LESSON NOTES

Intro to Linux

Security

2.4.1 Secure Shell Protocol (SSH)

Lesson Overview:

Students will:

- Understand what SSH is and how it is used

Guiding Question: What is SSH and how is it used?

Suggested Grade Levels: 9 - 12

Technology Needed: None

CompTIA Linux+ XK0-005 Objective:

2.4 - Given a scenario, configure and execute remote connectivity for system management

- SSH
  - Configuration files
    - /etc/ssh/sshd_config
    - /etc/ssh/ssh_config
    - ~/.ssh/known_hosts
    - ~/.ssh/authorized_keys
    - ~/.ssh/config
  - Commands
    - ssh-keygen
    - ssh-copy-id
    - ssh-add
  - Tunneling
    - X11 forwarding
    - Port forwarding
    - Dynamic forwarding

This content is based upon work supported by the US Department of Homeland Security's Cybersecurity & Infrastructure Security Agency under the Cybersecurity Education Training and Assistance Program (CETAP).

Secure Shell

SSH, or *Secure Shell*, stands as a robust protocol offering a secure means of connecting to and managing remote machines over a potentially insecure network. Its architecture involves a set of configuration files, commands, and tunneling capabilities that collectively enhance the security and functionality of remote connections.

SSH configuration relies on several key files, including **`/etc/ssh/sshd_config`** for server-side settings and **`/etc/ssh/ssh_config`** for client-side configurations. User-specific files like **`~/.ssh/known_hosts`** store host keys, while **`~/.ssh/authorized_keys`** lists authorized public keys for server access. Additionally, **`~/.ssh/config`** serves as a personal configuration file for customizing SSH behavior.

The SSH suite includes essential commands like **`ssh-keygen`** for generating secure key pairs, **`ssh-copy-id`** for facilitating passwordless logins by copying public keys to remote servers, and **`ssh-add`** for managing private keys through the SSH authentication agent.

SSH extends its utility with tunneling options such as *X11 Forwarding*, enabling graphical application forwarding from a remote server to the local machine. *Port Forwarding* redirects network traffic, while *Dynamic Forwarding* establishes a SOCKS proxy for secure browsing and traffic forwarding. These tunneling features enhance the versatility of SSH in various networking scenarios.