NỘI DUNG MÔN

An toàn trên đường truyền
Wireless Security
Dò quét và liệt kê
An toàn dịch vụ ở xa
TCP reverse
Tấn công APT
PSAD, PFSense

AN TOÀN TRÊN ĐƯỜNG TRUYỀN

TÀI LIỆU THAM KHẢO

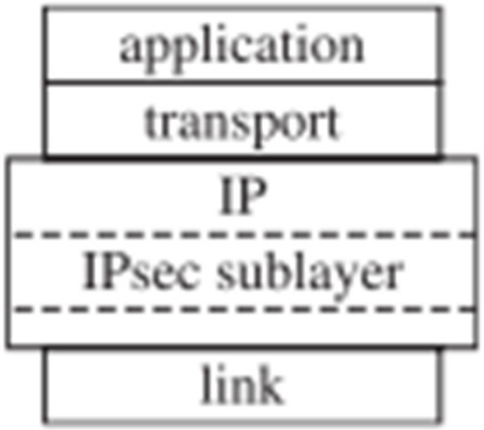
- Andrew Lockhart, Network Security Hacks, 2ed
- Eric Cole, Network Security Fundamentals
- Daniel J. Barrett, Richard E. Silverman, SSH, the Secure Shell: The Definitive Guide

CONTENTS

- IP SECURITY (IPsec)
- SSH
- SSL & TLS
- VPN

IP security: Overview (1/3)

- IPsec is a security protocol that operates at the Internet layer of the TCP/IP protocol stack.
- IPsec is optional with IPv4 and is not implemented by all operating systems. IPsec is required by the IPv6 specification.



IPsec on the stack.

IP security: Overview (2/3)

- IPsec can be used to secure traffic on a LAN or on a VPN. IPsec can be configured to offer the following:
  - Confidentiality
  - Authentication
  - Data integrity
  - Packet filtering
  - Protection against data reply attacks
- IPsec can be configured to use multiple security algorithm options. An administrator can decide which security algorithm to use for an application based on security requirements.

IP security: Overview (3/3)

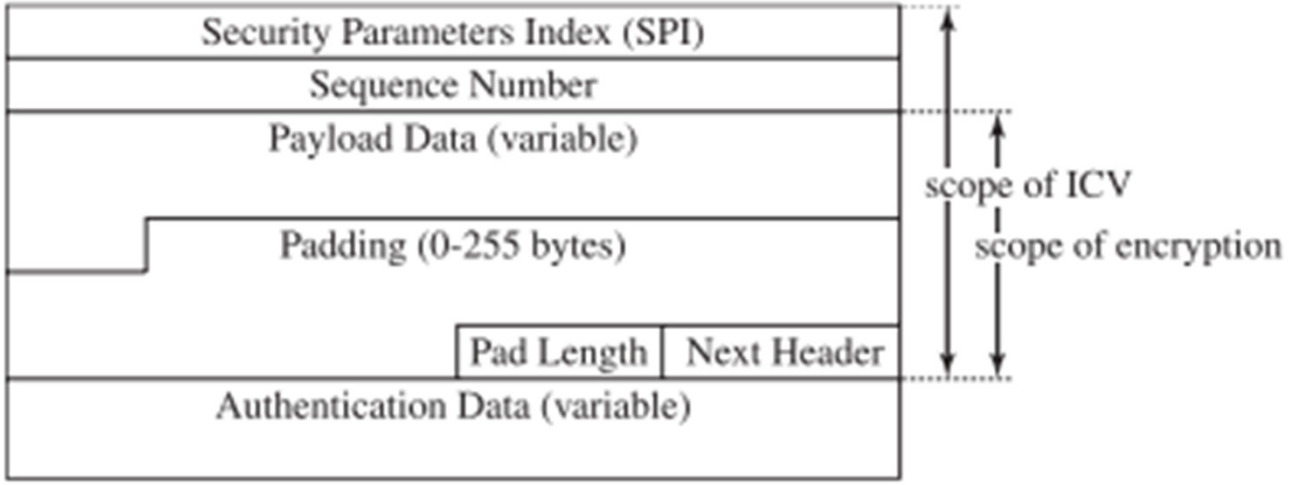
- IPsec architecture is described in RFC 2401. IPsec includes two major security mechanisms: Authentication Header (AH), described in RFC 2402, andn Encapsulating Security Payload (ESP), covered in RFC 2406.

IP security: Authentication Header

- AH protects the integrity and authenticity of IP packets but does not protect confidentiality.

IP security: Encapsulating Security Payload (ESP)

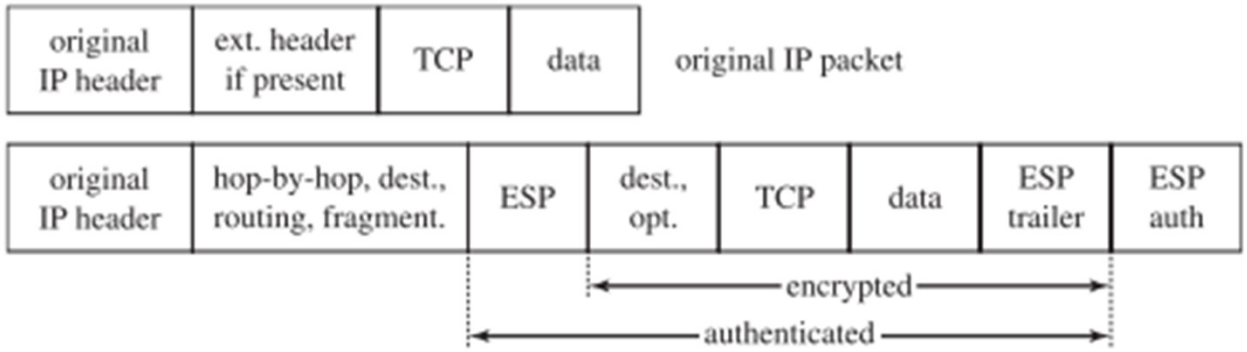
- ESP can be used to provide confidentiality, data origin authentication, data integrity, some replay protection, and limited traffic flow confidentiality



ESP packet.

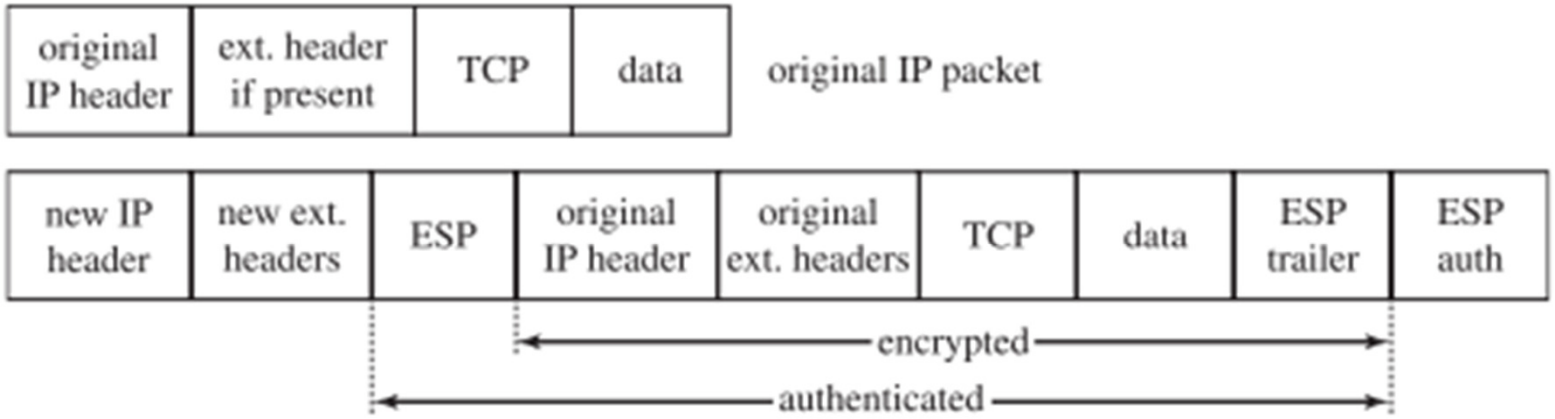
ESP Modes (1/2)

- **Transport mode**: the upper-layer protocol frame is encapsulated.The IP header is not encrypted. Transport mode provides end-to-end protection of packets exchanged between two end hosts. Both nodes have to be IPsec aware



ESP Modes (2/2)

- **Tunnel mode**: an entire datagram plus security fields are treated as a new payload of an outer IP datagram. The original inner IP datagram is encapsulated within the outer IP datagram
  - This mode can be used when IPsec processing is performed at security gateways on behalf of end hosts. The end hosts need not be IPsec aware.
  - The gateway could be a perimeter firewall or a router. This mode provides gateway-to-gateway security rather than end-to-end security.
  - On the other hand, you get traffic flow confidentiality as the inner IP datagram is not visible to intermediate routers, and the original source and destination addresses are hidden.


original IP header
ext. header if present
TCP
data
original IP packet
new IP header
new ext. headers
ESP
original IP header
original ext. headers
TCP
data
ESP trailer
ESP auth
encrypted
authenticated

Tunnel mode.

IP security: Security Associations (SA)

- To generate, decrypt, or verify an ESP packet a system has to know which algorithm and which key to use. This information is stored in a security association (SA)
- The SA is the common state between two hosts for communication in one direction. Bidirectional communication between two hosts requires two security associations, one in each direction. Therefore, SAs are usually created in pairs.
- An SA is uniquely identified by an SPI (carried in AH and ESP headers), the destination IP address, and a security protocol (AH or ESP) identifier. It contains the relevant cryptographic data, such as algorithm identifiers, keys, and key life times. There can be a sequence number counter and an anti-replay window. The SA also tells whether tunnel mode or transport mode is used.

IP security: Internet Key Exchange Protocol (IKE)

- Number of nodes is small: SA could be created manually.
- The alternative to manual keying is IKE (for lagre networks) IKEv1(RFC 2409), IKEv2(RFC 4306)
- Two goals of IKE: entity authentication and the establishment of a fresh shared secret.
- IKE operates in two phases: Phase 1 sets up an SA as a secure channel to carry further SA negotiation. In phase 2, SAs for general use are negotiated; multiple pairs of SAs can be negotiated during each phase 2 negotiation.

Set up IPsec under Linux

- The most popular way of configuring IPsec connections under Linux is by using the Openswan (http://www.openswan.org) package
- Openswan is made up of two components: pluto and, optionally, KerneL IP Security (KLIPS)
- Linux kernel includes support for IPsec, but KLIPS can be used instead for some additional features.
- pluto is the user-land daemon that controls Internet Key Exchange (IKE) negotiation

Set up IPsec…

- Download and install

```
$ tar xfz openswan-2.4.6rc3.tar.gz
$ cd openswan-2.4.6rc3
$ make programs
```

- Use KLIPS instead of native IPsec support in the kernel, download the appropriate patch from the Openswan download page. Apply the patch to your kernel source with the following commands:

```
# cd /usr/src/kernels/linux-2.6.14.6
# zcat /tmp/openswan-2.4.6rc3.kernel-2.6-klips.patch.gz | patch -p1
```

Set up IPsec…

- If patched kernel for KLIPS, rebuild it and reboot with it.
- If chose to use the kernel's built-in IPsec support, can go ahead and start it now:

```
# /etc/init.d/ipsec start
```

- Verify that your system settings are configured correctly to use IPsec:

```
# /usr/local/sbin/ipsec verify
```

Configuring Openswan

- Openswan's configuration is controlled by two configuration files:

  /etc/ipsec.conf and /etc/ipsec.secrets.

- The ipsec.conf file breaks a VPN connection into right and left segments.
- This is merely a logical division. The segment on the left can be either the internal or the external network; this allows the same configuration file to be used for both ends of a VPN network-to-network tunnel.

Example

- Adding an entry like this to *ipsec.conf* creates an encrypted tunnel between two hosts:

```
conn host-to-host
    left=192.168.0.64
    leftid=@colossus.nnc
    #leftnexthop=%defaultroute
    right=192.168.0.62
    rightid=@spek.nnc
    #rightnexthop=%defaultroute
    auto=add
```

- For authentication purposes, this connection uses RSA signatures, which are obtained by running */usr/local/sbin/ipsec showhostkey* on both hosts

```
# /usr/local/sbin/ipsec showhostkey --left
        # RSA 2192 bits   colossus.nnc   Thu Jul 13 20:48:58 2006
        leftrsasigkey=0sAQNpOndA2SO5aQnEmxqlM5c3JerA9cMwGB0wPE9PshVFBgY44
Ml8Lw7usdMzZTMNaSeXu3+80fK7aXWqBGVXWpIEw2EAFlGcbg1mrEoAVpLwbpM7ZmZPr6Cl0AdFyTF
xFK4k52y702h6xsdSoeTWabs2vkzPLDR8QqvlzIzPkDHE+MQG4q/F+fVUkn/TNeGL7axxfVkepqTHI1nwb
NsLdPXdWGKL9c28ho8TTSgmVMgr9jVLYMNwWjN/BgKMF5J/glALr6kjy19uNEpPFpcq9d0onjTMOts1xyfj
0bst2+IMufX21ePuCRDkWuYsfcTMlo7o7Cu+alW0AP4mZHz8Ze8PzRm9h3oGrUMmwCoLWzMeruud
```

- Note: replacing --left with –right in the right host
- Paste the output into configuration file

- Copy the configuration file to both hosts and restart the ipsec service on both systems:

```
# /etc/init.d/ipsec restart
ipsec_setup: Stopping Openswan IPsec...
ipsec_setup: Starting Openswan IPsec 2.4.6rc3...
ipsec_setup: insmod /lib/modules/2.6.16-
1.2115_FC4/kernel/net/key/af_key.ko
ipsec_setup: insmod /lib/modules/2.6.16-
1.2115_FC4/kernel/net/ipv4/xfrm4_
tunnel.ko
```

- To create the IPsec connection by running the following command on one of the hosts:

```
# /usr/local/sbin/ipsec auto --up host-to-host
```

- To test out the connection, ping one of the hosts in the tunnel from the other one:

```
$ ping spek.nnc

$ ping colossus.nnc
```

Set Up IPsec under FreeBSD & OpenBSD

- Set Up IPsec Under FreeBSD
  - Use FreeBSD's built-in IPsec support to secure traffic.
  - Requires enabling IPsec in the kernel and installing a user-land program, racoon, to handle the IKE negotiations
- Set Up IPsec in OpenBSD
  - Use IPsec the OpenBSD way, it's compiled into the kernel that ships with each release and is enabled by default
  - Create the appropriate /etc/isakmpd/isakmpd.conf and /etc/isakmpd/isakmpd.policy files and start isakmpd (the IPsec key-management daemon)

Configuring IPsec on a Windows Network

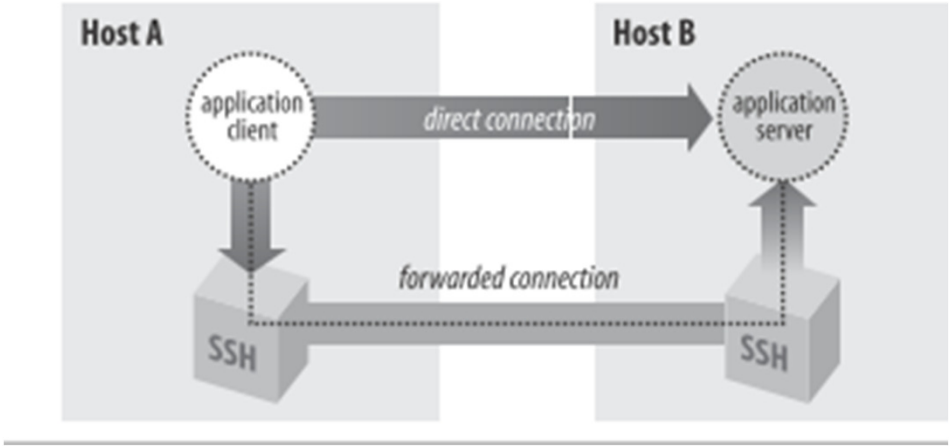
- Can enable and configure the IPsec protocol with Group Policy for Windows or through the Network Connection Wizard.
- Can configure rules that a computer will follow in applying IPsec to outgoing and incoming packets
- Exercise: Manage IPsec feature in your computer?

Encrypt Traffic Automatically with Openswan

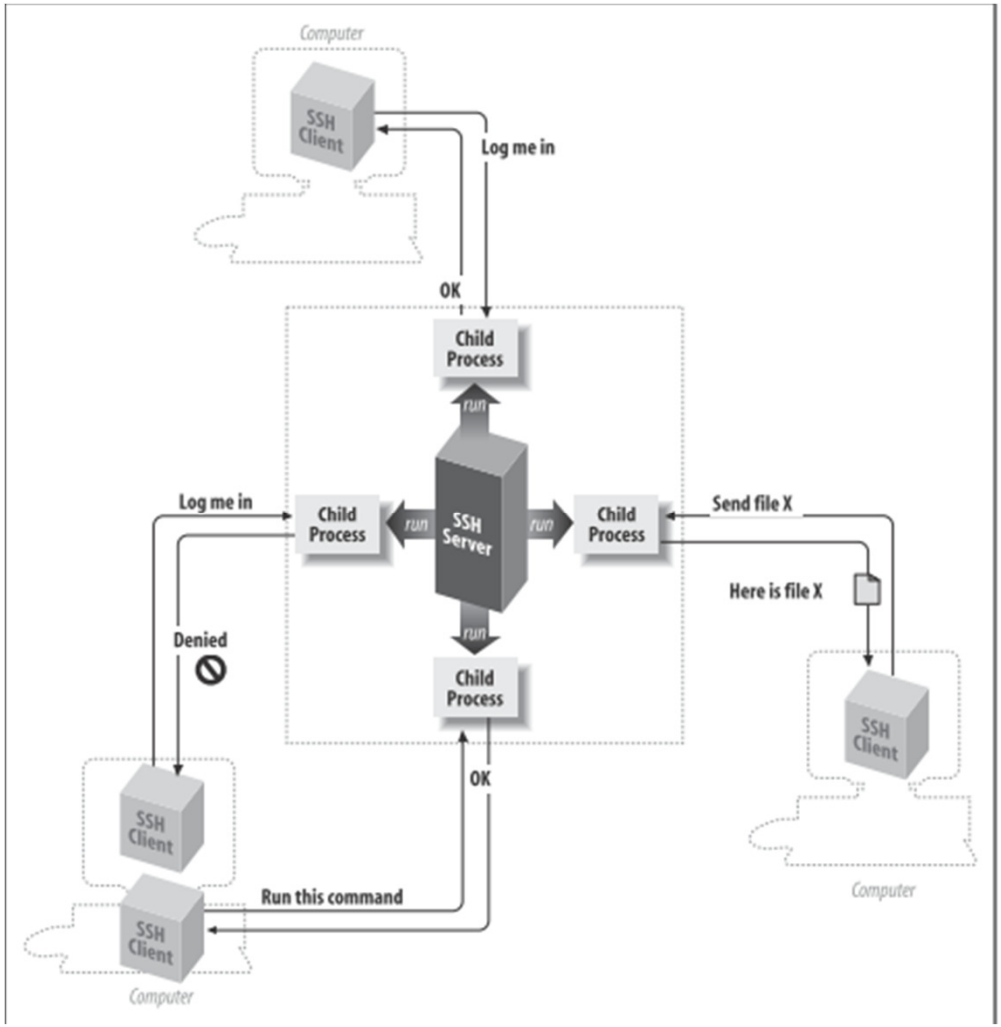
- Opportunistic encryption: Openswan transparently encrypts traffic between all hosts
- Each host must have a public key generated to use with Openswan (stored in a DNS TXT record)
- A host wants to initiate an encrypted connection with another host, it looks up the host's public key through DNS and uses it to initiate the connection

SSH

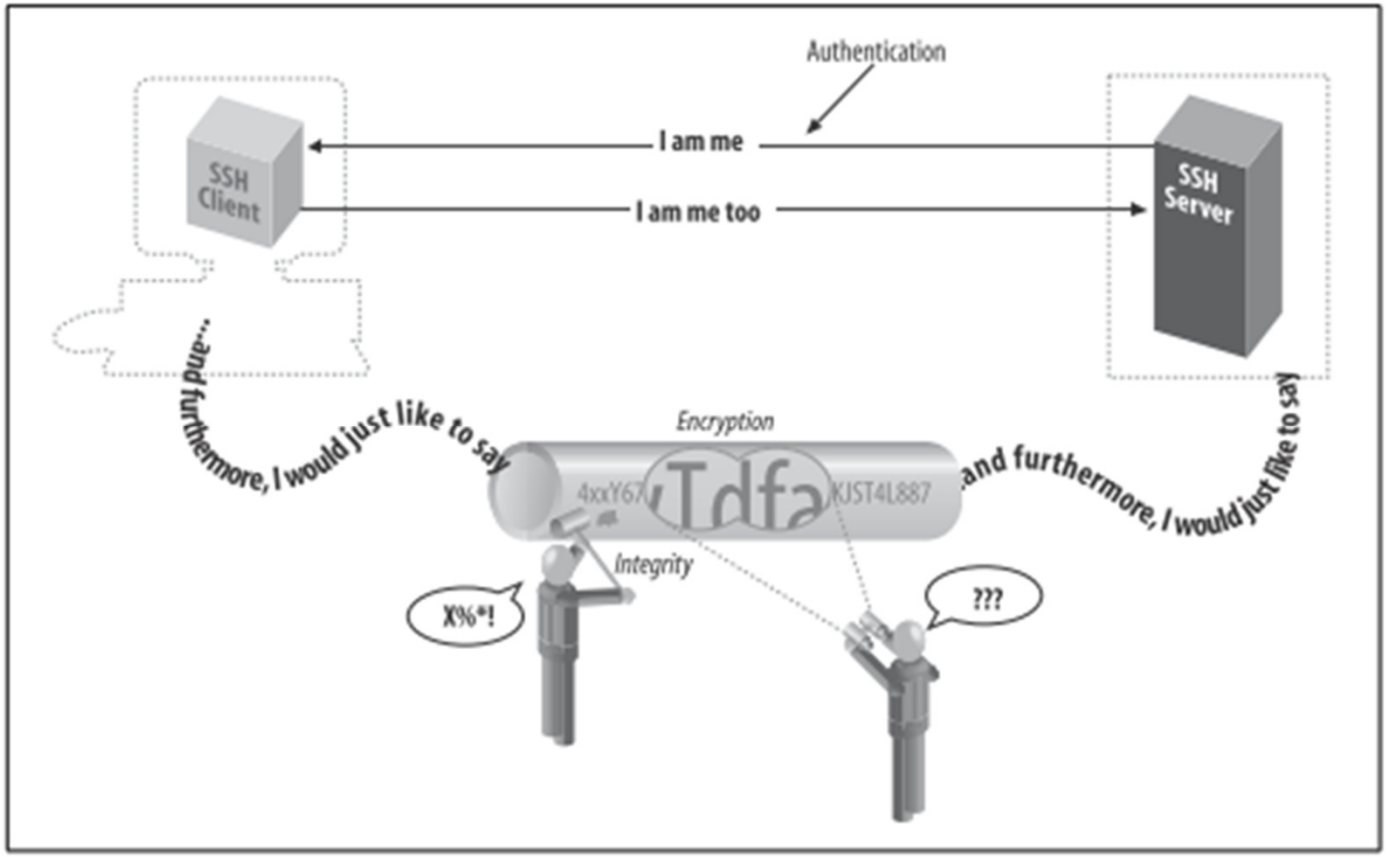
- SSH creates a channel for running a shell on a remote computer, with end-to-end encryption between the two systems
- Forwarding: whenever data is sent to the network, SSH automatically encrypts and decrypts it (transparent encryption)
- SSH were developed in 1995 by Tatu Ylönen, a researcher at the Helsinki University of Technology in Finland



- SSH has a client/server architecture



- SSH is a protocol, SSH protocol covers authentication, encryption, and the integrity of data transmitted over a network



Protocols, Products, Clients

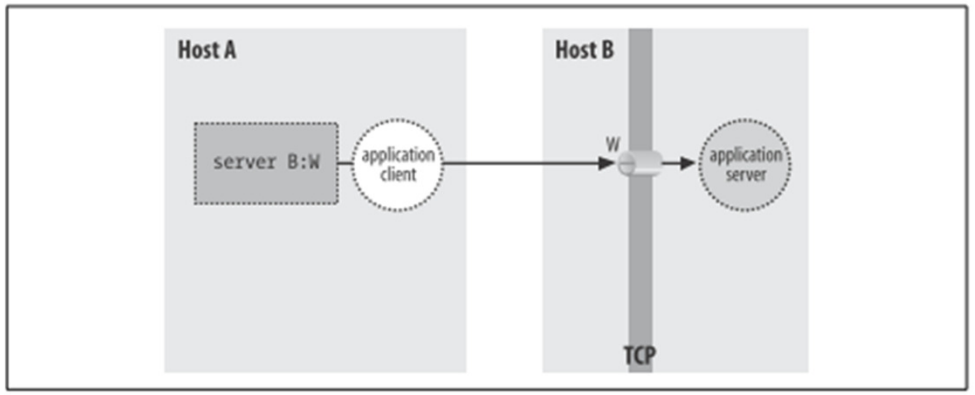
- Protocols are denoted with dashes: SSH-1, SSH-2.
- Products : OpenSSH, Tectia, PuTTY, etc.
- Client programs : ssh, scp, putty, etc.

Overview of SSH Features

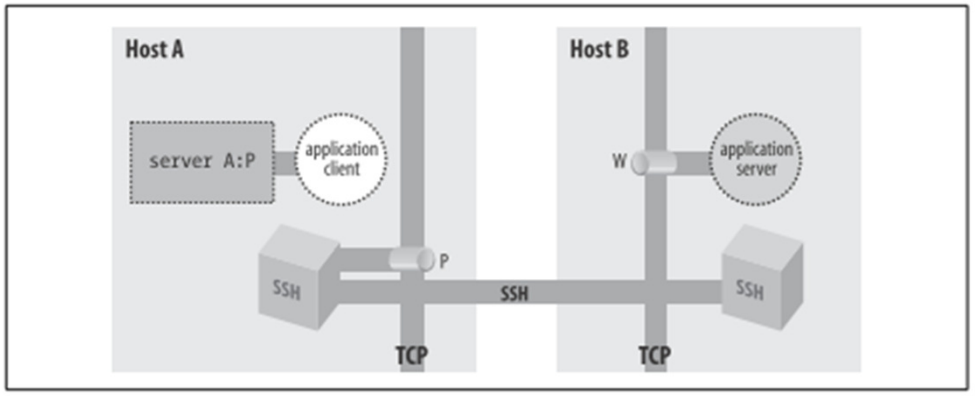
- Secure Remote Logins
- Secure File Transfer
- Secure Remote Command Execution
- Keys and Agents
- Access Control
- Port Forwarding

Port Forwarding

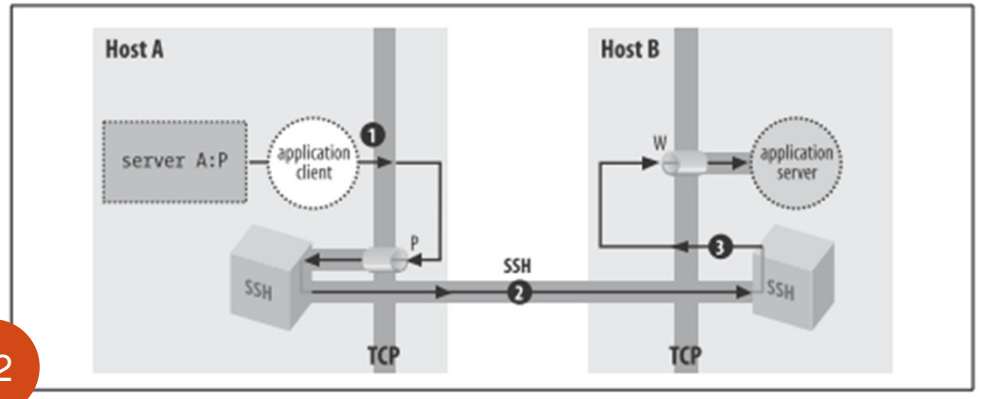
- SSH uses TCP/IP as its transport mechanism, usually TCP port 22 on the server machine, as it encrypts and decrypts the traffic passing over the connection



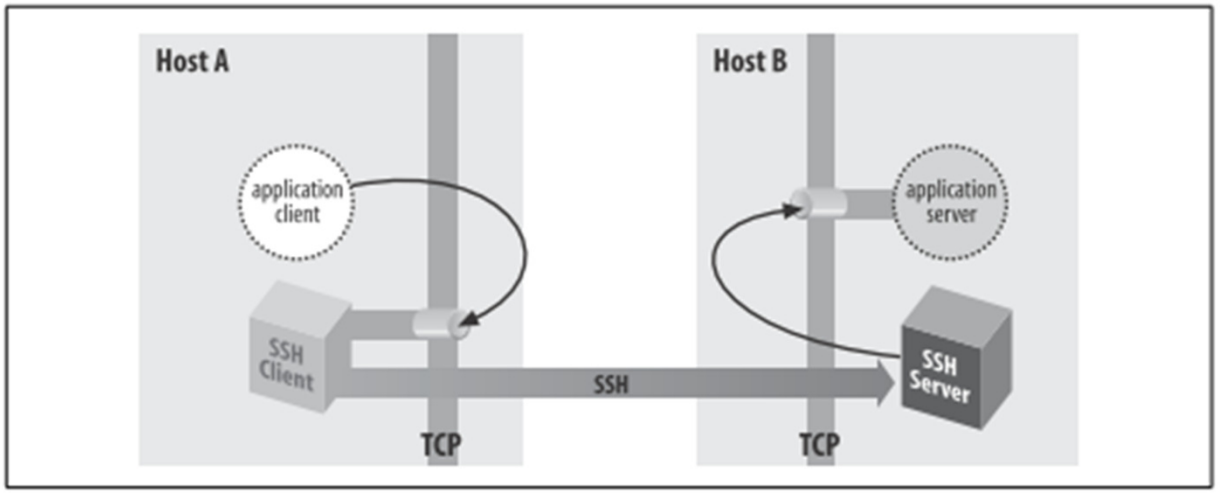
Direct client/server connection (no forwarding)



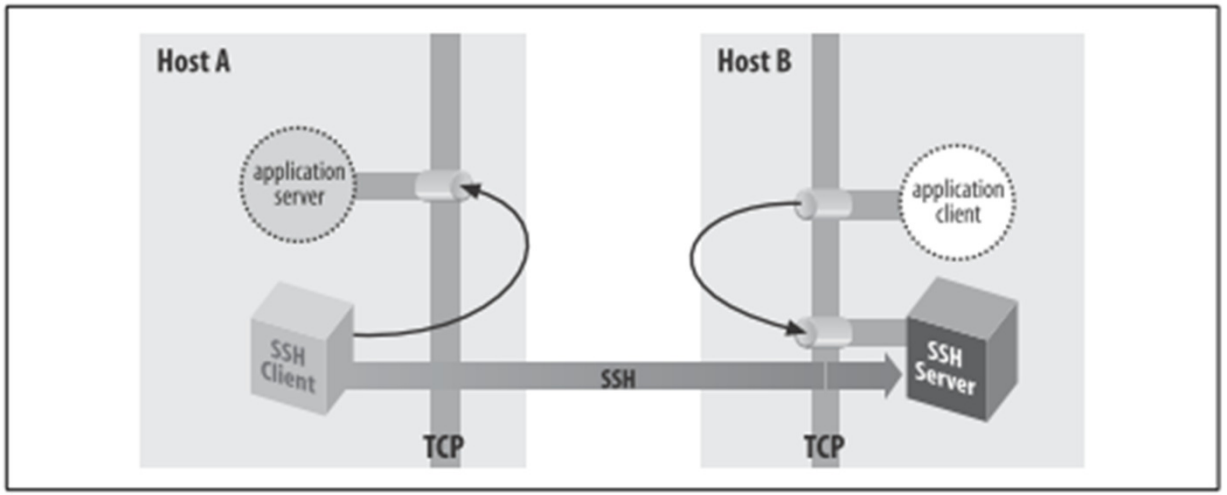
A forwarded port



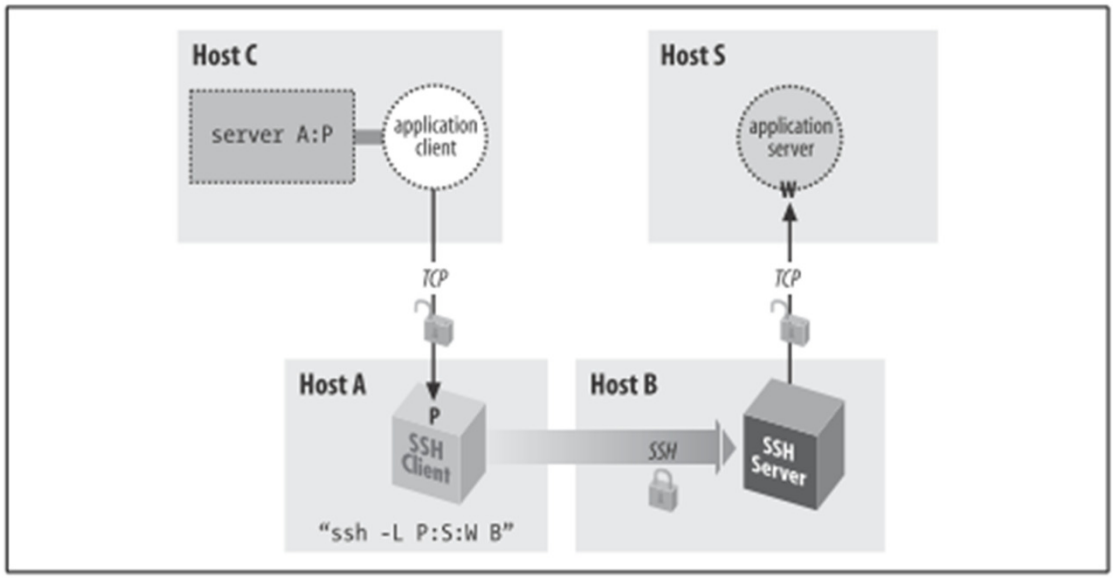
A forwarded connection



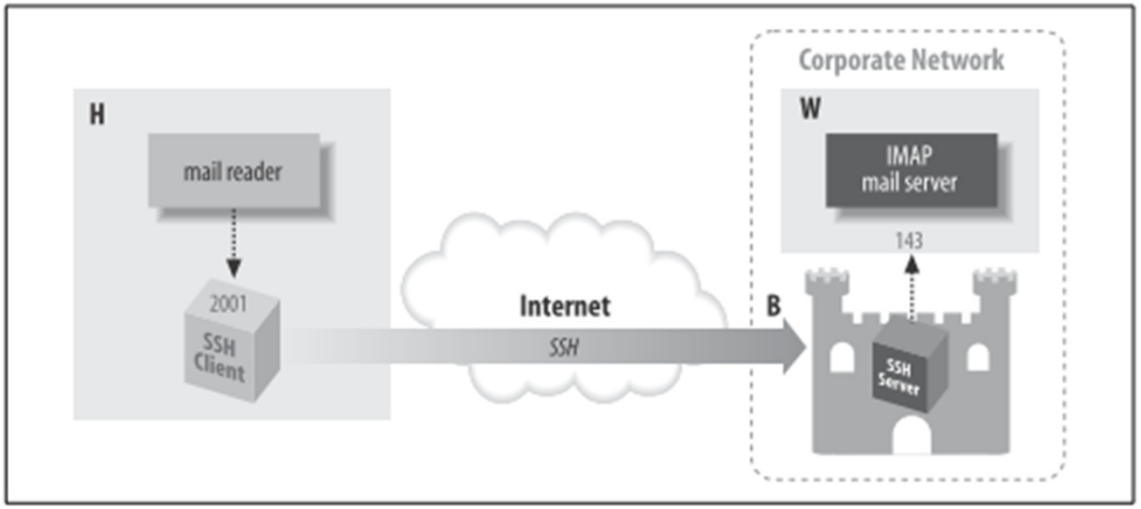
Local forwarding



Remote forwarding



Off-host port forwarding



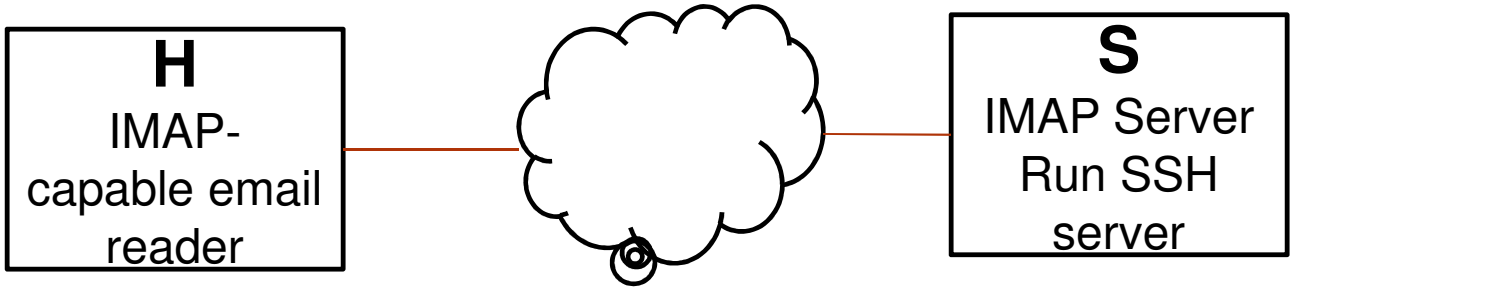
Bypassing a Firewall

Forward and Encrypt Traffic with SSH

- Keep network traffic to arbitrary ports secure with SSH port forwarding
- OpenSSH can forward arbitrary TCP ports to the other end of a connection.
- Start an ssh tunnel with the -L (short for “local”) switch:

```
# ssh -f -N -L 110:mailhost:110 user@mailhost
```

Port fowarding: Example

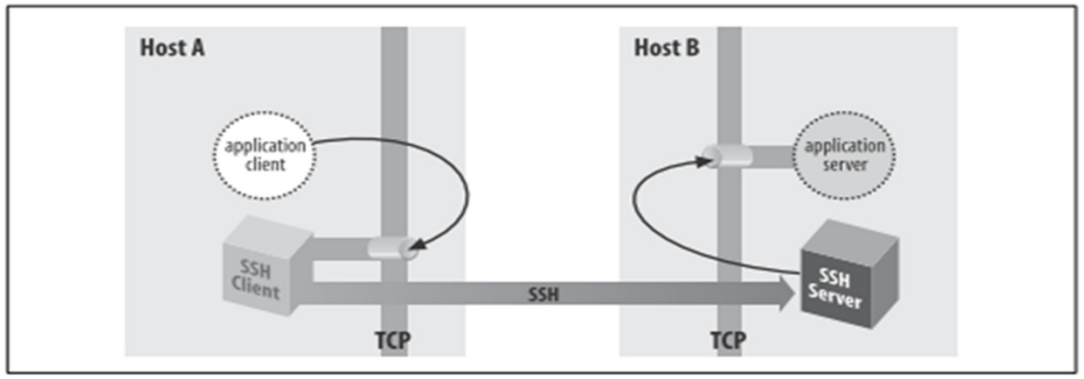


IMAP uses TCP port 143

- **Local port forwarding**
  - To tunnel the IMAP connection through SSH, we need to pick a local port on home machine H (between 1024 and 65535) and forward it to the remote socket (S,143). Creating the tunnel:

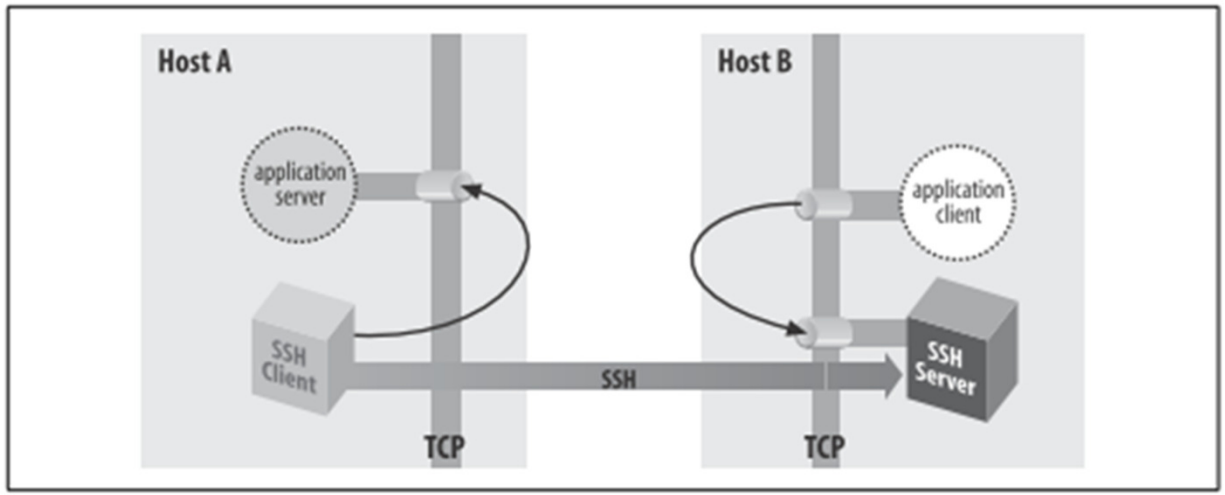
    $ ssh -L2001:localhost:143 S

  - –L option specifies local forwarding, in which the TCP client is on the local machine with the SSH client
  - local port to listen on (2001), the remote machine name or IP address (S), and the remote, target port number (143)
  - To make use of the tunnel, configuring email program to connect to port 2001 on home machine H, socket (localhost,2001)



- **Remote forwarding** : A remotely forwarded port is just like a local one, but the directions are reversed. Client is remote, its server is local, and a forwarded connection is from the remote machine.
  - Create a secure tunnel for remote clients (on machine H) to reach the IMAP server on port 143:

```
$ ssh -R2001:localhost:143 H
```



Use SSH As a SOCKS Proxy

- Local “dynamic” application-level port forwarding:

  -Allocating a socket to listen to port on the local side

  -Whenever a connection is made to this port, the connection is forwarded over the secure channel, and the application protocol is then used to determine where to connect to from the remote machine.

  -Currently the SOCKS 4 protocol is supported, and SSH will act as a SOCKS 4 server.

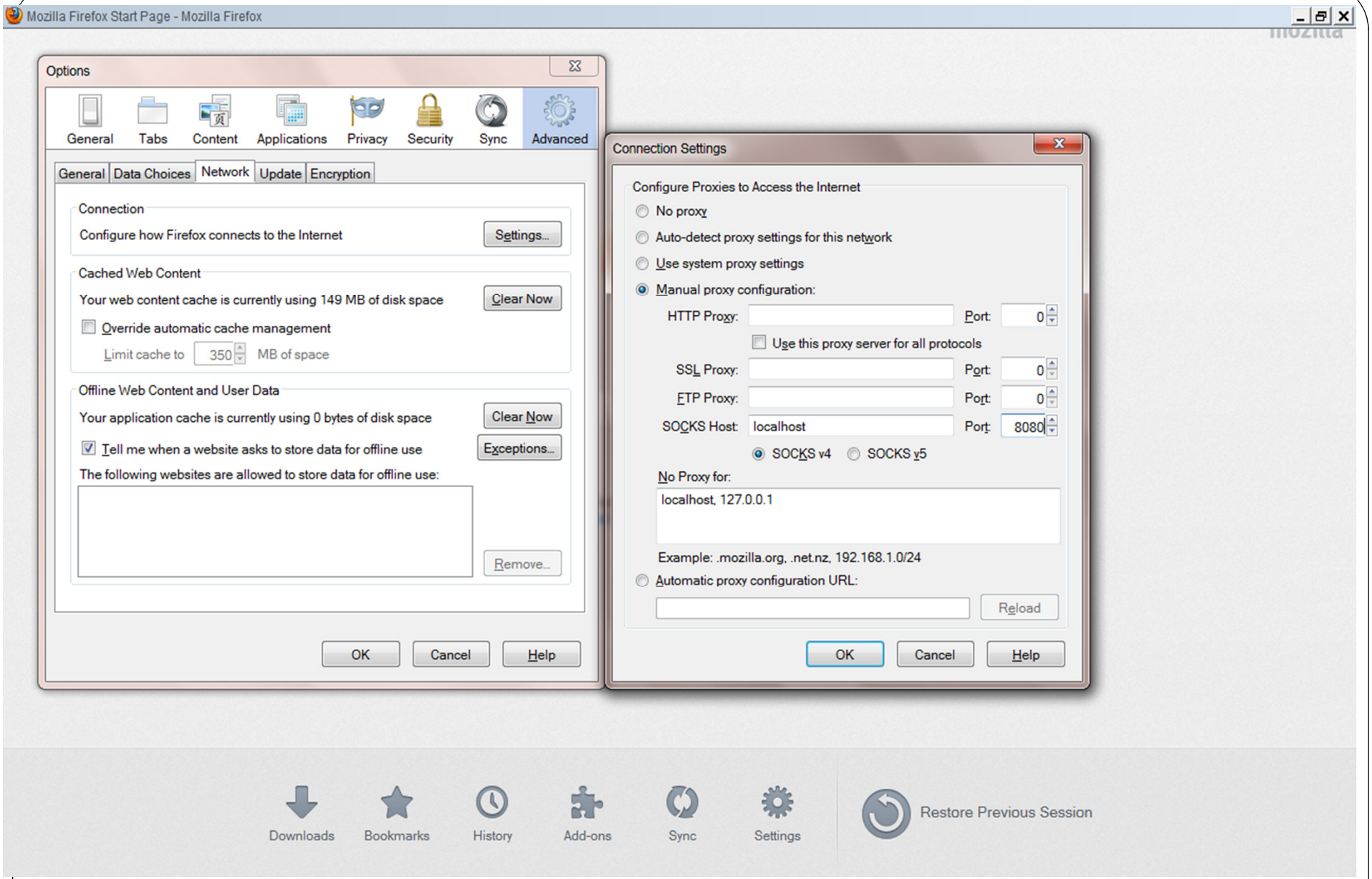
  (Only root can forward privileged ports)

Use SSH As a SOCKS Proxy: Example

- To set up a SOCKS 4 proxy from local port 8080 to remote, type the following:

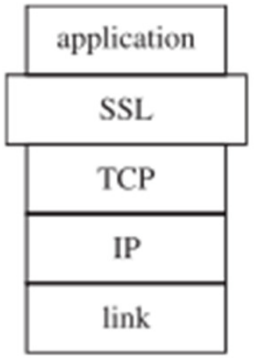
```
$ ssh -D 8080 remote
```

- To specify localhost:8080 as the SOCKS 4 proxy in application (example Firefox), and all connections made by that application will be sent down the encrypted tunnel.


Mozilla Firefox Start Page - Mozilla Firefox
Options
General
Tabs
Content
Applications
Privacy
Security
Sync
Advanced
General
Data Choices
Network
Update
Encryption
Connection
Configure how Firefox connects to the Internet
Settings...
Cached Web Content
Your web content cache is currently using 149 MB of disk space
Clear Now
Override automatic cache management
Limit cache to
350
MB of space
Offline Web Content and User Data
Your application cache is currently using 0 bytes of disk space
Clear Now
Tell me when a website asks to store data for offline use
Exceptions...
The following websites are allowed to store data for offline use:
Remove...
OK
Cancel
Help
Connection Settings
Configure Proxies to Access the Internet
No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Manual proxy configuration:
HTTP Proxy:
Port:
0
Use this proxy server for all protocols
SSL Proxy:
Port:
0
FTP Proxy:
Port:
0
SOCKS Host:
localhost
Port:
8080
SOCKS v4
SOCKS v5
No Proxy for:
localhost, 127.0.0.1
Example: .mozilla.org, .net.nz, 192.168.1.0/24
Automatic proxy configuration URL:
Reload
OK
Cancel
Help
Downloads
Bookmarks
History
Add-ons
Sync
Settings
Restore Previous Session

SSL and TLS

- SSL and TLS are protocols that provide session encryption and integrity for packets sent from one computer to another.
- They can be used to secure client-to-server or server-to-server network traffic.
- They also provide authentication of the server to the client and (optionally) of the the client to the server through X.509 certificates (digital certificates)
- TLS is an enhancement of SSL



The SSL Security layer.

SSL and TLS (2/2)

- The most common use of SSL is between a web client and a web server because it is supported by web browsers and web servers on all platforms and has become the standard for encrypting HTTP traffic
- HTTP over SSL uses port 443 by default, a firewall between the Internet and a web server that uses SSL on its default port would need to allow incoming and outgoing traffic on port 443
- SSL has two components, the SSL Handshake Protocol and the SSL Record Layer.

Encrypt and Tunnel Traffic with SSL

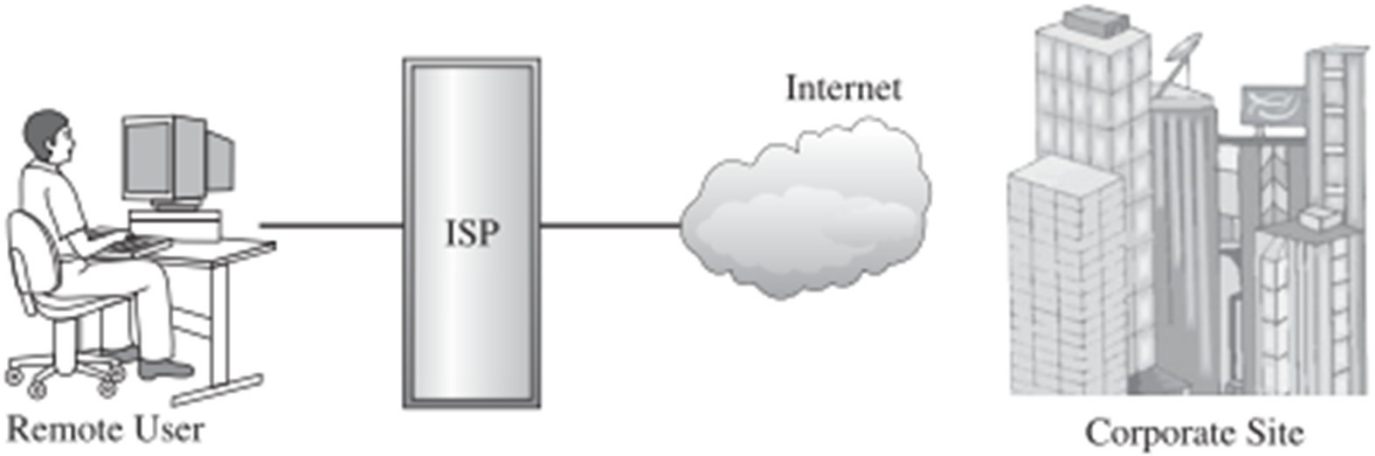
- Use stunnel to add SSL encryption to any network service
- Stunnel (http://www.stunnel.org) is a proxy designed to add TLS encryption functionality to existing clients and servers without any changes in the programs' code
- Building Stunnel: Install OpenSSL first, To install stunnel, simply run ./configure from the directory that was created when unpacked the archive file that was downloaded

- Configuring stunnel: the basic form of a configuration file used to forward a local port to a remote port with stunnel.

The client side:
pid =
client = yes
[<server port>]
accept = <forwarded port>
connect = <remote address>:<server port>
The server side:
cert = /etc/stunnel/stunnel.pem
pid =
client = no
[<forwarded port>]
accept = <server port>
connect = <forwarded port>

VPN

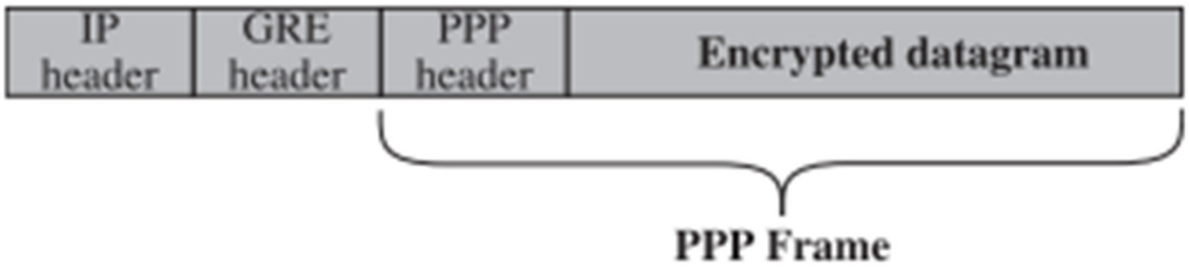
- Virtual Private Networks is a secure tunnel through a non-secure network, such as the Internet



Remote user connecting to a VPN.

VPN: PPTP

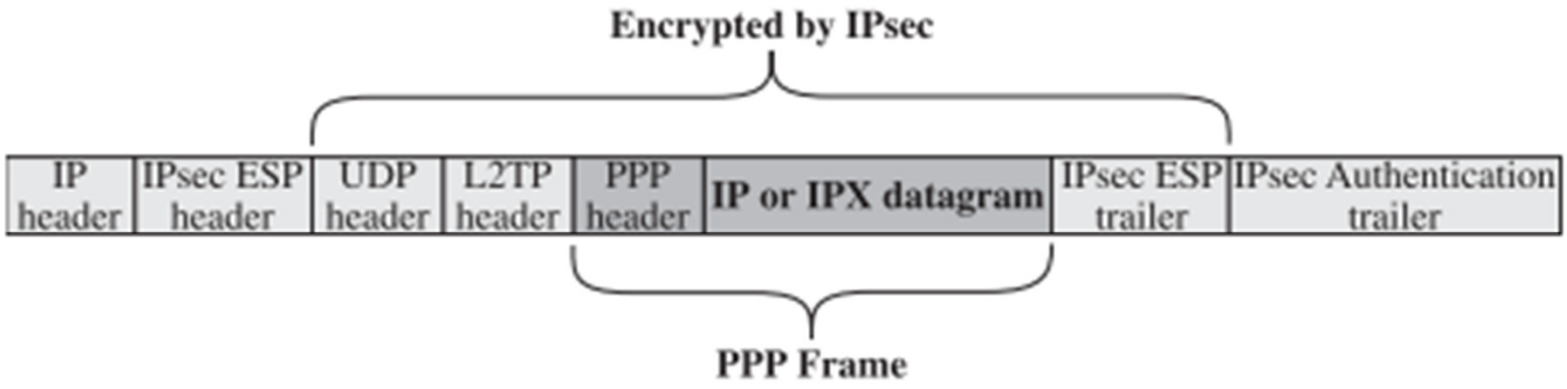
- Point-to-Point Tunneling Protocol : PPTP is a Layer 2 tunneling protocol that encapsulates PPP packets into IP datagrams by adding a Generic Routing Encapsulation (GRE) header and an IP header



PPTP encapsulation.

VPN: L2TP

- Layer 2 Tunneling Protocol (L2TP) is an industry-standard tunneling protocol.
- L2TP provides tunneling and authentication, and utilizes IPsec to provide encryption



L2TP encapsulation.

VPN: Hardware VPN Solutions

- Hardware VPN solutions that provide both IPsec and Secure Sockets Layer (SSL) encryption
- Cisco System, Juniper, Nortel….

Create a Cross-Platform VPN

- OpenVPN (http://openvpn.sourceforge.net)
- OpenVPN uses SSL and relies on the OpenSSL library
- To accomplish the tunneling, OpenVPN makes use of the host operating system's virtual TUN or TAP device (virtual network interfaces)
- Openvpn program

Installing OpenVPN

- Windows: download, install and configure
- Linux: make sure that have OpenSSL installed, download, install and configure

```
$ tar xfz openvpn-2.0.7.tar.gz
$ cd openvpn-2.0.7
$ ./configure && make
```

- Installing the LZO compression library (http://www.oberhumer.com/opensource/lzo/) make much more efficient use of bandwidth

```
$ ./configure --with-lzo-headers=/usr/local/include \
--with-lzo-lib=/usr/local/lib
```

Use PPP and SSH to create a secure VPN tunnel

- Create the actual PPP connection in one quick command

```
# /usr/sbin/pppd updetach noauth silent nodeflate \
pty "/usr/bin/ssh root@colossus /usr/sbin/pppd nodetach notty noauth" \
ipparam 10.1.1.20:10.1.1.1
root@colossus's password:
local  IP address 10.1.1.20
remote IP address 10.1.1.1
```

The End