How To Set Up a Firewall with Awall on Alpine Linux

Author: Vivek Gite Last updated: November 30, 2020 0 comments

Alpine Wall (Awall) is an easy to use interface to iptables for Alpine Linux users. While iptables and ip6tables is an excellent command, it can be challenging for new Alpine Linux users. The awall tool has easy to follow high-level concepts such as zones, limits, policies, and a single source for both IPv4 and IPv6. This tutorial will show you how to set up a firewall with Awall on Alpine Linux.

Tutorial requirements	
Operating system/app	Alpine Linux
Root privileges required	Yes
Difficulty	Advanced (rss)
Estimated completion time	20m

Table of contents

How to install Awall on Alpine Linux

You need to update system using the apk command:

```
# apk update && apk upgrade
## Install both IPv4 and IPv6 version of IPtables ##
# apk add ip6tables iptables
## Install awall ##
# apk add -u awall
## Verify it ##
```

```
# apk version awall
```

Understanding Awall concepts

Awall configuration starts with a single or multiple JSON-formatted files named policy file. The mandatory policies shipped with Alpine Linux are located at /usr/share/awall/mandatory and viewed with the ls command/cat command:

```
# ls -l /usr/share/awall/mandatory/
# more /usr/share/awall/mandatory/services.json
# cat /usr/share/awall/mandatory/defaults.json
```

However, developers and sysadmins need to install server specific custom policies in /etc/awall directory:

```
# ls /etc/awall/
```

Each awall policy can contain definitions for:

1. **Variables** (like $interface_name)
2. **zones** (like internet, lan, red_zone, green_zone, blue_zone) – ones are defined based on a interface and assigned a name to be used in your policies.
3. **interfaces** (like eth0)
4. **policies** – The policy is what tell Awall what to do with when a packet enters or leaves from one of the zones (interfaces).
5. **filters** and NAT rules
6. **services** (like nginx or mysql)

   snat – Apply source nat for outgoing packets. Translate from local ip to public ip. In other words your Alpine box act as a router.

Let us see how to configure Awall for a dedicated VM or bare metal server where eth0 is connected to the high-speed Internet, and we need to protect the server. Our goal is to allow ssh (22), ping, and HTTP (80) + HTTPS (4430 ports only.

Step 1. Prerequisites

First we must load Linux kernel drivers (modules) for firewall using the modprobe command:

```
# modprobe -v ip_tables ip6_tables # IPv4
# modprobe -v ip6_tables # if IPv6 is used
# modprobe -v iptable_nat # if NAT is used aka router
```

```
insmod /lib/modules/5.4.43-1-virt/kernel/net/netfilter/x_tables.ko
insmod /lib/modules/5.4.43-1-virt/kernel/net/ipv4/netfilter/ip_tables.ko
ip6_tables
```

Please note that the above commands needed only the first time, after awall installation. Next, make the firewall autostart at boot time and autoload the required Linux kernel modules:

```
# rc-update add iptables
* service ip6tables added to runlevel default
# rc-update add ip6tables
* service ip6tables added to runlevel default
```

Please note that Awall is a frontend tool that creates rules. All firewall rules stored in /etc/iptables/ directory. Firewall service can be stopped or restarted any time using the following commands:

```
# ls -l /etc/iptables/
# cat /etc/iptables/rules-save
# rc-service iptables {start|stop|restart|status}
# rc-service ip6tables {start|stop|restart|status}
```

Alpine Linux command to control iptables firewall

Step 2. Set up a firewall with Awall to protect Alpine Linux box

Create a new file called cloud-server.json as follows to drop all incoming, and outgoing traffic using a text editor. Here is my sample file to protect cloud server hosted at Linode:

```
# cat /etc/awall/optional/cloud-server.json
```

```
{
  "description": "Default awall policy to protect Cloud server",

  "variable": { "internet_if": "eth0" },

  "zone": {
    "internet": { "iface": "$internet_if" }
  },

  "policy": [
    { "in": "internet", "action": "drop" },
    { "action": "reject" }
  ]

}
```

Next open TCP port 22. In other words, allow incoming SSH access with the maximum ssh login rate limit to avoid attackers/bots brute-forcing into our Alpine cloud server:

```
# cat /etc/awall/optional/ssh.json
```

```
{

    "description": "Allow incoming SSH access (TCP/22)",

    "filter": [
        {
            "in": "internet",
            "out": "_fw",
            "service": "ssh",
            "action": "accept",
            "conn-limit": { "count": 3, "interval": 60 }
        }
    ]
}
```

Where,

- **`"in": "internet",`** : The internet is our default zone and we are creating incoming policy to open TCP port 22.
- **`"out": "_fw",`** : AWall has a built-in zone named “_fw” which is the “firewall itself”.

- **`"service": "ssh",`** : Open SSH port 22.
- **`"action": "accept",`** : Accept the packet.
- **`"conn-limit": { "count": 3, "interval": 60 }`** : Rate limit ssh connections.

A note about restricting ssh traffic from specific IP address or sub/net (CIDR)

Update your ssh.json as follows with `"src": "1.2.3.4",` option:

```
{
    "description": "Allow incoming SSH access (TCP/22) only from our office VPN
at 1.2.3.4",
    "filter": [
        {
            "in": "internet",
            "out": "_fw",
            "service": "ssh",
            "action": "accept",
            "src": "1.2.3.4",
            "conn-limit": { "count": 3, "interval": 60 }
        }
    ]
}
```

For multiple IPs/CIDRs, try it as follows:

```
      "src": [ "1.2.3.4", "192.168.2.0/24", "202.54.5.1" ],
```

Be a good netizen and allow ping/ICMP request to our server with rate limits applied:

```
# cat /etc/awall/optional/ping.json
```

```
{

    "description": "Allow ping-pong",

    "filter": [
        {
              "in": "internet",
              "service": "ping",
              "action": "accept",
              "flow-limit": { "count": 10, "interval": 6 }
        }
    ]
}
```

Finally allow selected outgoing connections for HTTP/HTTPS, DNS, ping, ssh, DNS and NTP protocols:

```
# cat /etc/awall/optional/outgoing.json
```

```
{

    "description": "Allow outgoing connections for dns, http/https, ssh, ntp,
ssh and ping",

    "filter": [
        {
            "in": "_fw",
            "out": "internet",
            "service": [ "dns", "http", "https", "ssh", "ntp", "ping" ],
            "action": "accept"
```

```
        }
    ]
}
```

Step 3. List Awall available policy(s)

Run:

```
# awall list
```

Outputs:

```
cloud-server  disabled  Default awall policy to protect Cloud server
outgoing      disabled  Allow outgoing connections for dns, http/https, ssh,
ntp, ssh and ping
ping          disabled  Allow ping-pong
ssh           disabled  Allow incoming SSH access (TCP/22)
```

Step 4. Enables the Awall firewall policy

Again run:

```
# awall enable cloud-server
# awall enable ssh
# awall enable ping
# awall enable outgoing
```

Step 5. Start the firewall

Finally we need to create firewall configuration from the policy files and enables it. In other words the following command will starts the firewall and load all rules:

```
# awall activate
```

You will be prompted to hit the RETURN key as follows:

```
New firewall configuration activated
Press RETURN to commit changes permanently:
```

NOTE: After enabling or disabling policy, you always need to run **`awall activate`** command to update the iptables rules.

Step 6. Open incoming HTTP and HTTPS ports

Create a new policy file called apache.json as follows:

```
# vi /etc/awall/optional/apache.json
```

```
{
    "description": "Allow incoming Apache HTTP/HTTPS (TCP/80 and 443) ports",
    "filter": [
        {
            "in": "internet",
            "out": "_fw",
            "service": [ "http", "https"],
            "action": "accept"
        }
    ]
}
```

Save and close the file. Next activate new firewall policy rule:

```
# awall enable apache
# awall activate
```

Step 7. Close port or diable existing policy

Say you have a policy named openvpn.json, and you no longer use the OpenVPN server. We can close the UDP/1194 port as follows by disabling the policy:

```
# awall list
# awall disable openvpn
# awall activate
```

Step 8. View iptables rules

You can list all iptables rules using the following commands:

```
# iptables -L -n -v | more
# ip6tables -L -n -v | more
# ip6tables -S
# iptables -S
```

Want to see dropped packets log? Try the dmesg command and grep command:

```
# dmesg
## See dropped packets from SSH port ##
# dmesg | grep -w DPT=22
```

```
[ 3532.077008] IN=eth0 OUT= MAC=f2:3c:92:64:16:11:aa:01:9d:43:81:e6:08:00
SRC=194.180.224.130 DST=172.105.xx.yy LEN=60 TOS=0x00 PREC=0x20 TTL=49 ID=9647
DF PROTO=TCP SPT=33106 DPT=22 WINDOW=29200 RES=0x00 SYN URGP=0
[ 3532.077347] IN=eth0 OUT= MAC=f2:3c:92:64:16:11:aa:01:9d:43:81:e6:08:00
SRC=194.180.224.130 DST=172.105.xx.yy LEN=60 TOS=0x00 PREC=0x20 TTL=50 ID=35762
DF PROTO=TCP SPT=33110 DPT=22 WINDOW=29200 RES=0x00 SYN URGP=0
[ 3533.078293] IN=eth0 OUT= MAC=f2:3c:92:64:16:11:aa:01:9d:43:81:e6:08:00
SRC=194.180.224.130 DST=172.105.xx.yy LEN=60 TOS=0x00 PREC=0x20 TTL=50 ID=35763
DF PROTO=TCP SPT=33110 DPT=22 WINDOW=29200 RES=0x00 SYN URGP=0
```

Step 9. Disabling and resetting Awall

For any reason you no longer want to use firewall and Awall, you can disable it with following commands:

```
# rc-service iptables stop
# rc-service ip6tables stop
```

```
 * Saving ip6tables state ...                          [ ok ]
 * Stopping firewall ...                               [ ok ]
```

Now list and disable all policies:

```
# awall list
# awall disable cloud-server
# awall disable ssh
# awall disable ping
# awall disable outgoing
```

Finally disable iptables firewall service on your Alpine Linux box using the rc-update command:

```
# rc-update del ip6tables
# rc-update del iptables
```

```
 * service iptables removed from runlevel default
```

Conclusion

In this quick tutorial, we learned about Awall and set up a default firewall policy that drops all incoming and outgoing connections on Alpine Linux. Next, we opened the required ports as per our needs. Try awall help as follows:

```
# awall help
```

See the following resources for further information:

- Awall project home page.
- Awall document from the Alpine Linux wiki.

https://www.cyberciti.biz/faq/how-to-set-up-a-firewall-with-awall-on-alpine-linux/