Hands-on Keyboard: Cyber Experiments for Strategists and Policy Makers

The Client-Server Model

1. Introduction

1.1. Exercise Description

Q: What is a secure computer?

A1: An abacus

A2: Unplugged

Functionality and complexity are enemies of security. Generally, the more functionality provided by a system, the harder it is to secure. Servers providing services and opening ports to remote users are particularly prone to vulnerabilities. Attackers often use open ports and exposed programmatic interfaces to subvert a system. This exercise involves reviewing the Client-Server model of computing by installing and utilizing Web, File

The Air Force Cyber College thanks the Advanced Cyber Engineering program at the Air Force Research Laboratory in Rome, NY, for providing the information to assist in educating the general Air Force on the technical aspects of cyberspace.

Transfer Protocol (FTP), and Secure Shell (SSH) clients and servers on Windows and Linux systems.

1.2. Objectives

- Compare and contrast the World Wide Web and the Internet.
- Describe the Client-Server model of computing.
- Memorize select server port assignment.
- Install and utilize a Web (HTTP) client and Web server.
    - Create and serve a simple web page.
    - Analyze HTTP Request and Response Headers with Wireshark.
- Install and utilize a SSH client and a SSH server.
- Command and control a remote system with SSH.
- Install and use an FTP client an a FTP server.
- Transfer files with FTP.

1.3. Materials

- Computer running Windows with access to an account with administrative rights
- VirtualBox
- Ubuntu OS .iso file

1.4. Assumptions

- The provided instructions were tested on an Ubuntu 15.10 image running on a Windows 8 physical machine. Instructions may vary for other OS.
- The student has administrative access to their system and possesses the right to install programs.
- The student's computer has Internet access.

1.5. Random Notes

- SSH uses public key encryption. Telnet is its unencrypted predecessor.
- Keep in mind there are still many attacks that compromise encrypted services. With today's technology, no computer or service is 100 percent secure if it provides any useful functionality!

2. Client-Server Model

2.1. What is a server?

A server provides a service and shares its resources (data, computational power) with one or more clients. Examples of web servers include Apache and the Microsoft Internet Information Services (IIS) web server.

2.2. What is a client?

A client requests or receives some services from another computing device. Examples of web clients include Firefox, Chrome, and Internet Explorer (IE).

2.3. The Client-Server Model of Computing

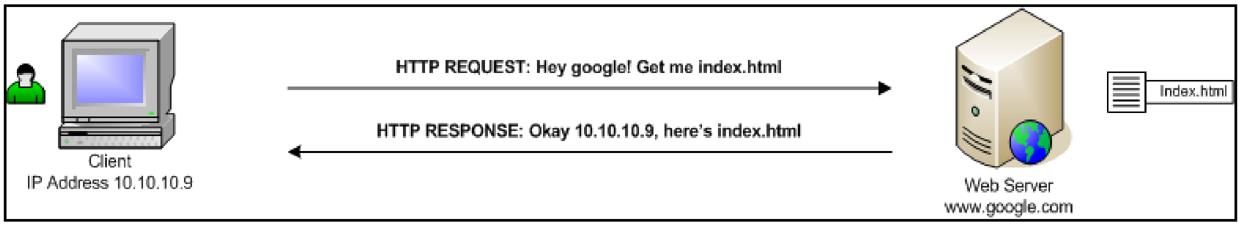
The Client-Server model of computing involves a service provider (server) and service user (client). If the server is multi-threaded, it can service multiple clients at one time. Examples of applications that utilize the Client-Server model include web, database, FTP, and SSH. For web applications, the server (Apache, IIS) typically listens on port 80 for HTTP or port 443 for HTTPS (encrypted). The client (Firefox, Chrome, IE) forms a Transport Control Protocol (TCP) connection to the server's listening port and requests information (GET http://www.page.com/index.html). The server responds by returning the requested information (index.html) back to the client.

2.4. Common Port Numbers

Common services often run on standardized port numbers. See below a list of a few common combinations:

- 20/21 - FTP (data/commands)
- 22 - SSH
- 23 - Telnet (An unencrypted SSH)
- 25 - Simple Mail Transfer Protocol (SMTP - Used by mail servers)
- 80 - HyperText Transfer Protocol (Unencrypted HTTP web traffic)
- 443 - HyperText Transfer Protocol over SSL (Encrypted HTTPS web traffic)

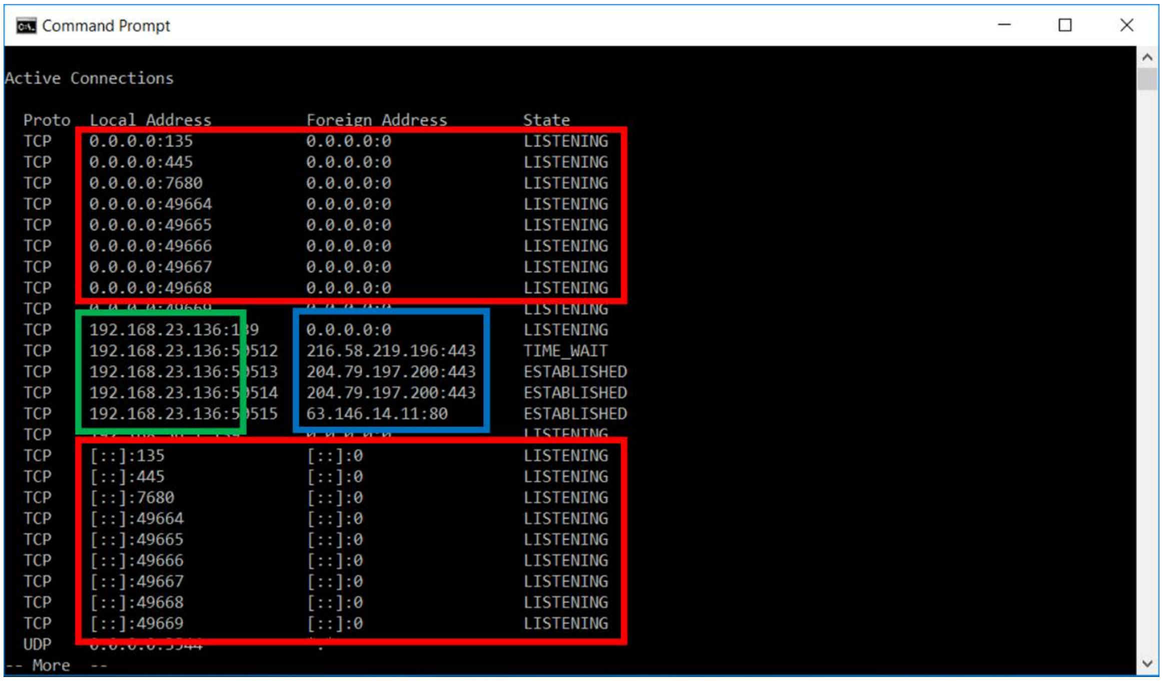
Keep in mind these are conventions. Nothing prevents administrators from configuring SSH servers from listening on port 20 or web servers from listening on port 5555.


HTTP REQUEST: Hey google! Get me index.html
HTTP RESPONSE: Okay 10.10.10.9, here's index.html
Client
IP Address 10.10.10.9
Web Server
www.google.com
Index.html

Web client requests data from a web server listening on a port.

2.5. Netstat

Network statistics or Netstat is a Windows and Linux command used to view open ports on your system. Open a terminal on either windows or Ubuntu and type "***netstat -an | more***" to view an active Internet Client-Server connection.



```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING
  TCP    192.168.23.136:139     0.0.0.0:0              LISTENING
  TCP    192.168.23.136:5 512   216.58.219.196:443     TIME_WAIT
  TCP    192.168.23.136:5 513   204.79.197.200:443     ESTABLISHED
  TCP    192.168.23.136:5 514   204.79.197.200:443     ESTABLISHED
  TCP    192.168.23.136:5 515   63.146.14.11:80        ESTABLISHED
  TCP                           0.0.0.0:0              LISTENING
  TCP    [::]:135               [::]:0                 LISTENING
  TCP    [::]:445               [::]:0                 LISTENING
  TCP    [::]:7680              [::]:0                 LISTENING
  TCP    [::]:49664             [::]:0                 LISTENING
  TCP    [::]:49665             [::]:0                 LISTENING
  TCP    [::]:49666             [::]:0                 LISTENING
  TCP    [::]:49667             [::]:0                 LISTENING
  TCP    [::]:49668             [::]:0                 LISTENING
  TCP    [::]:49669             [::]:0                 LISTENING
  UDP    0.0.0.0:3544           *:*
-- More  --
```

Some information gleaned from the screenshot above:

- The system's IP address is 172.16.1.244.
- The system is listening on ports 135, 445, 623 and so on.
- The system is connected to several IP addresses on port 443.

2.6. The Internet and the World Wide Web

The Internet and the World Wide Web are different entities, but people often use the terms interchangeably. The Internet existed before the World Wide Web. Internet-connected computers exchanged e-mail and files long before web browsers, web servers, HTTP, and HTML worked together to make up the World Wide Web. As http://www.netlingo.com put it, “The Web makes the Internet fun to look at and easy to use.”

Internet

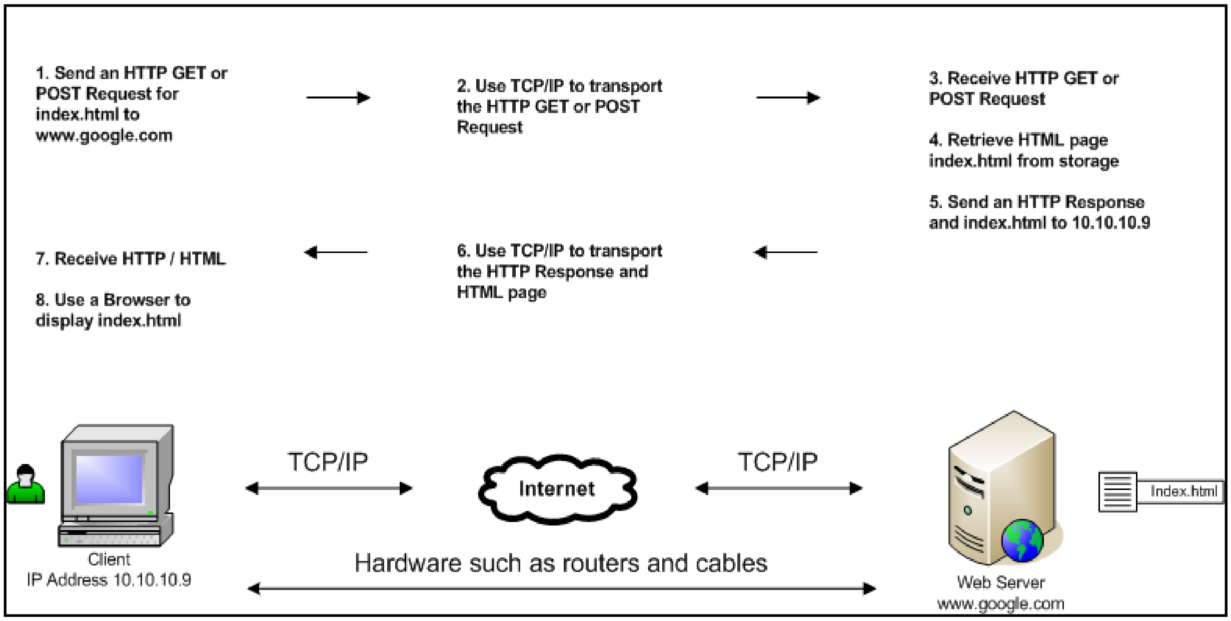
- A network of networks
- Made up of computers and cables
- Existed before the World Wide Web
- Uses TCP/IP

WWW

- A layer on top of the Internet
- Made up of web pages, web servers, and web browsers
- Depends on the Internet
- Uses HTTP

2.7. HTML vs. HTTP

HyperText Markup Language (HTML) and HyperText Transport Protocol (HTTP) are fancy phrases representing a simple process. Pause and process the individual words in each phrase. HTML is a **language** used to create web pages. HTTP is a communication protocol used to **move HTML** pages from one computer to another. Keep this in mind as you complete the lab.


1. Send an HTTP GET or POST Request for index.html to www.google.com
2. Use TCP/IP to transport the HTTP GET or POST Request
3. Receive HTTP GET or POST Request
4. Retrieve HTML page index.html from storage
5. Send an HTTP Response and index.html to 10.10.10.9
6. Use TCP/IP to transport the HTTP Response and HTML page
7. Receive HTTP / HTML
8. Use a Browser to display index.html
TCP/IP
Internet
TCP/IP
Index.html
Client
IP Address 10.10.10.9
Hardware such as routers and cables
Web Server
www.google.com

3. Servers and Clients

3.1. Download and Install ABYSS

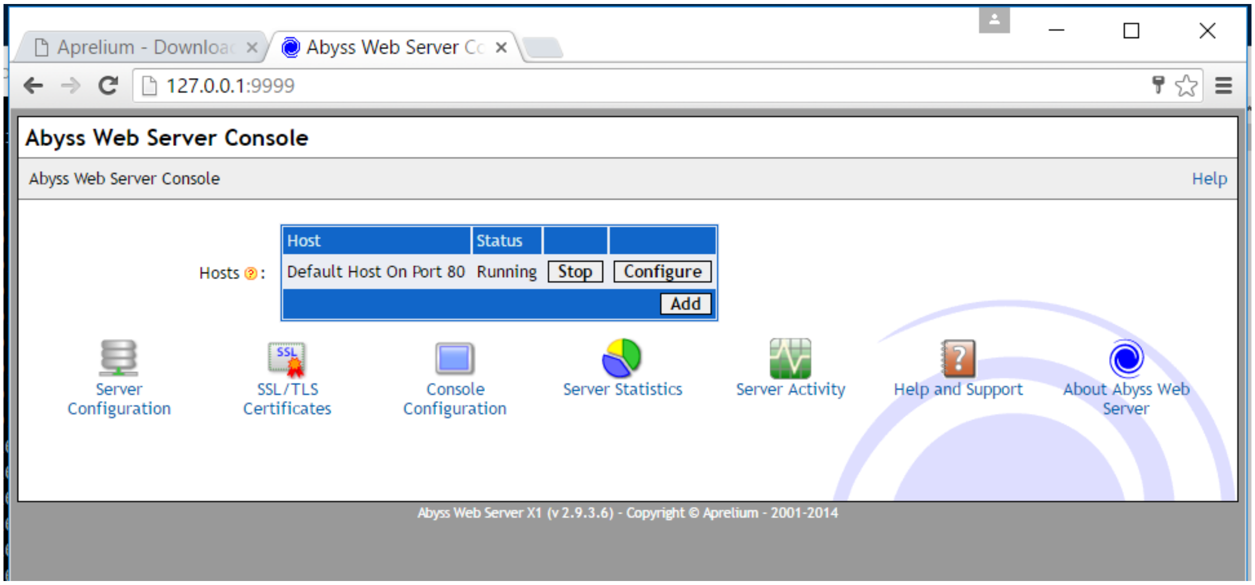
Windows can run the Internet Information Services (IIS) server to host web pages, integrate into Active Directory, serve dynamic content through .NET, and much more. However, for this lab we shall show another alternative, the Abyss Webserver.

Complete these steps on your Windows host machine or a VM running Windows.

- Visit www.aprelium.com/abyssws/download.php and download Abyss.
- Allow the installer to start Abyss when the installation is complete. Follow the instructions and accept defaults:
  - Click ***I Agree*** for the License Agreement.
  - Keep the default for components to install and select ***Next***.
  - Set the Destination Folder to ***C:\Abyss Web Server\*** and click ***Install***.
  - Select ***Manual Startup*** and click ***OK***.
  - If there are any prompts about firewalls, select "***run on a private network" and click "allow access."***
  - Click ***Yes*** to start Abyss now.
  - Hit ***OK*** to configure the web server via the web interface.

3.2. Configure the ABYSS Web Server

- The default web browser should open a web page at 127.0.0.1:9999/console/language.
- Select ***English***.
- Enter a login and password (remember them!) and select ***OK***.
- Enter the same login and password into the popup box to login to the Server Console.



Abyss Web Server Console

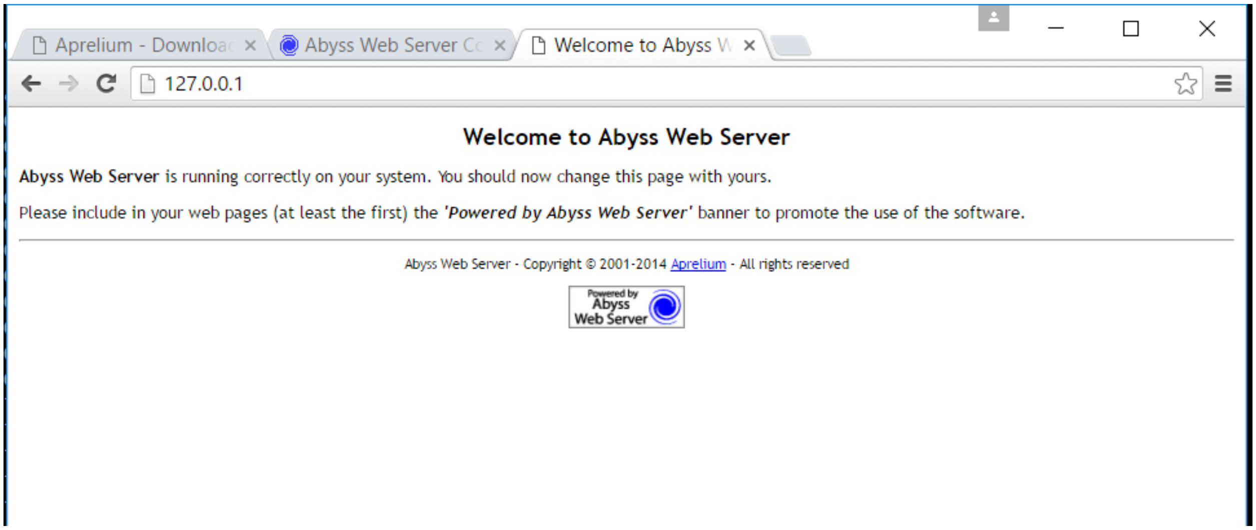
Abyss Web Server Console

Hosts:

Host	Status		
Default Host On Port 80	Running	Stop	Configure
			Add

Abyss Web Server X1 (v 2.9.3.6) - Copyright © Aprelium - 2001-2014

- Note the following:
  - The Abyss web administrative interface runs on port 9999.
  - The Abyss web server runs either on port 80 (default) or 8000 (if something else is already listening on port 80). Remember you can use netstat to view open ports.
- Explore the options provided by the Web Server Console.
- Open the windows terminal. Run ***netstat -an* | *more*** from the terminal to view that port 80 (or port 8000) is open.
- In a web browser (Edge, Internet Explorer, Chrome, Fire/Water-fox, etc.), open a new browser tab and visit ***http://127.0.0.1***. You should see a page similar to the image below if Abyss is up and running.



Welcome to Abyss Web Server

Abyss Web Server is running correctly on your system. You should now change this page with yours.

Please include in your web pages (at least the first) the ***'Powered by Abyss Web Server'*** banner to promote the use of the software.

Abyss Web Server - Copyright © 2001-2014 Aprelium - All rights reserved

3.3. View Web Page

- Visit http://www.securitywizardry.com/radar.htm.
- Right-click on the outer edge of the page (outside any image or box).
- Select "View Page Source" (you can shortcut this and the prior step by typing Ctrl+U).
- Tags are text strings surrounded by <brackets> that denote the appearance and format of a web page. Locate the following tags in the page source:
    - HTML
    - Head
    - Title
    - Body
    - Img (image tag)
    - a href (link)

3.4. Create a Web Page with Hyper Text Markup Language (HTML)

- Start a new Notepad document. (***Start*** → ***All programs*** → ***Accessories*** → ***Notepad***).
- Type the following into the document:

```
<html>
    <head>
            <title> This Might Be My First Webpage </title>
    </head>
    <body>
```

```
                <h1> This is the header of my first (maybe) web page. </h1>
                I created this using HTML.
        </body>
    </html>
```

- Save the webpage as ***index.html*** to the Desktop.
- Right click ***index.html***. Select ***Open with > Chrome***.
- Personalize the webpage:
    - Set the title to <your name>
    - Change the header and body any way you wish
- View the altered web page with Chrome by hitting the refresh button.
- For more information about creating HTML pages visit http://www.w3schools.com.

3.5. Server your Web Page with ABYSS Web Server

- Open Windows Explorer (Win+E) and open ***C:\Abyss Web Server\htdocs\***.
- Replace the Abyss ***index.html*** with your ***index.html***.
- Use your web browser to view ***http://127.0.01*** (or ***http://127.0.0.1:8000***) and record your observations.
    - Note: Any computer may use 127.0.0.1 to refer to itself (can also use ***http://localhost***).
    - You can also view by using the internal or external IP address found with the netstat command. (Example: ***http://192.168.1.22***.)
- Congratulations! You have served your first web page. If you have an externally routable IP address, anyone in the world with an Internet connection and a browser can view it.

3.6. Analyze a HTTP Request Header

- Start Wireshark and begin a live capture.
- Apply a filter to ensure Wireshark only displays HTTP traffic leaving from or arriving to your IP address.
- Open a browser and visit http://sectools.org/.
- Return to Wireshark.
- Select the very first HTTP GET/HTTP/1.1 packet originating from your IP address. Use the ***Packet Details Window*** to answer the following questions:
    - What is the request method?

    - What is the request URI?
    - What is the requests version?
- Why is there more than one HTTP GET request? What are the other requests asking for?
- Examine the ***Packet Details Window*** and find the values of the HTTP Request Headers (some fields may NOT be present, do not worry about it!):
    - Host
    - User-Agent
    - Accept
    - Accept-Encoding
    - Accept-Charset
    - Keep-Alive
    - Connection
    - If-Modified-Since
    - If-None-Match

3.7. Analyze a HTTP Response Header

- Select the first HTTP/1.1 200 OK (text/html) packet sent back to your IP address. Record the response code returned by the server.
- Examine the ***Packet Details Window*** and find the values of the HTTP Response Headers (if present):
    - Date
    - Server
    - Connection
    - Keep-Alive
    - Accept-Ranges
    - Content-Length
    - Connection
    - Content-Type
- Look at the Individual Packet Bytes window. What kind of data does it contain?

3.8. View HTTP Packet Data

- Examine the packet data of several HTTP packets with an Info field of HTTP/1.1 200 OK (text/html).

- Find a packet which has an Uncompressed entity body tag at the bottom of the packet data window.
- Select the Uncompressed entity body tag to view the uncompressed HTML.
- Describe the information available under this tag.

4. Servers and Clients: Secure Shell (SSH) on Linux

4.1. Introduction

SSH enables C3 (Command, Control, and Communication) of a remote system by giving a user terminal for the remote system. It also enables transferring files using the ***scp*** command. In this section you will accomplish the following objectives:

- Windows
  - Download and setup freeSSHd server.
  - Download and utilize the SSH PuTTY client.
- Linux
  - Install the Linux Open-SSH server.
  - Use the Linux Open-SSH client.
  - Practice issuing remote commands.

4.2. Download and Install an SSH Server on Windows

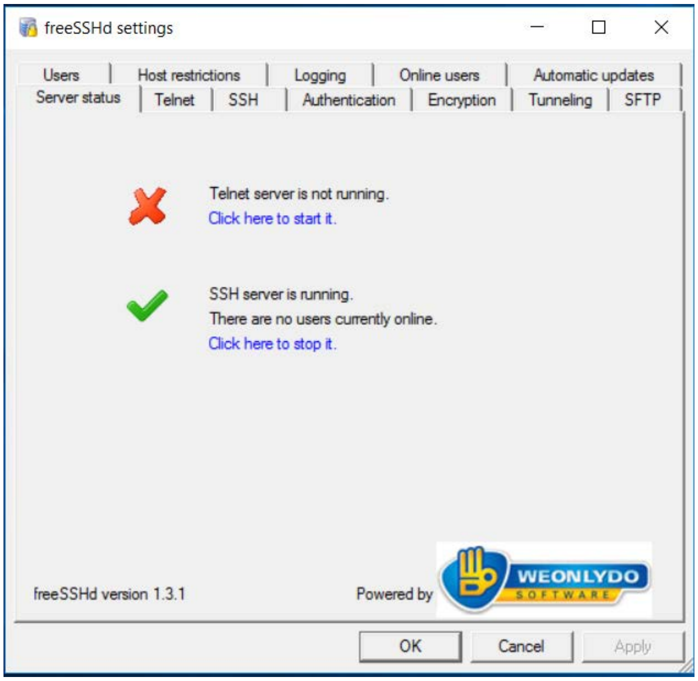
Complete these steps on your Windows VM or host.

4.2.1. Download and install freeSSHd

- Download the latest version of ***freeSSHd*** from www.freesshd.com/?ctt=download.
- Click the freeSSHd.exe to begin the installation and select ***Ok***.
- Accept all of the defaults and click ***Install***.
- Select ***Yes*** to create private keys.
- Select ***No*** to run as a system service.
- Select ***Finish***.

4.2.2. Start freeSSHd

- Double click the freeSSHd shortcut icon on the Desktop.
- If there is a firewall issue with this program, select "***Private networks, such as home or work network***" and "***Allow Access," then click "OK."***
- In the taskbar (bottom right-hand side of screen) right click the ssh icon and select ***Settings*** to open the configuration screen.

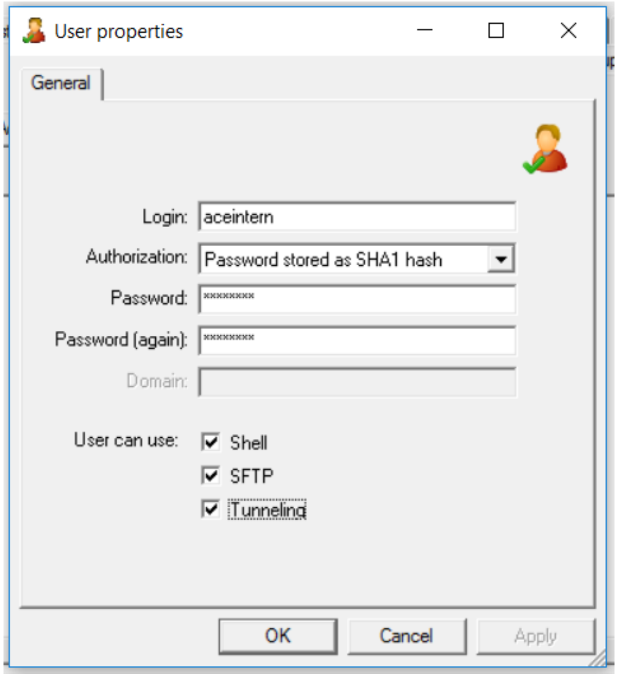


Telnet server is not running.
Click here to start it.

SSH server is running.
There are no users currently online.
Click here to stop it.

freeSSHd version 1.3.1

- Select the ***Users*** tab and click ***Add***.
- Add a login named ***aceintern*** and select "Password stored as SHA1 hash."
- Enter the password ***password*** and select the functions the "user can use" (Shell, SFTRP, and Tunneling) and select ***OK***.



Login: aceintern

Authorization: Password stored as SHA1 hash

Password: ********

Password (again): ********

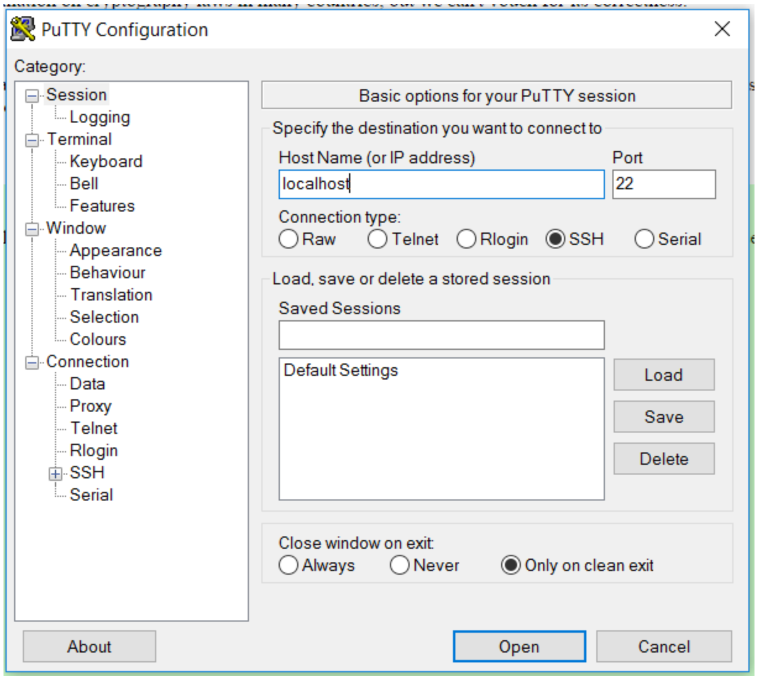
Domain:

User can use: ☑ Shell ☑ SFTP ☑ Tunneling

- Select the ***Server Status*** tab.
- Click to start the SSH server (if it is not already running).
- Open a terminal and run ***netstat -an | more***. Note: the SSH server is now listening on port 22.
- To exit netstat type ***Q***.

4.3. Download and Run an SSH Client for Windows

- Visit www.chiark.greenend.org.uk/~sgtatham/putty/download.html.
- Download ***putty.exe*** for Windows and click the file to start PuTTY.
- Enter ***localhost*** or ***127.0.0.1*** (both refer to your local computer or you could enter your host's external IP address) under Host Name and click on ***Open***.



Specify the destination you want to connect to

Host Name (or IP address): localhost

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

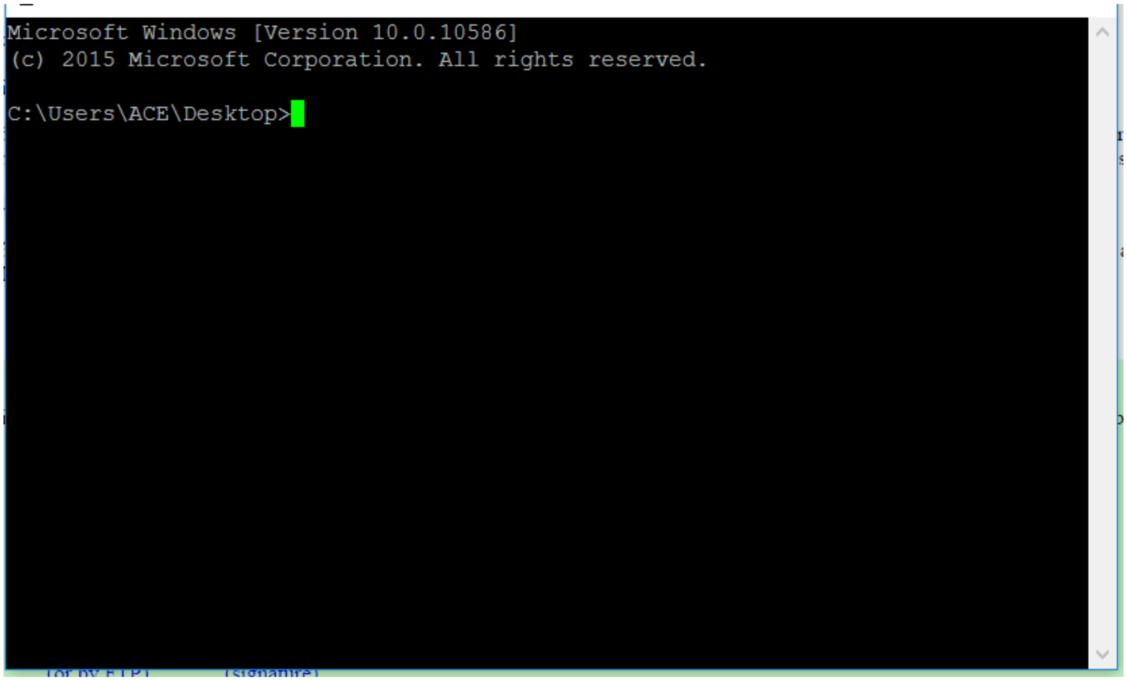
Load, save or delete a stored session

Saved Sessions:

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

- Click ***Yes*** when you get the Security Alert.
- Enter the user name and password you specified to get terminal access via SSH.
- At this point, you are accessing the Windows machine over SSH. You can now execute commands on the windows host over the SSH protocol.



```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\ACE\Desktop>
```

- Type ***dir*** to see the folder contents.
- Type ***net user*** to list the users on the machine.
- Type ***exit***.
- Close the PuTTY window if it is still open.

4.4. Download and Install an SH Server on Linux

Complete these steps on your Linux VM.

4.4.1. Download and install openssh

- Install the ***openssh-server*** package on your Linux image (with Synaptic Package manager or apt-get).
- Open a terminal:
  - ***netstat -an | more*** (Note: Port 22 is listening)
  - Type ***q***
  - ***ssh aceintern@localhost*** (Log into the openssh server with aceintern account. Note: Ubuntu OS has an ssh client installed by default.)
  - Enter ***yes*** to the security warning.
  - Enter your password.
  - You are now accessing the Linux VM over SSH.
  - To quit, type ***exit***.

4.4.2. Configure openssh

- Open a terminal:
  - ***sudo -s*** (Escalate privileges).
  - ***cd /etc/ssh*** (Remember the /etc directory holds configuration information).
  - ***cp sshd_config ssh_config***~ (Make a backup of the open ssh server configuration file).
  - ***gedit sshd_config***.
- Review the ssh server configuration file. Change the listening port from 22 to 333. Save the file and exit.
  - ***service ssh reload*** (restart the server to load the configuration file).
  - ***netstat -an| more*** (Note: Port 22 is closed, port 333 is listening).

- Type ***q*** to exit netstat.
- ***ssh aceintern@localhost*** (This will fail. We need to tell the SSH client to use port 333.)
- ***ssh -p 333 aceintern@localhost*** (Login on port 333.)
- Enter ***yes*** to the security warning.
- Enter your password.
- You are now accessing the Linux VM over SSH.
- ***Exit***.
- ***cp sshd_config~ sshd_config*** (Copy backup file over changed file to change listening port back to 22.)
- ***service ssh reload*** (Restart the server to load the configuration changes.)
- ***exit*** (leaves the root prompt)
- ***exit*** (closes the terminal)

4.4.3. Stop, Start, and Restart Servers (like openSSH)

- Open a new terminal (Ctrl+Alt+t).
- ***sudo -s***
- ***service ssh stop*** (Stop openSSH server)
- ***service ssh start*** (Start openSSH server)
- ***service ssh reload*** (Restart openSSH server)

Note: these changes are typically temporary. For example, if you stop the openSSH server and restart the system it will automatically start unless disabled (for example see Section 4.2.2).

4.5. Issue Commands to a Remote System

4.5.1. Reminder: How to Determine IP Addresses

- On Linux open a terminal and type ***ifconfig***.
- On Windows open a terminal and type ***ipconfig*** (interface is typically Local Area Connection).

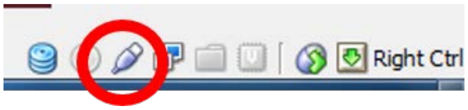
4.5.2. SSH into Linux from Windows

- Open PuTTY.
- Enter the Linux IP address and click ***Open***.

- Enter username ***aceintern*** and password ***password*** if prompted.

5. Servers and Clients: FTP on Linux

- Install the ***vsftpd*** package on your Linux Image (Synaptic Package Manager or apt-get).
- Open a terminal.
- ***cd*** ~ (Move to your home directory).
- ***netstat** -an* |***more*** (Note that port 21 is listening).
- Type ***q*** to exit netstat.
- ***echo hi*** > ***sample.txt*** (Create a file named sample.txt).
- ***ftp localhost*** (Log into the openSSH server with the iaintern account).
    - If you receive an error saying that the ***vsftpd*** install failed, switch your VM from NAT (Network Address Translation) to Bridged. If your network type is set to NAT then the VM hides behind the IP address of the VirtualBox host. If your network type is Bridged then the VM appears as if it was a physical host on the network and requires its own IP address from the network. To change your VM from NAT to Bridged:
        - Right click the Network Adapter icon in the bottom right of your screen and select Network Adapters



        - Under Attached to select "Bridged Adapter" then click ***OK***
        - Reinstall ***vsftpd***
- Enter your username and password
- ***pwd*** (List the present working directory)
- ***dir*** (List contents of directory)
- ***help*** (View available commands)
- ***put sample.txt*** (Upload sample.txt to the server)
- Open a new tab on the terminal
- ***cd*** ~ (Move to your home directory)
- ***rm sample.txt*** (Delete sample.txt on your local computer. Return to tab with ***ftp*** session)
- ***get sample.txt*** (Download sample.txt from the server)

- ***exit*** (End your session)

6. Review Exercises

6.1. You used a web browser and Abyss. Identify the client and the server.

6.2. Can a single computer act as a web client (browser) and a web server at the same time?

6.3. With the proper software, can any typical Internet-connected, functional computer serve as web pages?

6.4. Can the HyperText Transport Protocol transport anything besides HyperText Markup Language?

6.5. You run *netstat* from the terminal and notice that an application is listening on port 80. Does this indicate a web server running on the machine? Why or why not?