Table Of Contents

nixCraft: Linux Tips, Hacks, Tutorials, And Ideas In Blog Format
http://www.cyberciti.biz/

BSD FTP-Proxy: PF Firewall Allow Outgoing Active / Passive FTP Connections

Posted by Vivek Gite <vivek@nixcraft.com>

[1]

Q. I've FreeBSD based Apache webserver. I need to allow outgoing ftp client requests so that BSD ports collection can download from various ftp sites. How do I allow outgoing FTP connection via PF network firewall software under FreeBSD or OpenBSD operating system?

A. You need to use ftp-proxy, which is a proxy for the Internet File Transfer Protocol. ftp-proxy is installed by default along with PF firewall.

Step # 1: Turn on ftp-proxy under FreeBSD

Open /etc/rc.conf file under FreeBSD

```
# vi /etc/rc.conf
```

Append following line:

```
ftpproxy_enable="YES"
```

If you are using OpenBSD, type the following command to start the ftp proxy on boot:

```
echo 'ftpproxy_flags=""' >>/etc/rc.conf.local
```

By default ftp proxy listen on 8021 port bind to 127.0.0.1 IP address.

Step # 2: Configure pf and ftp-proxy

Open your /etc/pf.conf file and add following into your NAT section:
To activate it, put something like this in the NAT section of pf.conf:

```
nat-anchor "ftp-proxy/*"
rdr-anchor "ftp-proxy/*"
rdr pass proto tcp from any to any port ftp -> 127.0.0.1 port 8021
```

All three rules required, even if your setup does not use NAT. Find your filtering rule and append the following rules:

```
anchor "ftp-proxy/*"
```

Save and close the file.

Sample pf.conf rules

Here is my own working sample /etc/pf.conf file that allows outgoing ftp, along with ssh, http, dns service. It only allows incomming traffic on port 53, 80:

```
#### First declare a couple of variables ####
# outgoing services
tcp_services = "{ ssh, smtp, domain, www, https, ntp, 43}"
udp_services = "{ domain, ntp }"
icmp_types = "{ echoreq, unreach }"

martians = "{ 127.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, 169.254.0.0/16, 192.0

ext_if = "em1" # Internet
int_if = "em0" # vpn / lan

proxy="127.0.0.1" # ftp proxy IP
```

```
proxyport="8021" # ftp proxy port

#### Normalization
scrub in all

#### NAT and RDR start
nat-anchor "ftp-proxy/*"
rdr-anchor "ftp-proxy/*"

# Redirect ftp traffic to proxy
rdr pass proto tcp from any to any port ftp -> $proxy port $proxyport

#### Start filtering
# Drop incoming everything
block in all

# Default connection refused message to client
block return

# keep stats of outging connections
pass out keep state

# We need to have an anchor for ftp-proxy
anchor "ftp-proxy/*"

# Unlimited traffic for lo0 and VPN/Lan interface
set skip on {lo0, $int_if}

# activate spoofing protection for all interfaces
block in quick from urpf-failed

# Antispoof is a common special case of filtering and blocking. This mechanism protects aga
antispoof log for $ext_if

#Block RFC 1918 addresses
block drop in log (all)  quick on $ext_if from $martians to any
block drop out log (all) quick on $ext_if from any to $martians

# Allow outgoing via ssh, smtp, domain, www, https, whois etc
pass out on $ext_if proto tcp to any port $tcp_services
pass out on $ext_if proto udp to any port $udp_services

# Allow outgoing Trace route
pass out on $ext_if inet proto udp from any to any port 33433 >< 33626 keep state

# Allow incomming named udp / tcp 53
pass in on $ext_if proto udp from any to any port 53
# All tcp service protected using synproxy
pass in on $ext_if proto tcp from any to any port 53 flags S/SA synproxy state
# Allow http traffic
pass in on $ext_if proto tcp from any to any port 80 flags S/SA synproxy modulate state
# SSH
pass in on $ext_if proto tcp from any to any port 22 flags S/SA synproxy modulate state
# Allow ICMP ping
pass inet proto icmp all icmp-type $icmp_types keep state
```

Step # 3: Restart PF firewall

Type the following command under FreeBSD:

```
# /etc/rc.d/pf restart
```

OR type the following under OpenBSD (also works under FreeBSD):

```
# pfctl -nf /etc/pf.conf
# pfctl -f /etc/pf.conf
```

Step # 4: Start ftp-proxy

Type the following command to start ftp-proxy under, FreeBSD:

```
# /etc/rc.d/ftp-proxy start
```

Under OpenBSD, you can simply type the following to start ftp-proxy:

```
# /usr/sbin/ftp-proxy
```

Test your setup

Use ftp client to test your test, enter:

```
$ ftp ftp.freebsd.org
```

Further readings:

- man pages ftp-proxy, rc.conf, pf, pf.conf
- pf documentation [2]

URLs in this post:

[1] Image: **http://www.cyberciti.biz/faq/faq/category/freebsd/**
[2] pf documentation: **http://openbsd.org/faq/pf/index.html**