SSH in Solaris

Abstract

In this paper I will explain about SSH in Solaris. I will give a short introduction and summary of ssh principles and demonstrate operation of keys generation and public key interchange(PKI), along with relevant commands.

Introduction

ssh (secure shell) is a program for logging into a remote machine or for executing commands on a remote machine . Is intended to replace rlogin and and rsh and to provide secured and encrypted communication between two trusted hosts over an untrusted network.

System requirements

Ssh comes by default in Solaris 9/10 – sparc or x86/x64 platforms.

Solaris 8 - you will need to add ssh packages from http://www.sunfreeware.com or equivalent site.

Useful commands

#key generation with dsa encryption on client side and server side for dpecific user !!!

ssh-keygen –t dsa –N """ - generating dsa key

ssh <host1> - login in to a **host1**.

ssh <hostname> ls – performing ls on the server **host1**

scp (host1:)<source> (host2:)<destination> - copying from **host1**, source file or directory to destination host2, file or directory for example:

Copyright © 2009 Grigale Ltd. This document is licensed under the terms of Creative Commons Attribution 3.0 License

scp cango2:/etc/passwd . - copying **/etc/passwd** on cango2 to the current directory

scp /etc/hosts cango1: - copying **/etc/hosts** on the local server to the home directory on cango1.

User specific configuration files

$HOME/.ssh- a directory which contain private/public keys and authorized_keys file

$HOME/.ssh/id_dsa -

$HOME/.ssh/id_dsa.pub –

$HOME/.ssh/known_hosts - this file contain all the fingerprints (public key of the hosts) of servers which we already logged in .

$HOME/.ssh/authorized_keys – a file which contain all the hosts which we allow to connect without a password .(PKI)

System wide configuration file

/etc/ssh/ssh_host_rsa_key.pub – this file represent the host when a user is trying to connect to the host . (Fingerprint of the hosts) when the fingerprint of the host is changed there is a big probability that ssh will refuse to connect to this host and the ssh client software will tell that the fingerprint is changed and that someone is probably something nasty !! – in this situation you must remove the entry that contain the fingerprint of this host from the file - known_hosts

/etc/ssh/ssh_config – client configuration file . (authentication methods ,ssh version)

/etc/ssh/sshd_config – a server configuration file . (PermitRootLogin ,X11Forwarding)

/etc/issue – a banner file which contain a message for the user before logging in .

Copyright © 2009 Grigale Ltd. This document is licensed under the terms of Creative Commons Attribution 3.0 License

Best practice –Public key interchange between two Solaris servers (For root or any specific user).

1. Log in as root and check that **/etc/ssh/sshd_config** is configured to have
   - X11forwarding – set to **yes**
   - RSAauthentication – set to **yes**
   - PermitRootLogin – set to **yes**
2. check that **/etc/ssh/ssh_config** is configured to have
   - ForwardX11 – set to **yes**
   - PubkeyAuthentication – set to **yes** (usully default on solaris 10)
3. Generate the public key by typing :

ssh-keygen –t dsa –N ""

4. Copy the public key to the name of the server with suffix .pub by typing :

cd $HOME/.ssh/

cp id_dsa.pub cango1.pub

5. Copy the public key to the remote server using the command:

scp cango1.pub cango2:.ssh/

6. Append the public key file(on cango2) to a file called authorized_keys by typing:

cat cango1.pub >>authorized_keys

7. Now you can login as root without a password or run a command on remote host.

Example remote login: **ssh cango2**

Example remote command: **ssh cango2 ls**

Example copying a file: **scp file1 cango2:dir1**

Example copying the directory dir1 from cango2 to the current directory:

scp -r cango2:dir1 .

Copyright © 2009 Grigale Ltd. This document is licensed under the terms of Creative Commons Attribution 3.0 License

Example logging in as a different user: **ssh rafi@cango2**

Example copying as another user: **scp file1 rafi@cango2:dir2**

Example script using ssh:

```
#!/usr/bin/bash
HOSTS="server1 server2 server3"
for host in $HOSTS
do
ssh $host ls /tmp
done
```

Several public key authentication familiar problems:

1. Check that any user other than root does not have uid = 0 or gid = 0 .

   if a user(other than root) have gid = 0 – this user can not log in without a password, unless it's gid is set to number other than 0.

2. Check that **.ssh** directory permissions are 700 – Unless permissions are set to 700, sshd will prevent public key authentication.

Copyright © 2009 Grigale Ltd. This document is licensed under the terms of Creative Commons Attribution 3.0 License