Global Information Assurance Certification Paper

Using SSH To Manage Your Network Gear

Donald Marsee

GIAC Security Essentials Certification (GSEC)
Version: 1.4b (Option 1)

April 14, 2004

Table of Contents

Abstract

It has been my experience that system administrators and network administrators often use whatever tool is available and easy to use. We have more than enough work on our plates and do not always have time to be concerned with security. If a segment of the network is dropping all packets, I am not going to be too concerned with the security (or lack thereof) of the telnet session I use to connect to the faulty router – I just want to connect to the router or switch and begin troubleshooting. The same is true if a web server has problems; I just want to make a connection and fix the problem. In this paper I will suggest that systems and network administrators can have the connectivity they require and have it in a much more secure fashion than using the common telnet and r-tools from Berkeley such as rlogin. In addition, systems and network administrators often use ftp and/or rcp to move data around the network. This functionality can also be had in a much more secure way.

There are several options when looking at SSH solutions ranging from open source to full blown commercial products. I will explore what it takes to install, configure and use SSH to manage the Network Infrastructure (routers, switches, etc.) This will include configuring SSH to function on Cisco's network gear as an SSH server and on Windows PC's as an SSH client.

Background and Foundation for SSH

SSH enables remote login as well as other secure network services over an un-trusted network. SSH came about when a researcher, Tatu Ylonen, at a Finland university discovered that passwords were being "sniffed" on the university network. This happened in 1995 – only nine years ago! That makes SSH a relatively young concept when compared to other network concepts such as TCP/IP. Apparently, Tatu thought it would be a good idea to prevent passwords from traveling across networks in clear text. It seems pretty obvious today that clear text passwords are a bad thing; but Tatu was among the first to actually take steps to prevent them. Initially he released the source code as freeware; however, fairly quickly – within months – he formed a company that he called SSH Communications Security. That company is still going today and is releasing updates and additional products and services.

There has been some controversy around SSH. In 2001 Tatu took exception to OpenSSH using SSH as part of it name. [1] He asserted that people were getting his company and OpenSSH confused. It has been a little over three years since Tatu's open letter to the OpenSSH mailing list and OpenSSH is still OpenSSH. So, one can safely assume that his letter was not well received by the leaders of the OpenSSH movement. I must admit that when I began researching this subject it took a while for me to get a good grasp of the differences. One of the "confusing" concepts associated with SSH is how does

one describe it. SSH is a protocol. But wait, SSH is also an application. Therefore, when I am referring to the SSH protocol, I will explicitly say SSH protocol. SSH by itself will be used to talk about the application.

Quite a bit occurred early in the evolution of SSH. The first version, which has come to be known as SSH1, had some issues (with a RSA patent that no longer applies and with a CRC bug) and was succeeded by SSH2. However, the two are not compatible and SSH2 did not have all of the features that SSH1 had which created a situation where SSH2 was not accepted completely. With more recent updates and improvements, "…SSH2 is becoming the standard when referring to SSH." [2]

External networks are obviously not to be trusted. However, in most cases it is also wise to treat an internal network as un-trusted or insecure. Not too long ago, the prevailing thought among systems and network administrators (administrators) was that if the system is not "touching" the Internet, then there is no need to be worried about security. Of course, this mindset is changing rapidly in today's IT climate. For example, with the increasing number of laptops it is impossible to be sure that all PC's that connect to the internal network are "clean." Also, many businesses are allowing vendors or home users to connect into the internal network via a VPN connection. Most likely, the administrators have no control over the other end of the VPN connection. Unfortunately, administrators also have to alert to the possibility that a person on the inside could be the source of the next attack. This is especially true in large or medium size companies where users have a feeling of anonymity.

Managing Network Devices

The foundation of any network is the network gear (routers & switches) that connect the users and servers. The initial configuration of this equipment is often done through the console port and is obviously very secure. However, after this equipment is up and running, geographical distance or convenience usually dictates the use of some remote login tool. Telnet and even rlogin can work really well here – but the connections are not secure, not even close. This is a great situation for using SSH.

Configuring SSH on a Cisco Router

Until recently, only SSH1 has been widely implemented for Cisco's routers. However, "SSH Version 2.0 (SSHv2) support was introduced in some IOS platforms/images starting in 12.1(19)E." [3] Cisco added support for SSH2 in all IOS platforms/images in a later IOS Release as well as support for the SSH2 client. "…the SSH Version 2 server is supported in Cisco IOS Release 12.3(4)T; the SSH Version 2 client is supported beginning with Cisco IOS Release 12.3(7)T." [4] Other restrictions and prerequisites exist when using SSH server and SSH client on Cisco routers.

- Must be running an IPSEC (DES or 3DES) encryption software image found in IOS Release 12.1(1)T for the SSH server. SSH client requires IPSEC and IOS Release 12.1(3)T.
- RSA authentication found in SSH clients is not supported in the SSH server.
- The only application supported is the Execution shell.
- Must configure a host name and a host domain. (Can be omitted with SSH2 implementation.)
- Compression is not supported.
- An RSA key pair needs to be generated. (This step actually enables SSH on the router.)

Here are sample commands that show how a Cisco router is configured for SSH.

MyRouter(config) # hostname *HOSTNAME*
MyRouter(config) # ip domain-name *DOMAINNAME*
MyRouter(config) # crypto key generate rsa
Note: *You will be prompted to enter a modulus length. Cisco recommends using a minimum of 1024. A longer key is more secure but increases the overhead. Maximum length is 2048.*
MyRouter(config) # ip ssh version 2
Note: *This command is only available if you are running an IOS Release that supports SSH2. If supported and this command is not used, the server defaults to run in compatibility mode which means that both SSH1 and SSH2 are accepted.*

SSH server is now active on the router. The default parameters will be used unless you make changes. You can also view the status of SSH and SSH connections on the router. See below for command syntax.

MyRouter(config) # ip ssh {[timeout *seconds*] [authentication-retries *integer*]}
Note: *Timeout setting only apples to the SSH Negotiation. After EXEC session begins, the vty timeout applies. The default number of retries is three with a maximum value of five. Also, the default number of defined vty's is five and this limit applies to SSH sessions.*

MyRouter(config) # show ip ssh
Note: *The output of this command will show the version and configuration data if SSH has been enabled. Otherwise, it will just show that SSH is not enabled.*

This is all well and good. But what if the network administrator still wants to use telnet to connect to the router you just configured for SSH? Old habits are hard to break! There is a solution for this – force SSH to be the only way to remotely connect. That can be accomplished with the following commands.

MyRouter(config) # line vty 0 4
MyRouter(config-line) # transport input ssh
Note: *This command can also be used to enable SSH Terminal-Line Access. "The SSH Terminal-Line Access feature replaces reverse Telnet with secure shell (SSH)." [5] This feature can be used to connect to console ports of remote devices that normally do not support telnet. When combined with SSH this feature becomes secure.*

There are a couple of other commands that are useful to know when discussing SSH and Cisco routers. On some occasions you may need to disconnect or kill a SSH session. This can be done two ways: **disconnect ssh [vty] *session-id*** or **clear line vty *n***. Here are examples of each.

MyRouter(config) # disconnect ssh 2
MyRouter(config) # clear line vty 2
Note: *The session-id or n (2 in both examples) can be determined by using the **show ip ssh** command. Be careful to not cut your feet out from under yourself and kill the session you are working from! The SSH connection will die when the EXEC session ends (normally or abnormally.)*

What about copying files? To enable or disable Server-Side SCP, use this command: **[no] ip scp server enable**. This is disabled by default.

New commands/features were introduced along with support for SSH2. I covered the **ip ssh version** command earlier. To give the administrator more flexibility, the **ip ssh rsa keypair-name** command has been added. This gives the administrator more control over when SSH is enabled and the generation of the RSA keys. This can also remove the requirement to configure a hostname and a domain-name. Finally, the **ssh** command has been updated with more options to take advantage of the increased security that SSH2 brings to the table.

The SSH client is available only after the SSH server is up. The SSH client can be used to connect to another router or device as long as SSH server is enabled on the target machine. You can actually connect to another vendor's device such as a 3Com router.

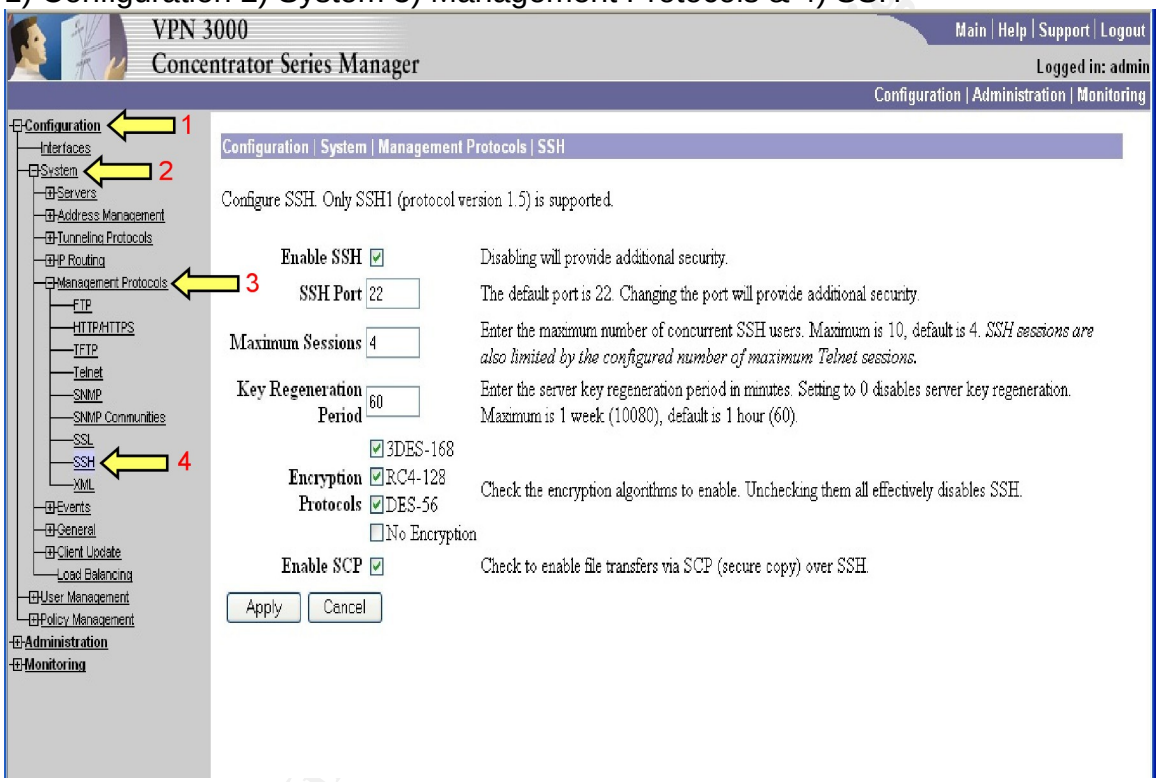
Configuring SSH on a Cisco Catalyst Switch running CatOS
The requirements are similar to the Cisco router. Support for SSH1 was introduced in CatOS 6.1. While SSH1 is not as advanced as SSH2 for speed and security, it is much better than the traditional telnet or Berkeley r-tools. As with the Cisco router, it is required that the software image supports IPSec (DES or 3DES) encryption. Once the proper OS is in place, the RSA key must be generated: **set crypto key rsa 1024**. It is also a good idea to restrict SSH to authorized hosts (or networks): **set ip permit 172.16.10.0 255.255.255.0**. Note

that I used an RFC 1918 address – modify the command to fit your network scheme. Finally, enable SSH with this command: **set ip permit enable ssh**.

Configuring SSH on a Cisco VPN Concentrator

Done! Well there is a little more to add. SSH support is standard on Cisco's VPN Concentrators. The web interface works best when for making configuration changes. Once connected with your favorite browser (since we are talking about security here, please connect over SSL!) navigate to the page as illustrated below.

1) Configuration 2) System 3) Management Protocols & 4) SSH



As you should see, SSH is enabled by default. Also, you can modify the parameters to best fit your environment as needed. While you are here, take a look at the other management protocols; you may more a more secure setup than the default. For instance, you can gain additional security by configuring the **HTTP/HTTPS** server to only enable HTTPS. Given the subject at hand, I strongly suggest visiting the **Telnet** page and unchecking the "enable Telnet" checkbox. Finally, you may want to disable **SNMP** support – you cannot use SNMP to make configuration changes anyway.

Configuring SSH on a Cisco PIX Firewall

"In October of 1995, Cisco Systems, Inc. began their first serious push into the Network Security market with the acquisition of NTI (Network Translation, Inc.). NTI's flagship PIX firewall became the Cisco Secure PIX Firewall." [6] The article goes on to say that until 2000 there was no way to have secure remote access to

the PIX. In 2000, Cisco released version 5.2 of the PIX OS which included support for SSH. There are two methods that provide secure, remote access to a PIX Firewall – IPSec and SSH. As you might guess, I'll discuss SSH.

PIX OS uses either a DES or a 3DES cipher. In order to use 3DES you must have the appropriate activation key. You can register for a free 3DES/AES IPSec VPN Encryption License at Cisco's web site (registered users only.) If your PIX is running PIX OS 6.2 or higher, just enter the new activation key via the **activation-key** command. To make the activation key active, you must do a reload. You must download a software image to your PIX Firewall in monitor mode if it is running PIX OS 6.1 (or earlier) in order to enter the new activation key.

PIX Firewalls use SSH version 1 (SSH1) only. However, the known deficiencies of SSH1 have been patched since PIX OS version 5.3(2). Be that as it may, I expect Cisco to add support for SSH2 which is surely becoming the preferred version and standard.

Enabling SSH on the PIX Firewall should seem familiar at this point. But, of course, there are some differences. The first three steps closely match those of the Cisco router.

pixfirewall(config) # hostname *MyPix*
MyPix(config) # ip domain-name *cisco.com*
MyPix(config) # ca generate rsa key 2048
Note: *Requesting a modulus key of 1024 or greater may take a few minutes according to the* ***help ca*** *command. A similar warning is issued when you actually execute the command.*

Other commands are just close enough to help make an administrators life even more complicated. We can hope that future OS's from Cisco will converge; but I would not count on it. In any event here are a few more commands to use with SSH on your PIX.

MyPix(config) # show ca mypubkey rsa
MyPix(config) # ca save all
Note: Take a look at your key and then save it! *You will lose your keys if you forget this step prior to executing a reload!*
MyPix(config) # ssh 172.16.1.1 255.255.255.0 inside
MyPix(config) # ssh timeout 60
Note: These above commands indicate which subnet is allowed to make an SSH connection and how long (in minutes) an SSH session can be idle before being disconnected.

It is likely (I hope) that you already have set the "enable password." If not you should do that now. Also, unless previously done, you must set the Telnet password; it is used to authenticate the SSH session.

```
MyPix(config) #  enable password 2ez2breal
MyPix(config) #  passwd t31N3t
```

Your PIX Firewall should be ready and willing to service connections from SSH clients now.

Installing and Using SSH Clients

There are several good SSH clients to choose from for Windows computers. Some are free and some are commercial. I down loaded and installed a few of them to get a feel for their pros and cons. Depending on when you are reading this, there is no telling how much the SSH landscape will have changed. Go to your favorite search engine and search for "SSH client Secure Shell." You will notice that a couple sites will appear more that any others. Tatu Ylonen's company (SSH Communications Security) and OpenSSH are the dominant players today, especially in the SSH Server arena. There are a few others that show up consistently in the searches as well – PuTTY, F-Secure, and SecureCRT. Let's discuss each of them and describe any configuration settings that are useful when connecting to Cisco network gear.

SSH Communications Security (SCS)
http://www.ssh.com/
SCS recently changed the names of their products. Their product line now includes SSH Tectia as the first part of every offering. This includes the SSH client which is now referred to as **SSH Tectia Client**. It was formerly known as SSH Secure Shell for Workstations. After providing registration information you can download an evaluation version which is good and fully functional for over two months.

The install is very straightforward. You are given some options. You can choose to not install Desktop Icons, Documentation and Command Line Tools. There are four command line tools: scp2.exe, sftp2.exe. ssh2.exe and ssh-keygen2.exe. You also have the option of not adding the Command Line tools to the PATH environment variable on the same screen. When the install completes you will have a new program group – SSH Tectia Client. Within that group will be the **Terminal Client** and **File Transfer Client**. Both clients are well laid out and intuitive to use. When just starting out, you can use the **Quick Connect** button and then easily save that connection by using the **Profiles** button. X11 port forwarding or tunneling can be enabled with one checkbox. You do have to have a separate X server running on your PC to take advantage of graphic connections. You also can configure tunneling for other ports as needed.

The SSH Tectia Client will handle both SSH1 and SSH2 connections. The default configuration setting is to give a warning when a connection is made using SSH1. You can opt to always deny or always accept SSH1 connections if desired. There are numerous other settings that you can modify. Many relate to the look and feel of the client which allows you to personalize based on your preferences. Other settings deal with user and server authentication and can really affect the level of realized security especially when connecting to a full-blown SSH2 server. You can also capture or log the screen output from your session to a text file.

F-Secure SSH
http://www.f-secure.com/
You can also download a free 30-day evaluation version of F-Secure SSH Client 5.X. You do have to register first. The install process is simple and there is very little reason to not just accept the defaults. Once installation completes, you have a new Program Group – **F-Secure SSH Client Trial**. It seems F-Secure wants to continually remind you that this is a trial! Within that group you will find two programs: **F-Secure SSH Client Trial** and **F-Secure SSH File Transfer Trial.** There is also a **Readme** and an **F-Secure SSH Client Help** in the group.

When you first execute the SSH Client or the SSH File Transfer programs, you are prompted to enter a license key code; actually you will get this prompt every time you start either program. This is a little annoying but it is a commercial product. You can just click on the Try button to begin the evaluation.

Now you are presented with the user interface which very much resembles the SSH Tectia product. It turns out that F-Secure and SSH Communications Security (SCS) had a partnership of sorts for a while. If you look at the Help/About… menu with F-Secure's client, SCS is listed as a copyright holder. It appears that relationship has ceased. SCS's financial statement from 2003 sheds some light: "Gross margin for the fourth quarter rose to nearly 100 percent as the royalties paid to F-Secure Oyj were terminated." [7]

While the two products look very similar, there are differences. F-Secure SSH Client has fewer settings to configure. This client does support SSH1 and SSH2 connections; however, the default setting to always allow. The setting can be set to warn or always deny. The configuration settings for user and security authorization are much more robust on the SSH Tectia client.

F-Secure does offer command line tools functionality: scp2.exe, sftp2.exe and ssh2.exe. If you evaluate both of these clients on the same PC be careful to know command line tools are found first. Creating and saving profiles works about the same in both products. In the end the F-Secure client software seems to be a lagging behind. It is noteworthy to add that F-Secure has a greater

breadth of products from anti-virus to security policy managers. SCS is focused primarily on SSH which is the brainchild of its founder.

PuTTY
http://www.chiark.greenend.org.uk/~sgtatham/putty/
This client is a free download and free to use. It also has a smaller footprint. The install file is a little over 1MB and the executables are equally compact at less than 2MB. As you should expect, there are not as many bells and whistles. However, it does the job.

The install process is extremely simple. Again, there are a couple options during install; just go with the defaults. Once the install finishes, you will have a new Program Group – PuTTY. Within that group will be a few programs: Pagent, PSFTP, PuTTY, PuTTY Manual, Putty Web Site and PuTTYgen. PSFTP is a command line tool for file transfer. There are other command line tools as well: **plink** can be used within batch files and **pscp** is a secure file copy. The layout of the Pagent screen reminds me of the Settings screen from SSH Tectia and F-Secure SSH client. For PuTTY though, the Pagent acts as a launching pad for various connections. PuTTY has very good options for logging including the logging of all SSH packet data.

SSH1 and SSH2 are handled and you can select to prefer SSH2 over SSH1 and vice versa OR you can dictate SSH1 only or SSH2 only. You can also configure the order in which encryption ciphers are tried. There can also be a warning threshold set to alert you when certain encryption ciphers are used.

Miscellaneous

ASG MobileControl Administrator
http://www.asg.com/
This client is primarily designed to support Windows servers. However it has a very cool feature that applies here. I haven't jumped on board the PDA bandwagon just yet, but I am intrigued by this functionality. "Using the ASG-MobileControl Administrator Telnet and SSH interfaces, an administrator can remotely access and remotely execute commands on devices such as routers, switches, and servers supporting these protocols." [8] I can see this becoming more accepted as we continue to become a more mobile society. I'm sure we'll be seeing more and more PDA's accessing our networks. I hope the SSH clients continue to be considered as well.

Summary

When can you trust the network? With a slight fear of being paranoid, I suggest that no network can be completely trusted. This does not include a network I can set up in my lab and I know all the endpoints and every device in between. If you have a real network, a useful network, then most likely you do not have control

over or even know users on the other end or know the network infrastructure in the middle. That network gear still has to be managed and to do that you must be able to easily and efficiently connect to them. That is why SSH is so important. The Cisco gear I work with all support at least SSH1. It is relatively easy to configure and use. Why not use it? Many administrators may answer that much as I would have a short time ago – I did not realize Cisco's Network gear support SSH. In addition, I didn't realize the strides SSH has made since Tatu birthed it back in 1995. We certainly can't say cost is a factor. The SSH server is already on most of the network equipment and there are free clients – that work well. Even if you purchase a commercial version to get support and maintenance, that cost is minimal compared to the cost of the equipment you are using it on. And what is the cost of having someone be able to see your password in clear text? Some risks have to be taken – not this one.

References

[1] “Tatu Ylonen requests OpenSSH to change its name”. From Linux News on the Linux Today web site.
http://linuxtoday.com/news_story.php3?ltsn=2001-02-14-003-04-NW-SW-BD

[2] Dwivedi, Himanshu. Implementing SSH. Indianapolis: Wiley Publishing, Inc., 2004. 4

[3] “Configuring Secure Shell on Routers and Switches Running Cisco IOS”. From the Cisco Systems web site.
http://www.cisco.com/warp/public/707/ssh.shtml

[4] “CISCO IOS SOFTWARE RELEASES 12.3 T - Secure Shell Version 2 Support”. From the Cisco Systems web site.
http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00802045dc.html

[5] “CISCO IOS SOFTWARE RELEASES 12.2 T - SSH Terminal-Line Access”. From the Cisco Systems web site.
http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide09186a0080087acb.html

[6] “Configuring the PIX Firewall for SSH (Secure Shell)”. By David Chapman, Jr. From InformIT web site.
http://www.informit.com/articles/printerfriendly.asp?p=25342

[7] “Financial Statements Bulletin for January 1 - December 31, 2003”. From the News Room section of the SSH Communications web site.
http://www.ssh.com/company/newsroom/article/506/

[8] “ASG-MobileControl Connection Center”. From the ASG Software Solutions web site.
http://mobilecontrol.asg.com/default.asp?pg=wmc