### 无需私钥的pssh的安装使用

#### 关于PSSH

pssh是python写的可以并发在多台机器上批量执行命令的工具，它支持文件并行复制，远程命令执行，杀掉远程主机上的进程等等。杀手锏是文件并行复制，，当进行再远程主机批量上传下载的时候，最好使用它。

在使用pssh之前，必须要保证管理主机和本地主机进行过密钥的认证，或者是在进行批量时，没有做过密钥认证，但是必须保证被管理的多台主机的密码相同。关于如何做密钥认证，这里就不多说了，可以自行百度。

INSECURE\_PSSH下可用的工具如下：

* pssh：在远程多台主机上并行运行命令
* pscp :把文件并行复制到多台远程主机上
* prsync:使用rsync协议本地文件同步到远程多台主机上。
* pnuke:在远程多台主机上并行killall某一进程
* pslurp:把文件从远程多台主机上复制到本地主机上

ps:建议分发文件，执行命令，批量杀死进程，使用pssh,pscp，pnuke,速度很快的

#### Step1下载insecure\_pssh

1. $ wget https://github.com/knktc/insecure\_pssh/archive/master.zip
2. #或者IE打开https://github.com/knktc/insecure\_pssh 然后点击zip即可下载

#### Step2 解压安装软件

1. $ su - root
2. # cp master.zip /usr/local/
3. # cd /usr/local/
4. # unzip master.zip
5. Archive:  master.zip
6. 954e8d61a07f8f30ee0eba34ae82852dbb6e412b
7. creating: insecure\_pssh-master/
8. inflating: insecure\_pssh-master/COPYING
9. inflating: insecure\_pssh-master/README
10. inflating: insecure\_pssh-master/hosts.txt
11. inflating: insecure\_pssh-master/pscp
12. inflating: insecure\_pssh-master/pslurp
13. inflating: insecure\_pssh-master/pssh
14. creating: insecure\_pssh-master/psshlib/
15. extracting: insecure\_pssh-master/psshlib/\_\_init\_\_.py
16. inflating: insecure\_pssh-master/psshlib/cli.py
17. inflating: insecure\_pssh-master/psshlib/color.py
18. inflating: insecure\_pssh-master/psshlib/manager.py
19. inflating: insecure\_pssh-master/psshlib/psshutil.py
20. inflating: insecure\_pssh-master/psshlib/task.py
21. extracting: insecure\_pssh-master/psshlib/version.py
22. # cd insecure\_pssh-master
23. # chmod +x pscp pslurp pssh

#### Step3 配置host.txt

host.txt格式如下：

1. 用户名:密码@主机
2. bigdata:bigdatause@192.168.1.112
4. 主机 用户名 密码
5. 192.168.1.112 bigdata bigdatause

注意：一般我们将该软件拷贝到 Master 节点上，修改 Host.txt 文件时候需要注意，只需要将 Slaver 节点的网络地址写进去。千万不要将 Master 节点的网络地址也写进去。 因为该批处理命令不支持对本机的操作，配置进去运行会报错 code 01 的错误。

#### Step4 使用参数说明

帮助命令：

1. $ pssh --help
2. Usage: pssh [OPTIONS] command [...]
4. Options:
5. --version             show program's version number and exit
6. --help                show **this** help message and exit
7. -h HOST\_FILE, --hosts=HOST\_FILE
8. hosts file (each line "[user[:password]@]host[:port]"
9. or "host[:port] user password")
10. -H HOST\_STRING, --host=HOST\_STRING
11. additional host entries
12. ("[user[:password]@]host[:port]")
13. -l USER, --user=USER  username (OPTIONAL)
14. -p PAR, --par=PAR     max number of parallel threads (OPTIONAL)
15. -o OUTDIR, --outdir=OUTDIR
16. output directory **for** stdout files (OPTIONAL)
17. -e ERRDIR, --errdir=ERRDIR
18. output directory **for** stderr files (OPTIONAL)
19. -t TIMEOUT, --timeout=TIMEOUT
20. timeout (secs) (0 = no timeout) per host (OPTIONAL)
21. -O OPTION, --option=OPTION
22. SSH option (OPTIONAL)
23. -v, --verbose         turn on warning and diagnostic messages (OPTIONAL)
24. -x ARGS, --extra-args=ARGS
25. Extra command-line arguments, with processing **for**
26. spaces, quotes, and backslashes
27. -X ARG, --extra-arg=ARG
28. Extra command-line argument
29. -i, --inline          inline aggregated output and error **for** each server
30. --inline-stdout       inline standard output **for** each server
31. -I, --send-input      read from standard input and send as input to ssh
32. -P, --print           print output as we get it
34. Example: pssh -h hosts.txt -l irb2 -o /tmp/foo uptime

pssh参数说明如下：

* -h   HOST\_FILE   后边跟远程主机列表（ip）
* -H   HOST\_STRING   后边跟远程主机名或者ip地址
* -l   USER  指定远程主机的用户名
* -p  PAR   指定pssh最大的并行线程数。
* -o  将输出的内容重定向到一个指定的文件中
* -O 指定ssh参数的具体配置
* -e  将执行错误重定向到一个指定的文件中
* -t  设定命令执行超时时间
* -x  传递ssh命令的一些参数
* -i  在远程主机上执行命令完成后显示标准输出和标准错误
* -P  在执行远程命令时，输出执行结果

#### PSSH实例

在远程多台主机上运行命令“ls -lrt”：

1. $ pssh -P -i -h $INSECURE\_HOME/hosts.txt "ls -lrt"

说明：“$INSECURE\_HOME/hosts.txt”这个文件可以自由指定。通常情况下使用alias进行配置命令别名。

1. $ vi ~/.bashrc
2. # User specific aliases and functions
3. alias pssh='pssh -P -i'

#### PSCP实例

pscp拷贝文件到远程主机

1. $ pscp -h iplist.txt ~/iplist.txt ~/

#### PSLURP实例

pslurp 远程主机拷贝文件到本地主机，将远程文件copy 到本地，-L指定本地目录，后面接名称表示将文件重命名为ip.txt

1. $ pslurp -v  -h iplist.txt -L /home/bigdata/ /home/bigdata/iplist.txt ip.txt
2. [1] 10:40:17 [SUCCESS] bigdata@192.168.1.112
4. $ cat /home/bigdata/192.168.1.112/ip.txt
5. 192.168.1.112 bigdata bigdatause

pslurp 远程主机拷贝目录到本地主机，使用-r表示递归复制

1. $ pssh -h iplist.txt 'mkdir -p ~/test;cd ~/test;touch a.txt;touch b.txt'
2. [1] 10:44:32 [SUCCESS] bigdata@192.168.1.112
4. $ pslurp -h iplist.txt -r -L /home/bigdata/ /home/bigdata/test open
5. [1] 10:45:29 [SUCCESS] bigdata@192.168.1.112
7. $ ls -lrt -R ~/192.168.1.112/
8. /home/bigdata/192.168.1.112/:
9. total 4
10. -rw-rw-r--. 1 bigdata bigdata 33 Jun 16 10:40 ip.txt
11. drwxrwxr-x. 2 bigdata bigdata 32 Jun 16 10:45 open
13. /home/bigdata/192.168.1.112/open:
14. total 0
15. -rw-rw-r--. 1 bigdata bigdata 0 Jun 16 10:45 a.txt
16. -rw-rw-r--. 1 bigdata bigdata 0 Jun 16 10:45 b.txt

#### pnuke实例

pnuke杀掉某一进程，这个命令类似yu  killall命令，能通过killall关闭的服务，都可以通过pnuke来批量完成。

1. pnuke  -h  iplist.txt   httpd

上边的意思是在远程主机上批量关闭httpd服务。