Bash Scripting

An overview of bash scripting, and cron jobs using iptables

DISCLAIMER: Delete these rules when done as they will in all likelihood interrupt your normal network traffic

First off, lets make sure we have a few things, if you don't already

- *sudo apt-get install iptables-persistent*

Let's begin by making a bash script to add the rules!

- nano rules.sh
    - To begin creating your file
- Script contents...

```
#!/bin/bash

iptables -F

#Loopback
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED -j ACCEPT

#FTP
iptables -A INPUT -p tcp --dport 21 -m conntrack --ctstate NEW,ESTABLISHED -j
ACCEPT
iptables -A OUTPUT -p tcp --sport 21 -m conntrack --ctstate ESTABLISHED -j
ACCEPT

#SSH
iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j
ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j
ACCEPT

#HTTP & HTTPS
```

```
iptables -A INPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate
NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate
ESTABLISHED -j ACCEPT

#DNS Outbound
iptables -A OUTPUT -p udp -o ens33 --dport 53 -j ACCEPT
iptables -A INPUT -p udp -i ens33 --sport 53 -j ACCEPT

#PING--ICMP
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

#Logging
iptables -N LOGGING
iptables -A INPUT -j LOGGING
iptables -A OUTPUT -j LOGGING
iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables-
Dropped: " --log-level 4
iptables -A LOGGING -j DROP

iptables-save > /etc/iptables/rules.v4
iptables -L
```

- chmod 755 rules.sh
    - To make it executable
- ./rules.sh
    - Run this as **root**, *mileage may vary with sudo*
    - To run your bash script
- To confirm they have been added to the right spot, *cat /etc/iptables/rules.v4*

Now let's make a bash script to get the logs from kern.log

- *nano getlog.sh*
- Script contents...

```
#!/bin/bash

# Incriment file name
name=iptables
if [[ -e $name.log ]] ; then
    i=0
    while [[ -e $name-$i.log ]] ; do
```

```
        let i++
    done
    name=$name-$i
fi

# Here we only get the last 50 lines from kern.log whether or not we find the
desired files.
log=$(tail -n 50 /var/log/kern.log | grep --ignore-case 'iptables')

# Here we take the results of the command and output it into the home
directory of our user
echo  "$log" > ~/$name.log
```

- *chmod 755 getlog.sh*
- ./getlog.sh
- cat ~/iptableslog.txt
    - If you're getting all logged dropped sent to this file you can do
        - *cat ~/iptables.sh* | *grep **whatever***

It's Cron Time!

- *crontab -e*
- Replace **$uname** with whatever your user name is.
- At the bottom of your crontab file add this line...

```
*/2 * * * * /home/$uname/getlog.sh
```

- This means that the script will be run every 2 minutes and the log will be offloaded

Q&A? [interactive]

-

Homework

- Send the output of *getlog.sh* to its own directory within the home folder as to keep things from getting cluttered.
- Purposefully try to establish communications on a blocked port and then search the files using grep or another tool for that port.
    - Hint: For this you may need to modify the logging speed of the iptables rules, or the retrieval speed in crontab
- Modify the bash script to get the last 50 iptables logs instead so you guarantee contents in your log