OPERATING SYSTEMS
CCGC-5000

Module - 12

Agenda – Module - 12

Authentic information is available from the given resources in course outline and URL's mentioned from this slides, and this presentation is only supportive document to be read with the given resources and corrected accordingly if required..

- openSSH
- openSSH commands – ssh, scp, sftp
- Secure system management with ssh
- Secure copy with scp
- Secure ftp with sftp

Course Book Reference : Chapters 19

OpenSSH

- Secure Shell (SSH) provides a secure mechanism for data transmission between source and destination systems over IP networks.
- *OpenSSH* is a free, open source implementation of proprietary SSH.
- The secure ***ssh*** command has replaced *telnetd, rlogin, rsh*, and *rexec*
- *rcp* and *ftp* are replaced to as ***scp*** and ***sftp***, respectively
- It was designed to replace the old remote login programs that transmitted user passwords in clear text and data unencrypted.
- SSH uses encryption techniques to secure a communication channel and employs digital signatures for user authentication.

OpenSSH

- Latest version of OpenSSH is 8.8 (https://www.openssh.com/)
- Latest version of ssh protocol is v2
- ssh protocol v2 supports RSA, DSA and ECDSA (a new variant of DSA)
- RSA includes the support for both encryption and authentication
- DSA and ECDS provides digital signature

**RSA – Rivest-Shamir-Adleman, DSA – Digital Signature Algorithm, ECDSA – Elliptic Curve Digital Signature Algorithm*

- Authentication Methods
  - GSSAPI-based *(Generic Security Service Application Program Interface)* Authentication
  - Host-based authentication
  - Public key-based authentication
  - Challenge-response authentication
  - Password-based authentication
- Required reading – Chapter 19

OpenSSH

- OpenSSH packages :
  - openssh
    - provides ssh-keygen command & supported library routines
  - openssh-clients
    - includes commands – scp, sftp, slogin, ssh, ssh-copy-id & client configuration file
  - openssh-server
    - contains sshd daemon, server configuration file & library routines
- OpenSSH service : **sshd** should be enabled and started
- Firewalld service: **ssh** should be added to firewalld services
- Default server configuration file : **/etc/ssh/sshd_config**
- Default client configuration file : **/etc/ssh/ssh_config**
- Default TCP port : 22

OpenSSH

- OpenSSH packages

```
[unixuser@toronto ~]$ yum list installed openssh*
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Installed Packages
openssh.x86_64                                  6.6.1p1-31.el7                  @anaconda/7.3
openssh-clients.x86_64                          6.6.1p1-31.el7                  @anaconda/7.3
openssh-server.x86_64                           6.6.1p1-31.el7                  @anaconda/7.3
```

- OpenSSH files

```
[unixuser@toronto ~]$ ls -l /etc/ssh
total 276
-rw-r--r--. 1 root root     242153 Sep  6 12:30 moduli
-rw-r--r--. 1 root root       2208 Sep  6 12:30 ssh_config
-rw-------. 1 root root       4361 Sep  6 12:30 sshd_config
-rw-r-----. 1 root ssh_keys    227 Dec 28 00:38 ssh_host_ecdsa_key
-rw-r--r--. 1 root root        162 Dec 28 00:38 ssh_host_ecdsa_key.pub
-rw-r-----. 1 root ssh_keys    387 Dec 28 00:38 ssh_host_ed25519_key
-rw-r--r--. 1 root root         82 Dec 28 00:38 ssh_host_ed25519_key.pub
-rw-r-----. 1 root ssh_keys   1679 Dec 28 00:38 ssh_host_rsa_key
-rw-r--r--. 1 root root        382 Dec 28 00:38 ssh_host_rsa_key.pub
```

- **ssh** is used to connect and manage remote systems securely
- **scp** is used to copy files securely to/from another system
- **sftp** is used to upload/download files securely between systems
- ssh server records connects and disconnects in /var/log/secure and /var/log/messages.
- Also ssh connections can be viewed in ssh server with journalctl command

OpenSSH - ssh

TORONTO OpenSSH Client
192.168.3.4

MONTREAL OpenSSH SERVER
192.168.3.3

- Using SSH with IP ADDRESS while connecting to MONTREAL from TORONTO using the user in Montreal

```
[unixuseryyz@toronto ~]$ ssh unixuseryul@192.168.3.3
The authenticity of host '192.168.3.3 (192.168.3.3)' can't be established.
ECDSA key fingerprint is be:86:d7:87:75:db:21:a1:6f:15:6a:40:57:04:b8:f5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.3.3' (ECDSA) to the list of known hosts.
unixuseryul@192.168.3.3's password:
Last login: Sat Feb  4 21:15:09 2017 from 192.168.3.4
[unixuseryul@montreal ~]$
[unixuseryul@montreal ~]$ exit
logout
Connection to 192.168.3.3 closed.
[unixuseryyz@toronto ~]$ _
```

The names and ip4 addresses are given here as an example, actual situation it would be different.

- When connecting for first time, known_hosts file is created in home directory of user running the ssh command.
- known_hosts is created in ~/.ssh directory

```
[unixuseryyz@toronto ~]$ ll .ssh
total 4
-rw-r--r--. 1 unixuseryyz unixuseryyz 173 Feb  4 21:16 known_hosts
[unixuseryyz@toronto ~]$ cat .ssh/known_hosts
192.168.3.3 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHG/MQedtPhmTt+34MMzry18m8KFps0ZwGqyssEAgnpwGW/WtNVUFu
Sdd13njd29o+iTabmsB7IZL85RSSQQY=
```

OpenSSH - ssh

HUMBER
Faculty of Applied Sciences & Technology

TORONTO OpenSSH Client
192.168.3.4

MONTREAL OpenSSH SERVER
192.168.3.3

- Using SSH with HOSTNAME while connecting to MONTREAL from TORONTO *(need to map ipaddress with hostname in /etc/hosts file)*

```
[unixuseryyz@toronto ~]$ ssh unixuseryul@montreal
The authenticity of host 'montreal (192.168.3.3)' can't be established.
ECDSA key fingerprint is be:86:d7:87:75:db:21:a1:6f:15:6a:40:57:04:b8:f5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'montreal' (ECDSA) to the list of known hosts.
unixuseryul@montreal's password:
Last login: Sat Feb  4 21:31:29 2017
[unixuseryul@montreal ~]$
[unixuseryul@montreal ~]$ exit
logout
Connection to montreal closed.
```

- Using hostname to connect for first time ~/.ssh/known_hosts files is updated.

```
[unixuseryyz@toronto ~]$ cat .ssh/known_hosts
192.168.3.3 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHG/MQedtPhmTt+34MMzry18m8KFps0ZwGqyssEAgnpwGW/WtNVUFu09
Sdd13njd29o+iTabmsB7IZL85RSSQQY=
montreal ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHG/MQedtPhmTt+34MMzry18m8KFps0ZwGqyssEAgnpwGW/WtNVUFu09Sdd
13njd29o+iTabmsB7IZL85RSSQQY=
```

OpenSSH - scp

TORONTO OpenSSH Client
192.168.3.4

MONTREAL OpenSSH SERVER
192.168.3.3

- Being logged in Toronto
  - Securely copying files from Montreal to Toronto

```
[unixuseryyz@toronto ~]$ ll file*
-rw-rw-r--. 1 unixuseryyz unixuseryyz 105710 Feb  4 21:45 file1.yyz
-rw-rw-r--. 1 unixuseryyz unixuseryyz   4563 Feb  4 21:45 file2.yyz
[unixuseryyz@toronto ~]$ scp unixuseryul@montreal:file1.yul .
unixuseryul@montreal's password:
file1.yul
[unixuseryyz@toronto ~]$ ll file*
-rw-rw-r--. 1 unixuseryyz unixuseryyz   5945 Feb  4 22:16 file1.yul
-rw-rw-r--. 1 unixuseryyz unixuseryyz 105710 Feb  4 21:45 file1.yyz
-rw-rw-r--. 1 unixuseryyz unixuseryyz   4563 Feb  4 21:45 file2.yyz
```

Note the period (.) to specify destination as current directory

  - Securely copying files from Toronto to Montreal

```
[unixuseryul@montreal ~]$ ll file*
-rw-rw-r--. 1 unixuseryul unixuseryul 5945 Feb  4 21:45 file1.yul
-rw-rw-r--. 1 unixuseryul unixuseryul 3251 Feb  4 21:42 file2.yul
```

```
[unixuseryyz@toronto ~]$ ll file*
-rw-rw-r--. 1 unixuseryyz unixuseryyz   5945 Feb  4 22:16 file1.yul
-rw-rw-r--. 1 unixuseryyz unixuseryyz 105710 Feb  4 21:45 file1.yyz
-rw-rw-r--. 1 unixuseryyz unixuseryyz   4563 Feb  4 21:45 file2.yyz
[unixuseryyz@toronto ~]$ scp file1.yyz unixuseryul@montreal:
unixuseryul@montreal's password:
file1.yyz
```

```
[unixuseryul@montreal ~]$ ll file*
-rw-rw-r--. 1 unixuseryul unixuseryul   5945 Feb  4 21:45 file1.yul
-rw-rw-r--. 1 unixuseryul unixuseryul 105710 Feb  4 22:20 file1.yyz
-rw-rw-r--. 1 unixuseryul unixuseryul   3251 Feb  4 21:42 file2.yul
```

OpenSSH - sftp

MONTREAL OpenSSH Client
192.168.3.3

TORONTO OpenSSH Server
192.168.3.4

- Montreal being OpenSSH Client and Toronto being OpenSSH Server
- Being logged in Montreal
  - Secure ftp from Montreal to Toronto
  - In sftp prompt, using ! before a command it executes the command in the local shell (Montreal)

```
[unixyul@montreal ~]$ sftp unixyyz@yyz
The authenticity of host 'yyz (192.168.3.4)' can't be established.
ECDSA key fingerprint is be:86:d7:87:75:db:21:a1:6f:15:6a:40:57:04:b8:f5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'yyz,192.168.3.4' (ECDSA) to the list of known host
unixyyz@yyz's password:
Connected to yyz.
sftp> ls
Desktop    Documents  Downloads  Music      Pictures   Public     Templates
Videos     file1.yyz  file2.yyz  file3.yyz  filea.yyz  fileb.yyz  filec.yyz
sftp> !ls
Desktop    Downloads  file2.yul  filea.yul  filec.yul  Pictures   Templates
Documents  file1.yul  file3.yul  fileb.yul  Music      Public     Videos
sftp>
```

Refer man sftp

OpenSSH – key based authentication

HUMBER
Faculty of Applied Sciences & Technology

MONTREAL OpenSSH Client
192.168.3.3

TORONTO OpenSSH Server
192.168.3.4

- SSH login without password by generating RSA private/public key combination for the user of the SSH server and copy the public key to the remote ssh server

1. Generate RSA private/public key using command **ssh-keygen**
   - This will create RSA keys and save it in ~/.ssh folder

```
[unixyul@montreal ~]$ ll .ssh
total 4
-rw-r--r--. 1 unixyul unixyul 177 Feb  4 23:58 known_hosts
[unixyul@montreal ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/unixyul/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/unixyul/.ssh/id_rsa.
Your public key has been saved in /home/unixyul/.ssh/id_rsa.pub.
The key fingerprint is:
e7:50:be:95:8c:bf:6d:ea:72:61:62:ed:19:4a:e5:d8 unixyul@montreal.nest253.net
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|     .           |
|      o o..      |
|     S +*+       |
|      +=+E       |
|      oo=.+      |
|       o +o.     |
|         ++o.    |
+-----------------+
[unixyul@montreal ~]$ ll .ssh
total 12
-rw-------. 1 unixyul unixyul 1679 Feb  5 00:17 id_rsa
-rw-r--r--. 1 unixyul unixyul  410 Feb  5 00:17 id_rsa.pub
-rw-r--r--. 1 unixyul unixyul  177 Feb  4 23:58 known_hosts
[unixyul@montreal ~]$
```

OpenSSH – key based authentication

HUMBER
Faculty of Applied Sciences & Technology

2. After generating the RSA key,
   - need to copy public key (id_rsa.pub) to the remote server using **ssh-copy-id** command, this will copy the public key as authorized_keys in user's .ssh folder (server side)
   - accept the fingerprints for remote server
   - enter password for the user of the remote system

```
[unixyul@montreal ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub unixyyz@yyz
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new ke
unixyyz@yyz's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'unixyyz@yyz'"
and check to make sure that only the key(s) you wanted were added.

[unixyul@montreal ~]$ ssh unixyyz@yyz
Last login: Sun Feb  5 00:35:47 2017 from 192.168.3.3
[unixyyz@toronto ~]$
```

   - Now can securely login to Toronto from Montreal using RSA key authentication
   - Compare the ~/.ssh/authorized_keys on server side with ~/.ssh/id_rsa.pub on client side.

SSH from Windows

- Windows requires OpenSSH package installed to function as Server
- Windows 10 has SSH Client packages installed
- The connection syntax from Windows command prompt is same as compared to Linux commands for ssh, scp and sftp.
- puTTY software can be used from Windows to connect LINUX using SSH
- IP address or HOSTNAME can be used to connect in puTTY
- HOSTNAME should be able to resolve to the SSH IP Address using DNS or local configuration in hosts file.
- PORT 22 and SSH protocol has to be entered in puTTY
- The TCP Wrappers package has been deprecated in RHEL 7 and therefore it will not be available in RHEL 8 or later RHEL releases.
  (https://access.redhat.com/solutions/3906701)