zSeries Security Update 2005

OpenSSH Updates

Redbooks

International Technical Support Organization

Overview

- **Problem:**

  Unencrypted network data (including passwords)

  - Numerous customer requirements

- **Solution:**

  OpenSSH – suite of network connectivity tools that provide secure encrypted communications between two untrusted hosts over an insecure network.

- Program product: IBM Ported Tools for z/OS
  - unpriced, runs on z/OS 1.4 or higher.
  - GA Version info: OpenSSH 3.5p1, OpenSSL 0.9.7b, zlib 1.1.4
  - Next version will be: OpenSSH 3.8.1p1, OpenSSL 0.9.7d, zlib 1.1.4

What is OpenSSH – Openssh provides:

- Authentication (both client and server) through:
  - Public key cryptography
  - Existing login passwords
  - Trusted hosts authentication
- Data Privacy - through encryption
- Data Integrity - guarantees data traveling over the network is unaltered
- Authorization – regulates access control to accounts
- Forwarding (a.k.a. tunneling) – encryption of other TCP/IP-based sessions

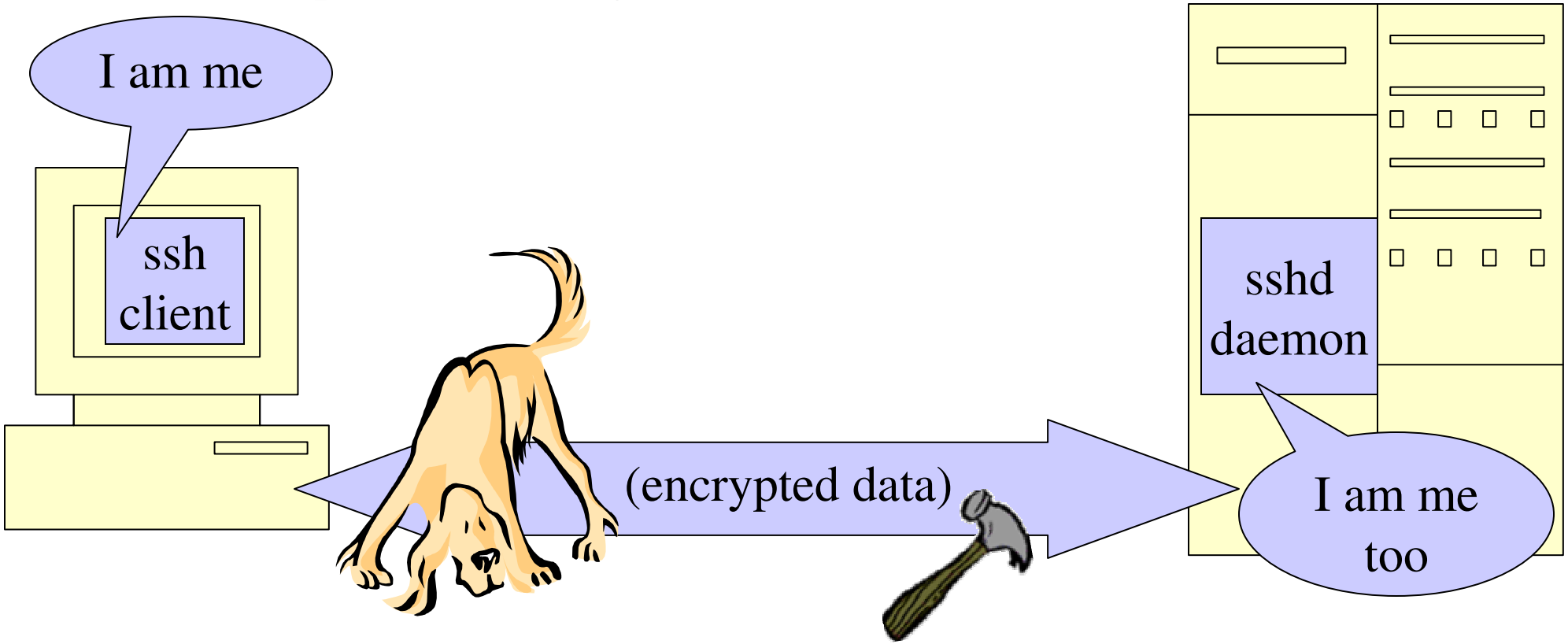
What is OpenSSH – Openssh provides:

Function	OpenSSH Utility	An alternative to…
Secure remote login	ssh, sshd	rlogin, rsh
Secure file transfer	sftp, sftp-server, scp	rcp

OpenSSH additionally provides these utilities:	
Key management	ssh-keygen, ssh-agent, ssh-add, ssh-keyscan

What is OpenSSH – an example

- Provides *secure*:
  - remote login
  - remote command execution
  - TCP/IP port forwarding
- ssh is not a shell in the "UNIX" sense.
- SSH protocols 1 and 2 supported

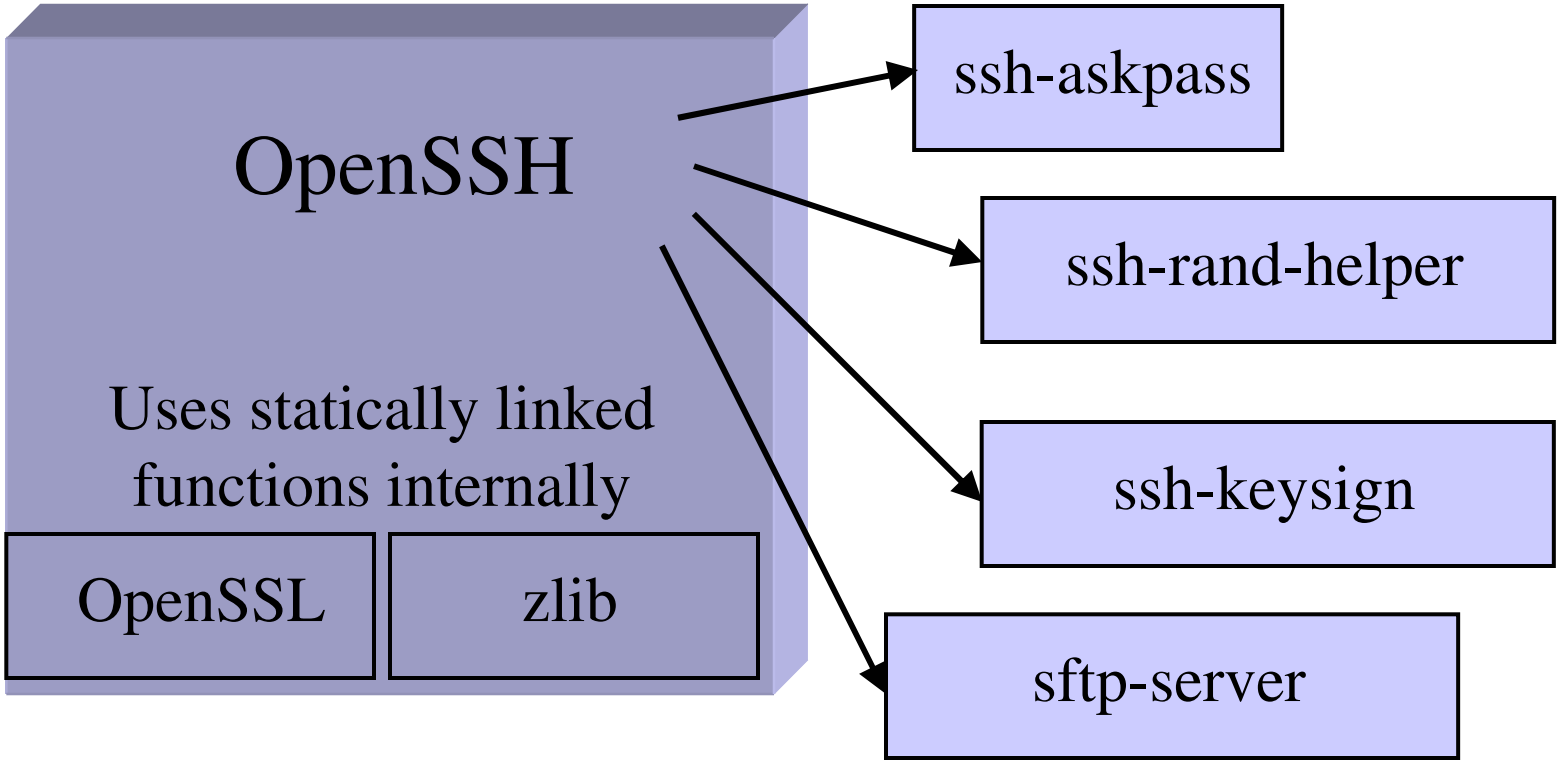


Threats OpenSSH can counter

- Eavesdropping
- Name service and IP spoofing
- Connection Hijacking
- Man-in-the-Middle Attacks
- Insertion Attacks

OpenSSH – functional content

OpenSSH provides the command interface for :
ssh, sshd, scp, sftp, ssh-agent, ssh-add, ssh-keygen, ssh-keyscan



OpenSSH – functional content – ssh and sshd

- ssh – secure remote login program
  - a secure alternative to rlogin, rsh, rexec

- sshd – secure remote login daemon
  - daemon that listens for connections from ssh clients
  - handles key exchange, encryption, authentication, command execution, and data exchange

- Once an SSH session is established, other connections can be forwarded over this secure channel:
  - TCP/IP, Authentication agents

- Together, ssh and sshd provide secure encrypted communications between two untrusted hosts over an insecure network.

OpenSSH – secure file transfer

- sftp (secure file transfer program)
  - An interactive file transfer program, similar to the ftp user interface
  - Performs all operations over an encrypted ssh transport
  - May also use many features of ssh
- sftp-server (SFTP server subsystem)
  - Server-side of the SFTP protocol
  - Invoked from sshd

NOTE: SFTP does not talk FTP protocols

- scp – secure copy (remote file copy program)
  - Also uses ssh for data transfer
  - Similar to rcp (command syntax)
  - Unlike rcp, scp asks for passwords/passphrases if necessary

OpenSSH – Key management

•OpenSSH's key generation and management is separate from other key management provided by IBM.

- ssh-keygen - creates public/private key pairs
- ssh-agent – holds private keys in memory, saving you from retyping your passphrase repeatedly
- ssh-add – loads private keys into the agent
- ssh-keyscan – gathers SSH public host keys

OpenSSH – port forwarding (tunneling)

- Insecure TCP/IP protocols can be made secure by forwarding the connections through SSH

- This also means that firewalls can be bypassed.
  - The administrator configuring OpenSSH may want to consult with network and/or security administrators or policies.

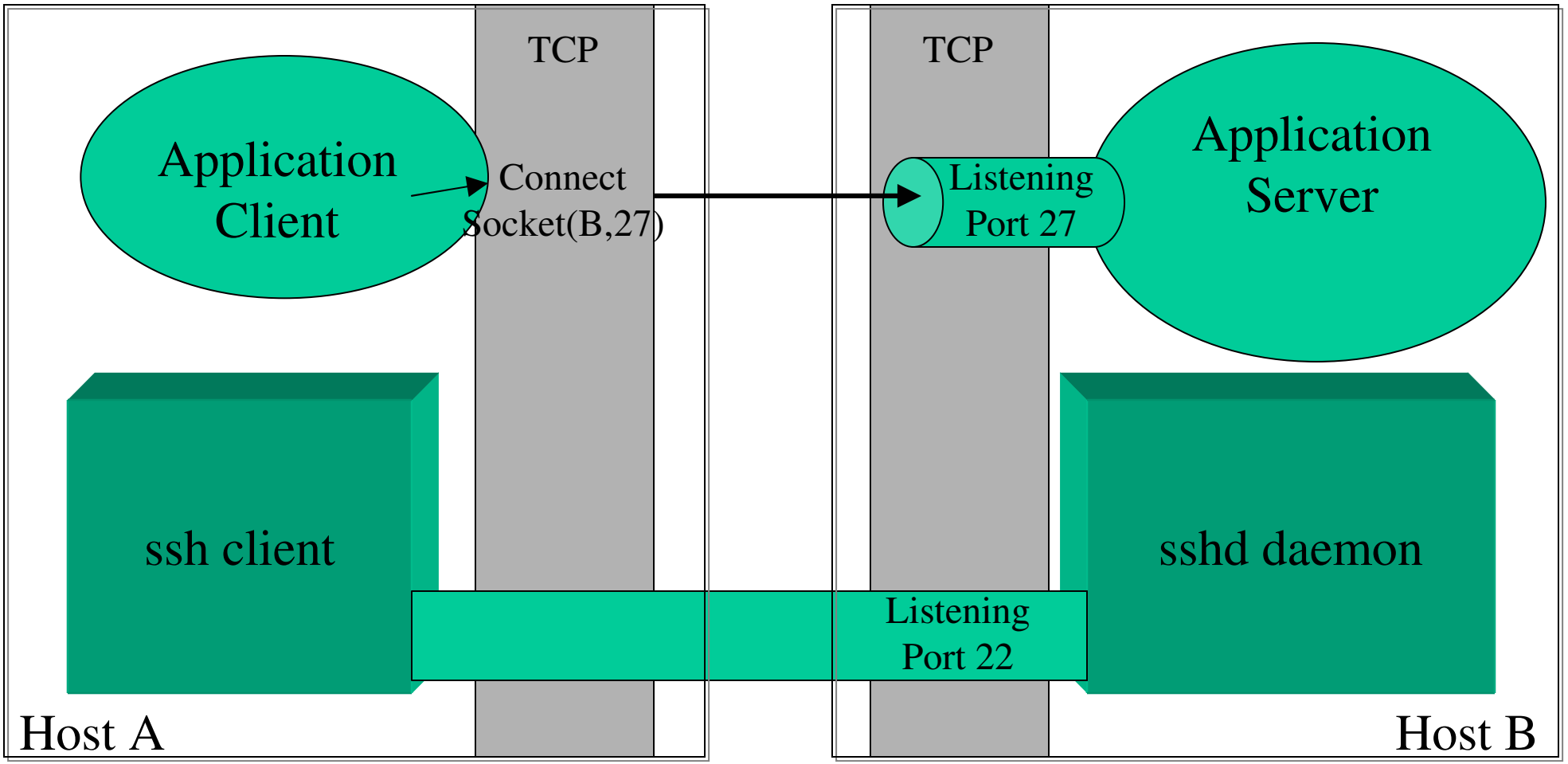
- This feature is disabled by default.

OpenSSH – Algorithms used

- Data privacy- Symmetric algorithms for session data
  - Advanced Encryption Standard (AES) with 128, 192, or 256 bit keys
  - arcfour, blowfish, DES, triple-DES (3DES), and CAST-128
- Data integrity – hash algorithms
  - SSH Protocol version 2 - uses MD5, SHA-1, RIPEMD-160
  - SSH Protocol version 1 – uses weaker CRC-32
- User and Host authentication – public key algorithms
  - RSA, DSA
- Session key exchange – Diffie-Hellman

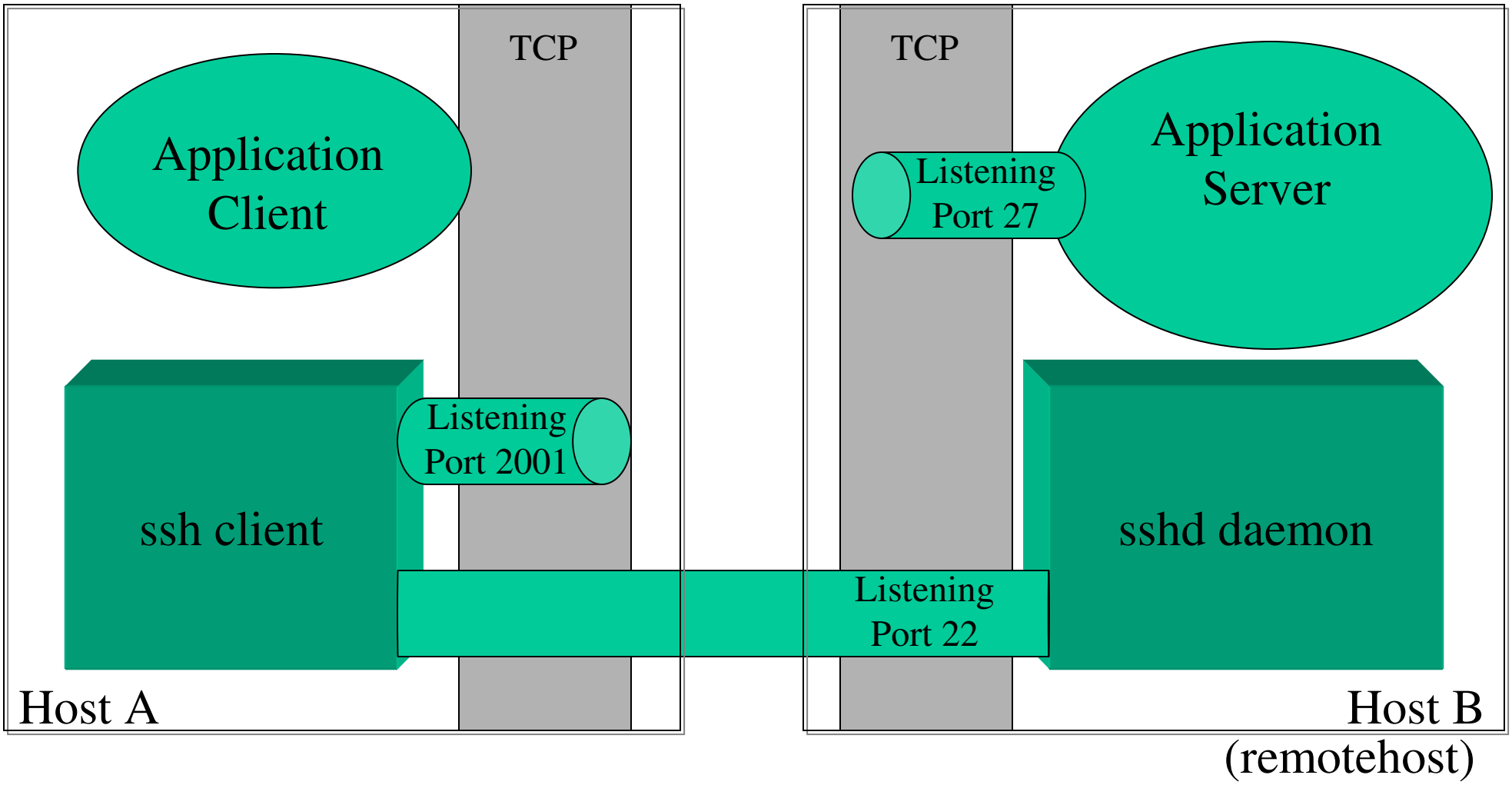
OpenSSH – Without TCP/IP Port Forwarding

Direct client/server connection (no forwarding)



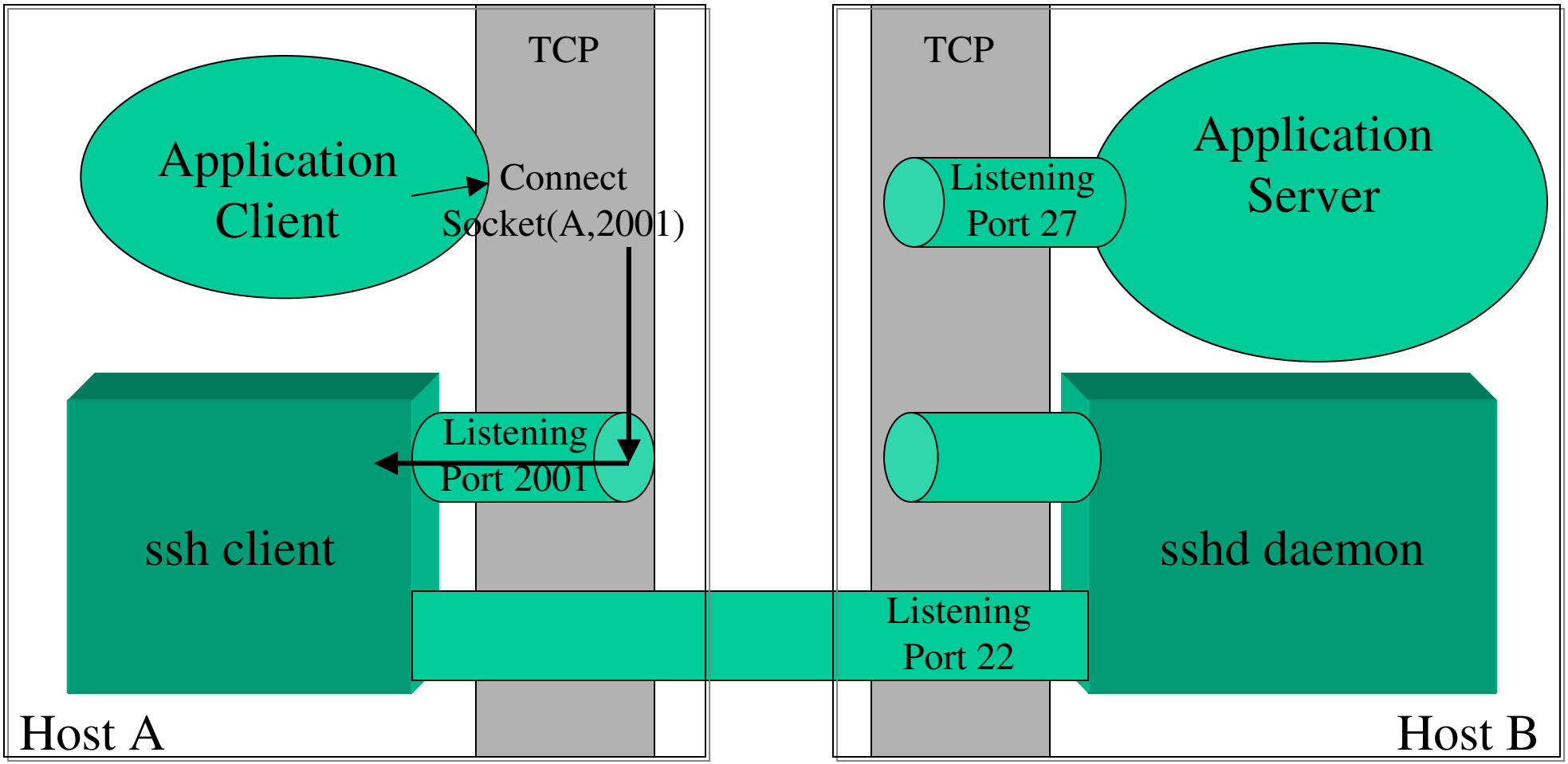
OpenSSH – TCP/IP Port Forwarding

ssh –L 2001:remotehost:27 billy@remotehost



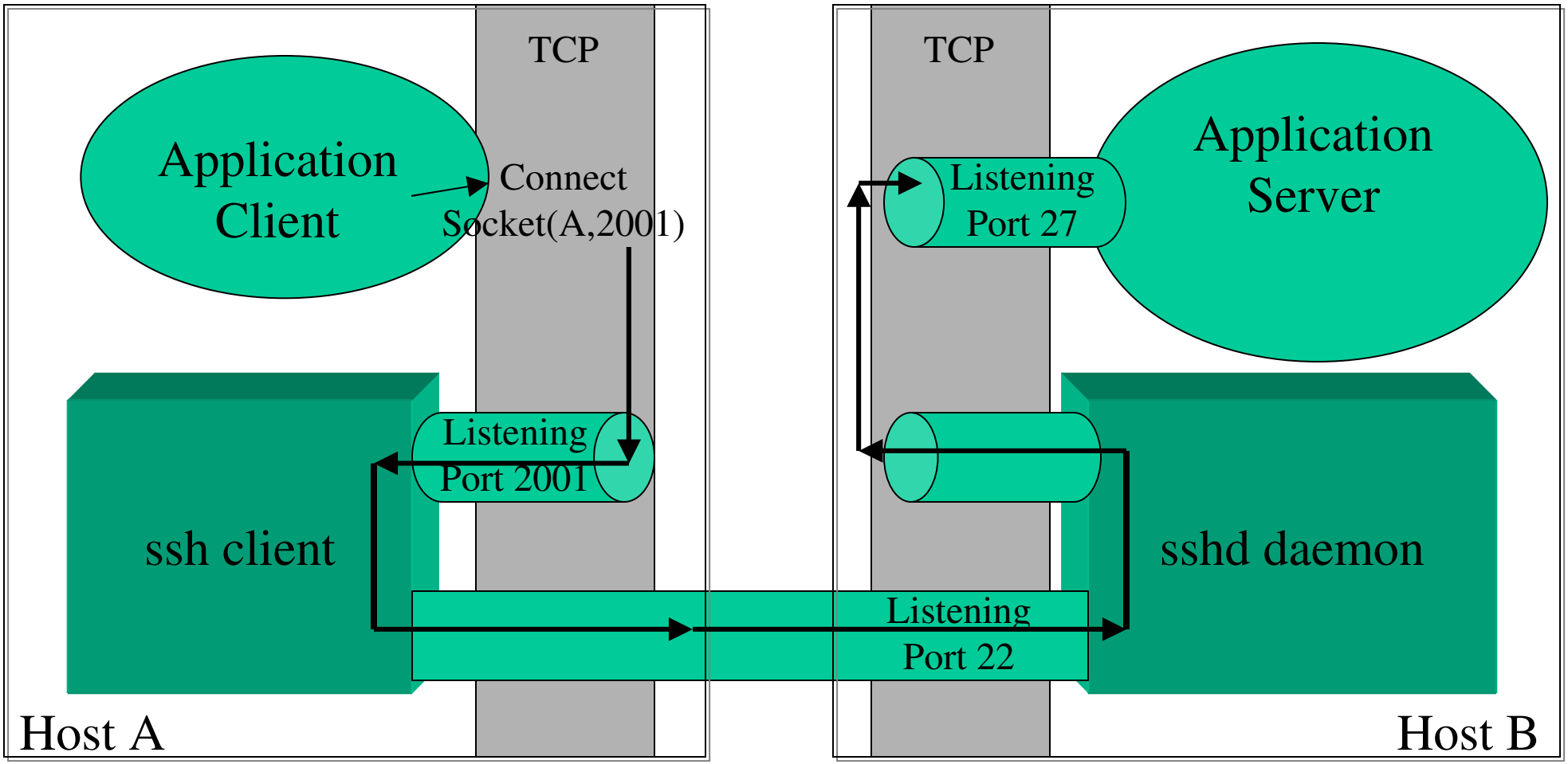
OpenSSH – TCP/IP Port Forwarding

TCP/IP application wants to contact server, through SSH connection



OpenSSH – TCP/IP Port Forwarding

ssh forwards the data through an SSH tunnel, sshd delivers to server



OpenSSH 3.8.1p1 will be next version on z/OS

- Provided via the **service stream – APAR OA10315**
- OpenSSL 0.9.7d
- What's new:
  - Multilevel Security support
  - Password reset capability
  - Daemon restart capability
    - if TCP/IP is recycled, sshd will not go down, but will wait until TCP/IP returns and re-initialize itself.
    - if sshd started from /etc/rc and TCP/IP hasn't been started yet, sshd will wait for TCP/IP to come up.
    - in CINET environment, a new stack will automatically be recognized by the daemon (no SIGHUP required.)

(continued…)

OpenSSH 3.8.1p1 – What's new (continued)

- Open Source function added in OpenSSH 3.8.1p1
  - New encryption algorithms, command-line options
  - Contains fixes, reduces vulnerabilities
  - New Configuration keywords:
    - ssh_config
      - AddressFamily
      - ConnectTimeout
      - EnableSSHKeysign
      - ForwardX11Trusted
      - IdentitiesOnly
      - ServerAliveInterval
      - ServerAliveCountMax
      - TCPKeepAlive
      - VerifyHostKeyDNS
    - sshd_config
      - TCPKeepAlive
      - UseDNS

OpenSSH 3.8.1p1 - Changes to Consider

- The following configuration keywords were changed
  - the previous names are still supported on z/OS, but not by the OpenSSH distribution.
  - we recommend you move to new settings, once all systems sharing a config file have been upgraded.

daemon (sshd_config)

OpenSSH 3.5p1	**OpenSSH 3.8.1p1**
KeepAlive	TCPKeepAlive
VerifyReverseMapping	UseDNS

client (ssh_config)

KeepAlive	TCPKeepAlive

NOTE: The default settings for these configuration keywords have not changed.

Migration/Coexistence Considerations

- The RhostsAuthentication keyword was removed from the OpenSSH base distribution.
  - This method of authentication is not secure.
  - IBM z/OS **still maintains support** but recommends against using this setting.
  - It is both a client and daemon configuration option
  - Protocol Version 1 only, and defaults are “no” (disabled)

- In OpenSSH 3.5p1, HostbasedAuthentication automatically enabled use of ssh-keysign.
  - For OpenSSH 3.8.1p1, ssh-keysign is controlled by a separate (new) configuration keyword: EnableSSHKeysign
  - Must be set in the global ssh_config file.
  - default settings for both HostbasedAuthentication and EnableSSHKeysign are “no” (disabled.)

Migration/Coexistence Considerations

- OpenSSH is a program product for z/OS 1.4 and higher. However….

- A version of OpenSSH has been on our z/OS Unix Tools & Toys page:
  http://www-1.ibm.com/servers/eserver/zseries/zos/unix/bpxa1toy.html

  The customer should remove all executables from that version before installing the OpenSSH officially provided by IBM.

  The customer will want to keep their key files (host keys, user keys), and merge their existing configuration files with our samples.

Installation

- Prerequisites for installation
  - No changes (all would have been done when installing original Program Product.)

- Publications References:
  - IBM Ported Tools for z/OS User's Guide (SA22-7985-00)
  - IBM Ported Tools for z/OS Program Directory (GI11-2847-00)

Session Summary

- OpenSSH upgraded from 3.5p1 to 3.8.1p1
- IBM also added new function

Publications:

- IBM Publications:
  - IBM Ported Tools for z/OS User's Guide (SA22-7985-00)
  - IBM Ported Tools for z/OS Program Directory (GI11-2847-00)
  - IBM Ported Tools for z/OS License Information (GA22-7986-00)

- Other References:
  - OpenSSH home page: www.openssh.org
  - IETF **Secure Shell (secsh) Working Group**
    - **http://www.ietf.org/html.charters/secsh-charter.html**
  - SSH The Secure Shell, The Definitive Guide. Barret & Silverman. 2001 O'Reilly & Associates, Inc.