 255.255.0.0, that would mean that the last two bytes are variable.

- The significant bits format. Here, the subnet is expressed as 11.22.33.0/24, which means subnet 11.22.33.0 with 24 **significant bits**. The 24 means that the first 24 bits of the subnet mask are 1, and all the following bits are 0. Thus, /24 is equivalent to a subnet mask of 255.255.255.0. /16 is equivalent to a subnet mask of 255.255.0.0. And because there are just 32 bits in an IP address, /32 indicates an IP address with no variable part: a fixed, constant IP address.

Our software expresses subnets using the significant bits format.

Types of IP addresses and subnets

There are three major types of IP addresses (or subnets) that you need to be aware of.

- **Public IP addresses.** Most IP addresses in the 32-bit address range have the purpose of uniquely identifying a computer on the internet. The IP address 207.155.248.18, for example, is a public IP address that uniquely identifies one of the servers hosting the www.bitvise.com website (and others). This is the type of IP address through which a server must be reachable in order to be accessible to computers throughout the internet.
- **Private subnets.** Special ranges of the 32-bit IP address range have been set aside for use in private networks, where the computers in such a network do not need to be directly accessible from the internet as servers (but may nevertheless access the internet through a gateway, as clients). These ranges include:
 - 10.0.0.0/8 (addresses from 10.0.0.0 to 10.255.255.255)
 - 172.16.0.0/12 (addresses from 172.16.0.0 to 172.31.255.255)
 - 192.168.0.0/16 (addresses from 192.168.0.0 to 192.168.255.255)
- **Special IP ranges.** There are several special purpose IP ranges, but the one you need to know about is 127.0.0.0/8 (addresses from 127.0.0.0 to 127.255.255.255). This is the **local loopback** range and is used to connect two programs running on the same machine. Any address in this range can be used for this kind of purpose, but the most commonly used are 127.0.0.1 and 127.0.0.2. The special DNS name 'localhost' translates to 127.0.0.1.

TCP and UDP

The Internet Protocol itself is a relatively rudimentary protocol which provides only the capability of delivering small chunks of data to other computers. The Internet Protocol does not provide reliability: chunks of data that are sent using the Internet Protocol may be lost. They also may arrive in an order different to the order in which the chunks were sent.

For some types of data transfer, the (un)reliability afforded by the Internet Protocol is fine. When streaming video, for example, it does not matter if chunks that make up intermediate frames of the video are lost. What matters is that most of the data arrives relatively quickly, allowing the video to be played with reasonable quality and on the fly. The **User Datagram Protocol**, or UDP, is a simple protocol layered on top of the Internet Protocol that provides this level of reliability. UDP is used for purposes such as relaying video and audio streams as well as for networked games; all environments where responsiveness and fast delivery is more important than perfect reliability.

For other types of data transfer, however, this level of reliability is not enough. When transferring a file, for example, you want to transfer all of its contents in perfect order and integrity; you don't want any chunks of it to accidentally be lost. When accessing a web page, likewise, you want all the text to be transferred without error. Data transfers that require this higher level of reliability use the **Transmission Control Protocol**, or TCP. Like UDP, TCP is a protocol layered on top of the Internet Protocol, but it is more complex than UDP: it contains mechanisms to ensure that data is received in order and that, if any chunks are lost, they are resent. The reliability provided by TCP has costs in terms of responsiveness. Before any data can be sent using TCP, the two computers must engage in a short back-to-forth to establish a TCP connection. If any data are lost during transmission, delivery of subsequent data awaits until the data that were lost are retransmitted and delivered. When there is a high rate of data loss on a connection, this may cause transmission to be jerky.

The majority of widely known protocols used on the internet are layered on top of TCP. These include:

- the Simple Mail Transfer Protocol (SMTP), used for email delivery;
- the Post Office Protocol (POP) and IMAP, used for email retrieval;
- the Hypertext Transfer Protocol (HTTP), used for accessing websites;
- as well as, of course, the Secure Shell protocol (SSH), which our products are about.

Direction of TCP connections

TCP connections are like phone calls: they are always initiated by one party and accepted (or not) by the other. The computer that originates the TCP connection is usually the client, and the computer that accepts it is usually the server. Sometimes, notably in the FTP protocol, a secondary TCP connection will be established in the reverse direction, from the server to the client. But, in protocols other than FTP, connections are almost always initiated by the client.

Regardless of the direction in which a TCP connection is established, data can always flow both ways. However, the direction of the TCP connection matters because it determines

who the initiating party is, and is also used by network components to impose rules on whether a connection can be established.

Ports

In order to handle multiple simultaneous connections with the same computer, your computer must be able to distinguish them. To do so, each connection is assigned two port numbers, one at each end point of the connection. A connection is then uniquely identified with four pieces of information: (1) local address, (2) local port, (3) remote address, (4) remote port. Valid port numbers are between 1 and 65535. The party that originates a TCP connection usually selects a local port number at random. On the other hand, the port number of the party that accepts the connection must be known in advance by the party that originates the connection. You can confirm this by executing 'netstat -n' from a Windows Command Prompt just after loading a web page in your browser.

For example, this excerpt from 'netstat -n' output was taken just after opening www.bitvise.com in a browser.

Proto	Local Address	Foreign Address	State
TCP	10.10.10.123:21681	207.155.248.122:80	ESTABLISHED

The above output indicates an established TCP connection with local address 10.10.10.123, local port 21681, remote address 207.155.248.122 and remote port 80. The connection was initiated by the local machine, therefore the local port number 21681 was randomly selected, whereas the remote port number 80 is the well-known **HTTP** port. This is the port where the vast majority of web servers accept connection, so even when access to other ports is blocked, connections to port 80 will very likely be permitted.

Other well-known destination ports are:

- 21 - FTP (control connection)
- 22 - SSH
- 23 - Telnet
- 25 - SMTP
- 80 - **HTTP**
- 110 - POP3
- 143 - IMAP4
- 443 - **HTTPS** (**HTTP** over TLS or SSL)
- 1080 - SOCKS proxy

On Windows, a more exhaustive list of well-known ports can be found in the file `\Windows\System32\Drivers\etc\services` (open it with Notepad).

Connecting to the internet from office

In an office environment, your computer will most likely be connected to a subnet in one of the private address ranges. This means that your computer will have an IP address not unique throughout the internet, so it cannot communicate with other computers on the internet directly. However, the network administrators at your office have most likely applied one of the following solutions to allow you to access the internet.

- **Network address translation (NAT).** In this setup, your computer directs all traffic destined to the internet through a gateway in your local subnet. This gateway has a public IP address which is unique and can be used for internet addressing. The gateway substitutes its own IP address and port in place of your computer's. When chunks of data arrive in reply, the gateway knows from the port number in the data that they must be forwarded to your computer and local port.

In this setup, your computer is led to believe that it is present on the internet with its private subnet IP address; but it isn't. The gateway is present on the internet and represents all computers in the subnet with its own public IP. All connections initiated to the internet by computers on the subnet appear to outside observers as coming from the gateway's public IP address.

- **Proxy.** In this setup, your computer cannot initiate connections to the internet directly. Instead, applications on your computer must contact one of several types of proxy servers residing on your local subnet, and ask the proxy server if it would kindly relay a connection to the outside. This is conceptually similar to NAT. However, whereas NAT works for all applications on your machine and requires from them no special awareness, the proxy setup works only with those applications which can connect to the internet through the proxy. The proxy setup also affords administrators more control: they can more easily restrict and monitor your traffic and permit or deny access selectively based not just on port numbers, but the content being accessed and protocols being used.

There is also a number of office environments where each computer has a separate, own public IP address. These are simple and involve no NAT or proxy servers as outlined above.

Connecting to the internet from home

From home, you usually connect to the internet through a modem - whether it is phone, cable, ISDN or DSL. In any case, you can either hook the modem directly to your computer; or, if you have multiple computers, you can buy a router, connect the router to your modem and your computers to the router.

- If you use a router, the machines connected to it are assigned addresses in a private subnet, and the router performs Network Address Translation to allow your machines

to access the internet.

- If you connect the modem to your machine directly, the computer gets a public IP address directly accessible from the internet. If you then connect other machines to this machine (through a second network interface), those machines are joined to a private subnet. The directly connected machine then performs Network Address Translation to allow the other computers to access the internet.

In most cases, you will be provided a single public IP address by your internet provider. Sometimes this IP address will be fixed; this is called a **static IP address**. In other situations, the IP address will periodically change; this is called a **dynamic IP address**. With dial-up modems, you will get a different public IP address every time you dial up. With DSL and cable modems, your IP address may change at a predefined time every day or night.

Dynamic IP address issues

The following issues correspond with a continuously changing IP address.

Whenever your public IP address changes, all ongoing TCP connections to and from your machine are terminated and must be reestablished using the new IP.

Since the IP address of your computer is unpredictable, it is difficult for others to connect to it. If you want to host any kind of network-accessible service on your machine, you need to either use a dynamic DNS service; this works by allocating you a DNS name which is regularly updated to reflect your changing IP address; or you need to implement a more pedestrian solution, such as configuring a program on your computer to periodically connect to another server and store your current IP address there, making it available for retrieval.

If you want to host a service on your home machine and find that your IP address changes periodically, the best way around this problem is to ask your ISP to grant you a static IP. They will frequently agree to do this free of charge. If this is unavailable, you can use a dynamic DNS service.

Virtual servers - port forwarding at the router

If you want to make a server accessible from the internet, but the computer on which the server will be based has only a private subnet IP addresses, there is a solution. Usually, the router which connects the private subnet to the internet can be configured to forward all incoming connections on a certain port to one of the computers inside the private network. This is called port forwarding (not the same thing as SSH port forwarding) or a 'virtual server' facility (although the server is quite real; it's just its IP address that is not).

This setup generally works just fine, but there is one thing to remember. The IP address by which the server is known to internet clients is not the IP address that the server machine actually has. This distinction between the public IP address at the router, and the private IP address of the actual server machine inside, frequently arises in SSH connection tunneling, leading to incorrect configuration if not properly understood.

Firewalls

Modern computers run a large number of local services (such as Windows file and printer sharing) which accept connections on various port numbers, but are meant to be accessible only from locally trusted subnets. Preventing the wider internet from accessing these services in possibly malicious ways is the purpose of **ingress firewalls**.

In organizations, gateways that connect the local subnet to the internet usually feature an ingress firewall. This firewall should normally be configured to allow no connections into the subnet, except connections to servers that must accept connections from the internet.

At home, your ISP will usually **not** protect your PC from malicious access from the internet. Instead, this task must be performed by a firewall installed on your home router, or if your computer is connected to the internet directly, a software firewall in your machine. Windows XP comes equipped with such a firewall; you should use it. Software firewall solutions are available for earlier versions of Windows.

There is another type of firewall called an **egress firewall**, or a firewall that filters outbound connections from your machine to the internet. This is generally software which tries to control what programs on your machine access the internet. This is intended to block malicious software from doing too much damage after it has already infected your computer. However, cleverly written malware can fool an egress firewall like this with fairly simple and straightforward deceptions. The only real medicine against malware is therefore to prevent it from infecting your computer in the first place.