How to protect from port scanning and smurf attack in Linux Server by iptables

Written by BiRU
Sunday, 30 April 2017 09:08 -

In this post I will share the iptable script in which we will learn **How to protect from port scanning and smurf attack in Linux Server**
.

Features Of Script :

(1) When a attacker try to port scan your server, first because of iptable attacker will not get any information which port is open. Second the Attacking IP address will be blacklisted for 24 Hour (You can change it in script) . Third , after that attacker will not able to open access anything for eg. even attacker will not see any website running on server via web browser, not able to ssh,telnet also. Means completely restricted.

(2) Protects from smurf attack

(3) Written with the help of IPTABLE hence no System Performance issue like CPU high,Memory usage etc. No third party tool is used

Note: You can add or remove port no. as per your requirement.

Description about Server where we will implement IPTABLE script:

Operating Syetem : CentOS 6.4 (applicable to Red hat and CentOS servers)
 IP Address: 192.168.1.4

Now we will create the script

How to protect from port scanning and smurf attack in Linux Server by iptables

Written by BiRU
Sunday, 30 April 2017 09:08 -

Step 1: Create a bash script with the name of **iptablescript.sh**

  vi /root/iptablescript.sh

Step 2: Now paste the below given script contents in your bash script file **iptablescript.sh**

```
  #!/bin/sh  #  #  # Script is for stoping Portscan and smurf attack    ### first flush all the
iptables Rules                                                        iptables -F
                                                                          #
INPUT iptables Rules

# Accept loopback input
 iptables
-
A INPUT
-
i lo
-
p all
-
j ACCEPT
# allow 3 way handshake
 iptables
-
A INPUT
-
m state
--
state ESTABLISHED
,
RELATED
-
j ACCEPT
### DROPspoofing packets
 iptables
-
A INPUT
-
s
10.0
.
0.0
/
8
-
j DROP   iptables
```

How to protect from port scanning and smurf attack in Linux Server by iptables

Written by BiRU

Sunday, 30 April 2017 09:08 -

-

A INPUT

-

s

169.254

.

0.0

/

16

-

j DROP iptables

-

A INPUT

-

s

172.16

.

0.0

/

12

-

j DROP iptables

-

A INPUT

-

s

127.0

.

0.0

/

8

-

j DROP iptables

-

A INPUT

-

s

192.168

.

0.0

/

24

-

j DROP   iptables

-

A INPUT

Written by BiRU
Sunday, 30 April 2017 09:08 -

-

s

224.0

.

0.0

/

4

-

j DROP iptables

-

A INPUT

-

d

224.0

.

0.0

/

4

-

j DROP iptables

-

A INPUT

-

s

240.0

.

0.0

/

5

-

j DROP iptables

-

A INPUT

-

d

240.0

.

0.0

/

5

-

j DROP iptables

-

A INPUT

-

s

How to protect from port scanning and smurf attack in Linux Server by iptables

Written by BiRU
Sunday, 30 April 2017 09:08 -

0.0

.

0.0

/

8

-

j DROP iptables

-

A INPUT

-

d

0.0

.

0.0

/

8

-

j DROP iptables

-

A INPUT

-

d

239.255

.

255.0

/

24

-

j DROP iptables

-

A INPUT

-

d

255.255

.

255.255

-

j DROP

#for SMURF attack protection

 iptables

-

A INPUT

-

p icmp

-

m icmp

Written by BiRU

Sunday, 30 April 2017 09:08 -

--

icmp

-

type address

-

mask

-

request

-

j DROP iptables

-

A INPUT

-

p icmp

-

m icmp

--

icmp

-

type timestamp

-

request

-

j DROP iptables

-

A INPUT

-

p icmp

-

m icmp

-

m limit

--

limit

1

/

second

-

j ACCEPT

Droping all invalid packets

iptables

-

A INPUT

-

m state

--

Written by BiRU
Sunday, 30 April 2017 09:08 -

state INVALID

-

j DROP iptables

-

A FORWARD

-

m state

--

state INVALID

-

j DROP iptables

-

A OUTPUT

-

m state

--

state INVALID

-

j DROP
flooding of RST packets, smurf attack Rejection
 iptables

-

A INPUT

-

p tcp

-

m tcp

--

tcp

-

flags RST RST

-

m limit

--

limit
2
/
second

--

limit

-

burst
2

-

j ACCEPT
Protecting portscans

Written by BiRU
Sunday, 30 April 2017 09:08 -

Attacking IP will be locked for 24 hours (3600 x 24 = 86400 Seconds)
 iptables
-
A INPUT
-
m recent
--
name portscan
--
rcheck
--
seconds
86400
-
j DROP iptables
-
A FORWARD
-
m recent
--
name portscan
--
rcheck
--
seconds
86400
-
j DROP
Remove attacking IP after 24 hours
 iptables
-
A INPUT
-
m recent
--
name portscan
--
remove iptables
-
A FORWARD
-
m recent
--
name portscan
--

remove

These rules add scanners to the portscan list, and log the attempt.

iptables

-

A INPUT

-

p tcp

-

m tcp

--

dport

139

-

m recent

--

name portscan

--

set

-

j LOG

--

log

-

prefix

"portscan:"

iptables

-

A INPUT

-

p tcp

-

m tcp

--

dport

139

-

m recent

--

name portscan

--

set

-

j DROP   iptables

-

A FORWARD

-

How to protect from port scanning and smurf attack in Linux Server by iptables

Written by BiRU
Sunday, 30 April 2017 09:08 -

```
p tcp
-
m tcp
--
dport
139
-
m recent
--
name portscan
--
set
-
j LOG
--
log
-
prefix
"portscan:"
 iptables
-
A FORWARD
-
p tcp
-
m tcp
--
dport
139
-
m recent
--
name portscan
--
set
-
j DROP
# Allow the following ports through from outside
 iptables
-
A INPUT
-
p tcp
-
m tcp
--
```

dport
25
-
j ACCEPT iptables
-
A INPUT
-
p tcp
-
m tcp
--
dport
80
-
j ACCEPT iptables
-
A INPUT
-
p tcp
-
m tcp
--
dport
443
-
j ACCEPT iptables
-
A INPUT
-
p tcp
-
m tcp
--
dport
22
-
j ACCEPT
Allow ping means ICMP port is open (If you do not want ping replace ACCEPT with REJECT)
 iptables
-
A INPUT
-
p icmp
-
m icmp
--

icmp

-

type
8

-

j ACCEPT
Lastly reject All INPUT traffic
 iptables

-

A INPUT

-

j REJECT
################ Below are for OUTPUT iptables rules
###

Allow loopback OUTPUT
 iptables

-

A OUTPUT

-

o lo

-

j ACCEPT iptables

-

A OUTPUT

-

m state

--

state ESTABLISHED

,

RELATED

-

j ACCEPT
Allow the following ports through from outside

SMTP = 25

DNS =53

HTTP = 80

HTTPS = 443

SSH = 22

You can also add or remove port no. as per your requirement

How to protect from port scanning and smurf attack in Linux Server by iptables

Written by BiRU
Sunday, 30 April 2017 09:08 -

iptables

-

A OUTPUT

-

p tcp

-

m tcp

--

dport
25

-

j ACCEPT iptables

-

A OUTPUT

-

p udp

-

m udp

--

dport
53

-

j ACCEPT iptables

-

A OUTPUT

-

p tcp

-

m tcp

--

dport
80

-

j ACCEPT iptables

-

A OUTPUT

-

p tcp

-

m tcp

--

dport
443

-

j ACCEPT iptables

-

A OUTPUT

-

p tcp

-

m tcp

--

dport
22

-

j ACCEPT
Allow pings
 iptables

-

A OUTPUT

-

p icmp

-

m icmp

--

icmp

-

type
8

-

j ACCEPT
Lastly Reject all Output traffic
 iptables

-

A OUTPUT

-

j REJECT
Reject Forwarding traffic
 iptables

-

A FORWARD

-

j REJECT

Step 3: Make the Read Write Execute permission only to root user. (For security)

   chmod 700 /root/iptablescript.sh   chown root:root /root/iptablescript.sh

Step 4 : Now run the script

  sh /root/iptablescript.sh   or   ./root/iptablescript.sh

Step 6: Now check the IPTABLES rule with following command

iptables -nL   **Now we will do testing from remote server to our server where we have implemented the iptable**

Step 7: login into any system and try to do port scanning

nmap -sT Server-ip-address   eg.   nmap -sT 192.168.1.4

Step 8: The result should be now from your system like following types

(a) Not getting any output from nmap
 (b) Not able to do telnet to any port for eg. telnet Server-ip-address 22

After running nmap means port scan your ip-address is blacklisted.

You can find your system ip address in message logs in Server with the keyword called **portscan**   .
 So login back to your server and check the messages log in /var/log

Note : how to install nmap

 In Red Hat and CentOS   yum install nmap   In Debian and Ubuntu   apt-get install nmap