Implementing SSH

Using Telnet to control your network devices is the same as shouting your username and password when you exit a building. Pretty soon, someone is going to be listening and take advantage of your lack of security. Secure shell or SSH is the industry standard replacement for telnet and other remote console management applications. SSH commands are encrypted and secure in several ways.

The client (your workstation) and the server (your network device) connections are authenticated using a digital certificate, and passwords are protected by encryption. SSHv1 uses Rivest, Shamir, and Adelman (RSA) keys and SSHv2 uses Digital Signature Algorithm (DSA) keys to secure both the connection and authentication. Encryption algorithms include Blowfish, DES, and 3DES. SSH protects and prevents against "spoofing", man in the middle attacks, and packet sniffing.

Securing your connection

The first step is to logon to your router or switch and determine if you have an IPSec IOS imaged loaded that supports SSH. For our examples, we'll use Cisco IOS commands and run the command:

Router> Show flash – This will display the name of the IOS image that is loaded and you can compare that with your vendors’ list of supported features.

After you've verified that your device supports SSH, ensure that the device has a hostname and host domain properly configured:

Router> config terminal
Router (config)# hostname hostname (the name of the router)
Router (config)# ip domain-name domainname (a domain that the router services)

At this point you’re ready to enable the SSH server on the router. To enable the SSH server, you must first generate an RSA key pair with the command:

Router (config)# crypto key generate rsa

When you generate an RSA key-pair for the router, you automatically enable SSH. When you delete the RSA key-pair (using the no crypto key generate rsa command), you automatically disable the SSH server.

The last step to implementing SSH is to enable Authentication, Authorization and Accounting (AAA). When you configure AAA, you’ll specify usernames and passwords, the session timeout and the number of retries allowed during a connection attempt. Use the global commands:
Router (config)# aaa new-model
Router (config)# username username password password
Router (config)# ip ssh timeout seconds
Router (config)# ip ssh authentication-retries integer

To verify that SSH has been configured and is running on the router, give the command:

Router# show ip ssh

**Use SSH**

You’re now ready to force the users that you added during the AAA configuration to utilize SSH instead of Telnet, by requiring SSH for virtual terminal connections, with the following commands:

Router (config)# line vty 0 4
Router (config-line)# transport input SSH

All you need now is a SSH terminal client program to test your configuration before you kill the current Telnet session. I highly recommend PuTTY. It is free and an excellent terminal emulator.

**Final thoughts**

After you’ve enabled SSH on your routes and switches, make sure you modify any existing access control lists to allow the connections to these devices. You can now report to your superiors that all network management sessions are encrypted and secure and that is huge security hole filled.