1 SSH CheatSheet LINUX

- PDF Link: cheatsheet-ssh-A4.pdf, Category: tools
- Blog URL: `https://cheatsheet.dennyzhang.com/cheatsheet-ssh-A4`

File me Issues or star this repo.
See more CheatSheets from Denny: #denny-cheatsheets

- Related post: Tmux/Tmate Cheatsheet

1.1 SSH general

Name	Summary
Install ssh server	`apt-get install openssh`
Run ssh command	`ssh -o StrictHostKeyChecking=no -p 2702 root@172.17.0.8 date`
SSH with verbose ouptut	`ssh -vvv -p 2702 root@45.33.87.74 date 2>&1`
ssh without input password	`sshpass -p '<your-passwd>' ssh <username>@<ssh_host>`, `brew install sshp`
SSH passwordless login	`ssh-copy-id <username>@<ssh_host>`, Or manually update `~/.ssh/authorized_k`
Remove an entry from `known_hosts file`	`ssh-keygen -f ~/.ssh/known_hosts -R github.com`
Diff local file with remote one	`diff local_file.txt <(ssh <username>@<ssh_host> 'cat remote_file.txt')`
Diff two remote ssh files	`diff <(ssh user@remote_host 'cat file1.txt') <(ssh user2@remote_host2`
Upload with timestamps/permissions kept	`scp -rp /tmp/abc/ ec2-user@<ssh-host>:/root/`
SSH agent load key	`exec ssh-agent bash && ssh-keygen`, `ssh-add`
Emacs read remote file with tramp	`emacs /ssh:<username>@<ssh_host>:/path/to/file`

1.2 SCP

Name	Summary
Download a remote folder	`scp -r ec2-user@<ssh-host>:/home/letsencrypt-20180825 ./`
Upload a file	`scp -i <ssh-keyfile> /tmp/hosts ec2-user@<ssh-host>:/root/`
Upload a folder	`scp -r /tmp/abc/ ec2-user@<ssh-host>:/root/`
Upload with timestamps/permissions kept	`scp -rp /tmp/abc/ ec2-user@<ssh-host>:/root/`
Mount remote directory as local folder	`sshfs name@server:/path/remote_folder /path/local_folder`

1.3 SSH security

Name	Summary
Disable ssh by password	`sed -i 's/PasswordAuthentication yes/PasswordAuthentication no/g' /et`
Disable root login	`sed -i 's/^PermitRootLogin yes/#PermitRootLogin yes/' /etc/ssh/sshd_c`
Enable/Disable SSH Host Key Checking	`StrictHostKeyChecking yes` change `~/.ssh/config`
Protect SSH server from brute force attacks	fail2ban command line tool

1.4 SSH tunnel

Name	Summary
SSH port forward to a local port	`ssh -N -i <ssh-keyfile> -f root@54.179.178.214 -L *:18085:localhost:8085 -`
Reverse port forward to remote server	`ssh -R *:40099:localhost:22 root@54.179.178.214`, `ssh -p 40099 root@54.179.1`
Export local env to Internet	ngrok.com

1.5 SSH files

Name	Summary
Generate a new key pair	`ssh-keygen`, `ssh-keygen -C "your_email@example.com" -t rsa`
Generate key pair without interaction	`ssh-keygen -t rsa -f /tmp/sshkey -N "" -q`
Add passphrase protection to ssh keyfile	`ssh-keygen -p -f id_rsa` link: Manage SSH Key File With Passphrase
Convert OpenSSL format to SSH-RSA format	`ssh-keygen -f my_ssh.pub -i`
Critical ssh files/folders	`~/.ssh/authorized_keys`, `~/.ssh/config`, `~/.ssh/known_hosts`
SSH config file	`/etc/ssh/ssh_config`, `/etc/ssh/sshd_config`
SSH key file permission	`chmod 600 ~/.ssh/id_rsa`
SSH folder permission	`chmod 700 ~/.ssh`, `chown -R $USER:$USER ~/.ssh`
$\text{Authorized}_{\text{keys}}$ file permission	`chmod 644 ~/.ssh/authorized_keys`

1.6 Parse ssh log file

Name	Command
Events of ssh down	`grep -R "ssh.*Received signal 15" /var/log/auth.log`
Events of ssh up	`grep -R "sshd.*Server listening" /var/log/auth.log`
Events of ssh failed login	`grep -R "sshd.*Failed password for invalid user" /var/log/auth.log`
Events of ssh break-in attemp	`grep -R "sshd.*POSSIBLE BREAK-IN ATTEMPT!" /var/log/auth.log`
Events of ssh port scap	`grep -R "sshd.*Bad protocol version identification" /var/log/auth.log`
Events of ssh login by public key	`grep -R "sshd.*Accepted publickey for" /var/log/auth.log`
Events of ssh login by password	`grep -R "sshd.*Accepted password for" /var/log/auth.log`
Events of ssh logout event	`grep -R "sshd.*pam_unix(sshd:session):  session closed for" /var/log/auth.log`

1.7 Scripts

- Inject local key to remote ssh server server

```
cat ~/.ssh/id_rsa.pub | ssh $username@$ssh_hostk "cat - >> ~/.ssh/authorized_keys"

ssh $username@$ssh_hostk "cat ~/.ssh/authorized_keys"
```

- SSH Config file

```
Host sandbox
     HostName 192.168.50.10
     StrictHostKeyChecking no
     User root

Host 192.168.1.*
   StrictHostKeyChecking no
   Port 32882
   UserKnownHostsFile=/dev/null
   IdentityFile ~/.ssh/id_rsa
```

- Use expect to run ssh command with credential auto input

```
#!/usr/bin/expect
set timeout 20
set command "cat /etc/hosts"
set user "vagrant"
set password "vagrant"
set ip "192.168.50.10"
spawn ssh -o stricthostkeychecking=no $user@$ip "$command"
expect "*password:*"
send "$password\r"
expect eof;
```

- ssh reverse tunnel

```
# https://www.howtoforge.com/reverse-ssh-tunneling

autossh -M 40000 -p 2702 -i /home/denny/al -fN \
    -o "PubkeyAuthentication=yes" \
    -o "StrictHostKeyChecking=false" -o "PasswordAuthentication=no" \
    -o "ServerAliveInterval 60" -o "ServerAliveCountMax 3" \
    -R 123.57.240.189:29995:localhost:22 root@123.57.240.189
```

1.8 More Resources

https://neverendingsecurity.wordpress.com/2015/04/07/ssh-cheatsheet/
http://patrickward.com/cheatsheets/2015/02/16/ssh-cheatsheet/
https://bitrot.sh/cheatsheet/13-12-2017-ssh-cheatsheet/
https://gist.github.com/CodyKochmann/166833b3b31cdb936d69
http://pentestmonkey.net/cheat-sheet/ssh-cheat-sheet
https://www.thegeekstuff.com/2008/11/3-steps-to-perform-ssh-login-without-password-using-ssh-keygen-ssh-c