

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	<code>apt-get install openssh</code>
Run ssh command	<code>ssh -o StrictHostKeyChecking=no -p 2702 root@172.17.0.8 date</code>
SSH with verbose output	<code>ssh -vvv -p 2702 root@45.33.87.74 date 2>&1</code>
ssh without input password	<code>sshpass -p '<your-passwd>' ssh <username>@<ssh_host></code> , brew install sshpass
SSH passwordless login	<code>ssh-copy-id <username>@<ssh_host></code> , Or manually update <code>~/.ssh/authorized_keys</code>
Remove an entry from <code>known_hosts</code> file	<code>ssh-keygen -f ~/.ssh/known_hosts -R github.com</code>
Diff local file with remote one	<code>diff local_file.txt <(ssh <username>@<ssh_host> 'cat remote_file.txt')</code>
Diff two remote ssh files	<code>diff <(ssh user@remote_host 'cat file1.txt') <(ssh user2@remote_host2 'cat file2.txt')</code>
Upload with timestamps/permissions kept	<code>scp -rp /tmp/abc/ ec2-user@<ssh-host>:/root/</code>
SSH agent load key	<code>exec ssh-agent bash && ssh-keygen, ssh-add</code>
Emacs read remote file with tramp	<code>emacs /ssh:<username>@<ssh_host>:/path/to/file</code>

1.2 SCP

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

1.3 SSH security

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

1.4 SSH tunnel

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

1.5 SSH files

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

1.6 Parse ssh log file

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

1.7 Scripts

- Inject local key to remote ssh 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

License: Code is licensed under MIT License.

<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>