Principal , Fotolia

User-level firewalling with Portsmith

BEST BEHAVIOR

The Linux packet filter iptables lacks a function that dynamically enables ports for authenticated users. Portsmith plugs this gap, allowing users to enable their own connections. **BY CHRISTIAN NEY**

Check Point and Cisco administrators are familiar with firewalls that enable ports after a user logs in. Unfortunately, this technique, sometimes referred to as Client Authentication or Cut-Through Proxy, is often subject to restrictions. Because of the problems associated with authenticating firewalls, iptables does not include this functionality out of the box. Of course, you could add your own custom authentication feature with some scripting, but few admins go to so much effort.

Portsmith [1] offers a free and easy option for authentication at the firewall, and this innovative tool even lets authenticated users enable ports in their own web browsers. To avoid potential security threats, the administrator still keeps control of the permissions. Each user is assigned a set of required communication links and can only access the resources assigned to those links. This approach stops users from simply punching holes in the firewall ruleset anytime they feel the urge.

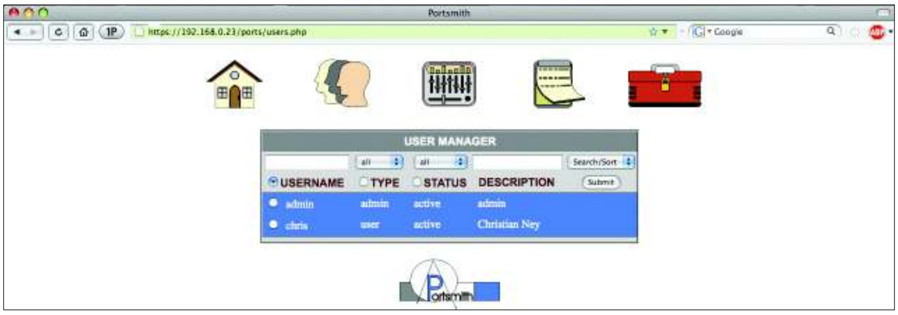
One potential application for a Portsmith firewall is as a replacement for a virtual private network (VPN). The traffic is stopped at the firewall until a user authenticates, and once the user logs in, only traffic from the user's IP address is admitted. By ensuring that a specific service is only available to the authenticated user, the admin can avoid the need for a dedicated VPN server. On the other end, the client also does not need special software to initiate the connection. According to the Portsmith website, "… you can take control of your office computer from home, from a friend's house, from a WiFi hotspot, or from any other location with Internet access. Since there is no software requirement, after you leave the location, there will be no trace that you were ever there, and nothing is installed on the computer you were using." Another of Portsmith's goodies is an integrated, browser-based backup solution. Just point and click to write the ruleset, critical files, and database to a CD.

Portsmith prefers Ubuntu 8.0.4 Server LTS [2]. Of course, you can also use other distributions, although you will need to modify the paths and component handling to match Portsmith's requirements. Portsmith is designed to work with the Apache web server [3], the PHP5 module [4], and the PostgreSQL database module [5]. Tough luck if you prefer a different database system; other systems are not supported.

The web server acts as a front end for the administrator and users. Accounts are stored in the database. The dynamic ruleset is also stored in the database and retrieved from the database whenever it

AUTHOR

Christian Ney works as Systems Engineer for Security at Computacenter AG & Co. oHG, where networks, security, and firewalls are his daily bread.



USER MANAGER

USERNAME	TYPE	STATUS	DESCRIPTION
admin	admin	active	admin
chris	user	active	Christian Ney

Figure 1: User accounts are modified easily by adding rules.

is needed. Scripts running in the background regularly generate the corresponding iptables commands from database extracts and then add them to the ruleset.

To start, install the Ubuntu 8.04 server framework. If you opt for a more complex partitioning scheme for security or performance reasons, you are well advised to swap out the */var* filesystem, which is where both the logfiles and the database reside later on.

The Portsmith documentation tells you to install the ubuntu-desktop package macro, which provides a complete, Gnome-based user interface. However, you will not actually need a full Gnome desktop, unless the thought of using vi and the command line worries you. On the other hand, the documentation does not tell you anything about more meaningful tasks, such as basic hardening of the operating system. (Because Portsmith has a direct Internet connection, it requires far more attention to security than a normal end-user system.)

The next step is to configure the network interfaces and then go on to install the remaining packages. Users working on a graphical desktop can use the GUI tools for this; everyone else should just fire up their favorite editor and edit */etc/network/interfaces*.
The command

```
aptitude install -y ↩
openssh-server apache2 ↩
libapache2-mod-php5 ↩
libapache2-mod-auth-pgsql ↩
postgresql-8.3 php5-pgsql
```

drops the required packages onto your system. If you will be using the internal backup solution, you will additionally need the mkisofs and cdrecord packages.

Setting up the web servers takes a couple more steps. If you haven't configured DNS entries for your firewall, set the *ServerName* parameter, then enable the SSL module with *a2enmod ssl*.

To avoid transmitting user logins in the clear, it makes sense to use SSL-encrypted connections. Whether you use self-signed certificates or have them signed by a commercial CA is up to you and your company's security policy.

Portsmith

Portsmith is a 17MB ISO image download [6] , which you can either burn onto a CD or mount by typing:

```
mount -o loop ↩
/tmp/Portsmith_4.iso ↩
/media/cdrom
```

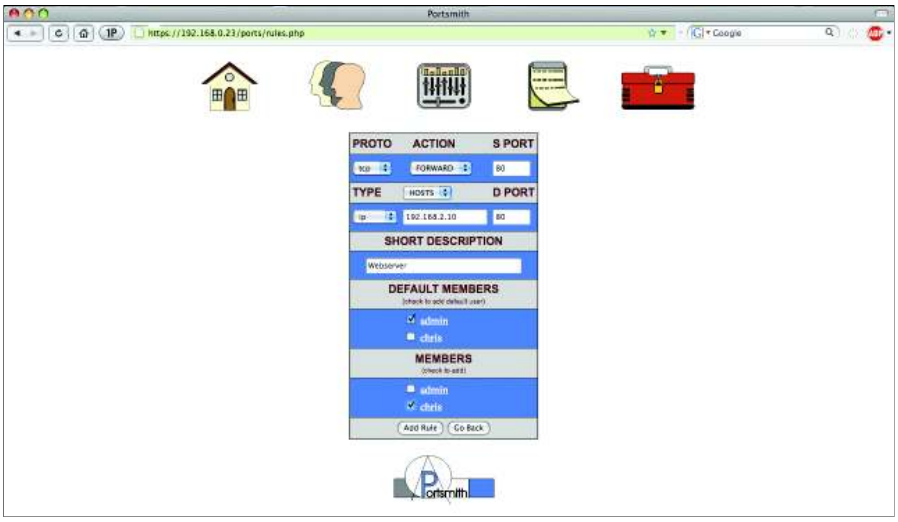
The Portsmith package includes a number of tarballs with shell and PHP scripts, the database table structure, and a sample configuration that helps you set up your Portsmith machine as a DNS or DHCP server. On top of this, you will find a short installation guide, as well as two Windows binaries (one for Internet Explorer and the other for Firefox) that support the use of RDP (Remote Desktop) via a Portsmith-empowered firewall. To install Portsmith, unpack the tarball:

```
tar xvf /media/cdrom/server/↩
ports.tar -C /
```

The shell scripts, which are copied into */usr/local/bin*, contain variables that reflect the network structure; the Portsmith admin will need to modify the scripts accordingly. The scripts are located in the *fw_policy* and *fw_lookup* files. Among other things, you need to specify the external network and the official firewall IP address. It is easier to have a static IP address; however, users with a dynamic IP can use DynDNS [7] or a scripting workaround to publish the current address.

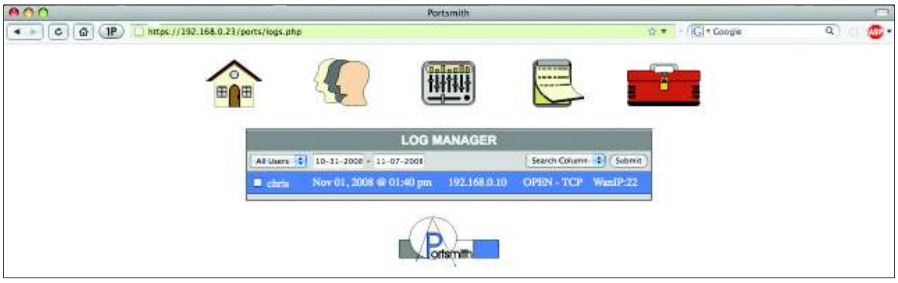
According to the documentation, you should launch Portsmith directly by adding it to your */etc/rc.local* file. If your firewall has a direct Internet link, this would mean the firewall and the networks hiding behind it would not be protected by the packet filter for a short period after you launch the machine because the firewall would be the last item processed at boot time. When you enable the network interfaces, you might prefer to run an *init* script or call Portsmith.

The database also needs some preparation. Unfortunately, the documentation tells users to cut and paste, an error-prone process; *psql* < */media/cdrom/server/TABLES.txt* is an easier approach. Because the hard work in the background is handled by a couple of shell scripts that need to run periodically, you



PROTO: tcp
ACTION: FORWARD
S PORT: 80
TYPE: HOSTS
ip 192.168.2.10
D PORT: 80
SHORT DESCRIPTION: Webserver
DEFAULT MEMBERS (check to add default user)
☑ admin
□ chris
MEMBERS (check to add)
□ admin
☑ chris

Figure 2: Modifying the ruleset: The example forwards port 80 on the firewall to an internal web server.



□ chris Nov 01, 2008 @ 01:40 pm 192.168.0.10 OPEN - TCP WanIP:22

Figure 3: The Log Manager tells you who enabled what rule.

need to set up cron entries for the scripts.

Relaunch the server by typing */etc/init.d/apache2 restart*, and Portsmith should be up and running.

Administrative Activities

Administrators can use their browser for all system management activities: *https://*Firewall*/ports/* takes you to a login window. The login defaults to *admin* with *admin* as the password. Although the documentation points out that you should change this, it could be too late if your firewall is already connected to the Internet – these defaults are very easy to guess.

After logging in, administrative users can configure the system, manage users, and enable a couple of features for themselves. The two main areas of system management are user administration and the ruleset. The list of users is fairly simple, and it gives administrators a large selection of entries and search functions (Figure 1). The drop-down list at top right lets you sort, add, modify, or delete accounts. When adding or modifying, you need to specify a username and password. The user is then assigned a role as a standard user or an admin; admins have full access to the system.

Instead of deleting accounts you no longer need, you can simply disable them. However, Portsmith does not have a time-controlled function for blocking temporary accounts; again manual attention by the admin is required. The admin can also set up a default rule, revoke rules, or add new rules. Ruleset management is similar to user management (Figure 2). Again, the Portsmith admin can search for, modify, add, and delete rules.

When adding or modifying, you can specify the protocol to use (TCP, UDP, ICMP), the action to perform (permit or forward), and the target port. Also, you can target a host via its IP or (assuming you have DNS name resolution working) its hostname. The admin interface also gives you access to a couple of useful tools: the Log Manager (Figure 3) tells you exactly which rules were enabled by which users from which IP. A handy search function helps you keep track of a larger number of log entries.

The tools button takes you to four more helpers that are worth examining. The Login Analyzer (Figure 4) not only gives the administrator a useful overview of who logged in when, but it also reports the number and times of recent invalid login attempts.

To avoid brute force attacks, the client source IP is blocked after 3 logins until the administrator resets it. The locking function does not rely on a packet filter; instead, the potential attacker is told *You are up to no good – you are now going to be blocked*, and the username and password fields disappear to prevent login attempts with an alternative user account. If worst comes to worst, the administrator will not even be able to use an administrative account to unblock the account, in which case, the only alternative is to modify the database manually.

Under normal circumstances, you can also release a computer blocked because of an excessive number of login attempts through the Login Analyzer. Unfortunately, this function is fairly well hidden: You need to enable the checkbox to the left of the entry and then click *Submit*.

The status display lists the current Firewall ruleset, although this is just the output from *iptables -L -v*. Anybody who is not familiar with iptables will probably be more confused after reading the list; a more readable report would be useful for newcomers.

The restart button restores the original status of any rules that were set by admins and users. This step keeps any existing connections.

User Viewpoint

Portsmith is really easy from the user's point of view. As with admin access, user access is browser-based, and users

Ruleset

A firewall is only as good as the ruleset implemented by its packet filter. In Portsmith's case, the default policy is similar to Listing 1.

The internal network, 192.168.2.0/24, which is hiding behind the *eth1* interface, can access the big, bad Internet without restrictions with port address translation. This approach might be fine for a home network or a small commercial setup, but it also poses many dangers.

The Portsmith homepage states "… all external ports are blocked until released after login," but the reality is a bit more complicated. Not only are ports 80 and 443 open for access to websites, but port 25 is open for mail traffic. On the other hand, any ICMP traffic reaching the external interface is blocked, although this sacrifices important troubleshooting information.

Portsmith uses the *RELATED* and *ESTABLISHED* states advantageously to automatically (statefully) allow the backchannel for existing TCP and DNS connections from the internal network. Source NAT, which is enabled for a generic IP address (192.168.1.250) poses more questions (Listing 2).

This appears to be the remnants of a forward to the LPD port of an internal print server. Although TCP port 9100 is blocked by default, the entry could cause hard-to-find problems in a similar setup.

Fortunately, you can modify the basic policy to match your own needs. Simply delete the defaults from */usr/local/bin/fw_policy*, or customize to create as complex an initial ruleset as you like. If you do not have basic iptables skills, you should turn to an expert for help.

After logging in a user and enabling an SSH connection for the user, the INPUT chain was extended as in Listing 3.

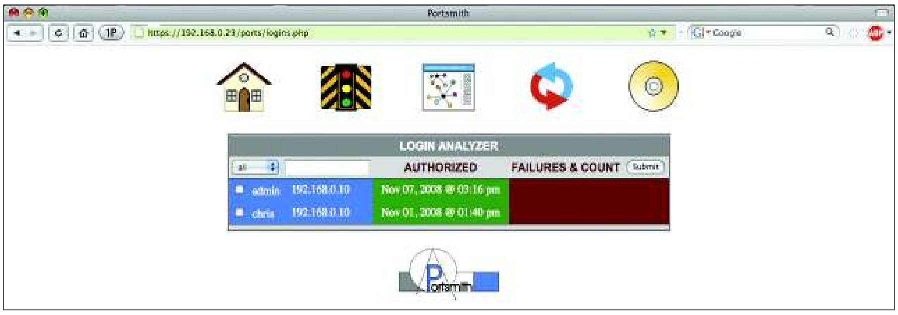
As you can see, dynamic rules come first. This avoids potential conflict with more generic existing rules. The destination (0.0.0.0) for the new rule is somewhat disturbing, in that Portsmith's own policy states that it should be restricted to an external IP.

need to enter their credentials. After logging in, users are shown a list of available firewall rules. A user can select a rule and then click it to enable.

As you can see from Figure 5, the list of enabled functions is fairly technical. Most end users are unlikely to understand the meanings of the individual ports and really do need a description in plain English. Some creativity is required from the administrator, since the database restricts the length of the description to 25 characters. Of course, you could change the size of the database column manually.

In contrast to a VPN, Portsmith does not send connections directly to the target machine; instead, it forwards incoming packets from the firewall once the user is authenticated. End users might therefore need to take some action. For example, if you want to access an internal printer remotely, you would have to set up a printer on the client side for the user to select.

It is a good idea to start with a clear definition of the tasks you want to handle with Portsmith (e.g., the previous example with the printer would be easy enough to implement, but you have to question the security of transmitting printed output in the clear across the Internet.)



Portsmith

https://192.168.0.23/ports/logins.php

LOGIN ANALYZER

all		AUTHORIZED	FAILURES & COUNT Submit
admin	192.168.0.10	Nov 07, 2008 @ 03:16 pm	
chris	192.168.0.10	Nov 01, 2008 @ 01:40 pm	

Figure 4: The Login Analyzer.

Security

The most critical issue with Portsmith security is that the web server is permanently accessible, more particularly so because it uses the PHP scripting language. If an attacker can compromise the authentication feature because of a vulnerability in the programming or the PHP language itself, the whole network becomes vulnerable.

The database link is another potential target. For example, imagine a black hat launching an SQL injection attack to “modify” the ruleset, making the protection the firewall is supposed to provide fairly useless.

Although research into Portsmith’s PHP has not revealed any potential vulnerabilities, one cannot rule out the possibility of a successful attack. It goes without saying that you should update often. For more protection, you might also want to install Suhosin [8] or Mod-Security [9] to give you some defense against zero-day exploits.

The system is based on allowing an authenticated client’s IP address for a connection. The source IP arriving at port 443 on the web server is referenced to ascertain the address. This behavior is also a potential source of danger: In the simplest case, the client might be unaware that it is hiding behind a proxy that sets up a connection to the Portsmith firewall for the web browser. This would lead to the proxy address, rather than the client address, being enabled. In turn, this scenario means that any user on the proxy is assigned the same privileges as the user client-side and, thus, can access the client user’s resources.

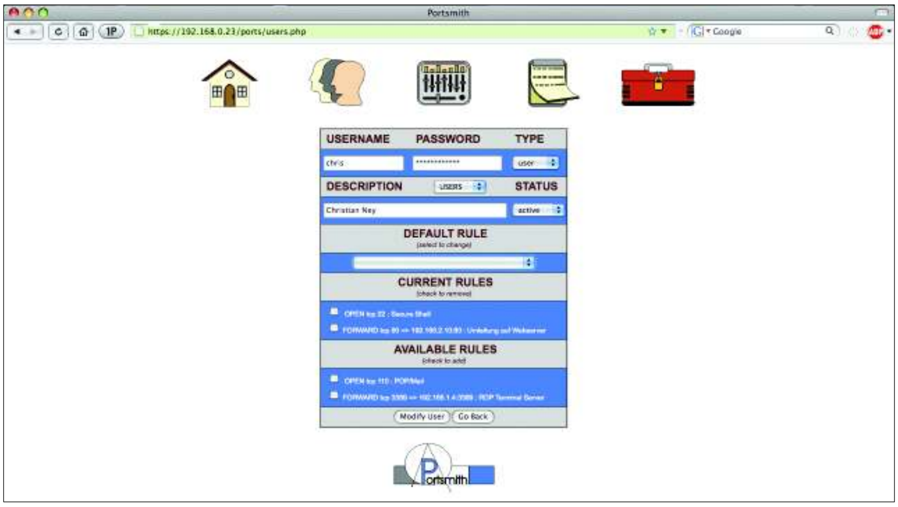
A situation in which an attacker works through a compromised proxy server would be even worse. Considering how long the period of access is granted, this would again leave a gaping hole open for attacks.

Practical Supplement

Portsmith is a practical supplement to a conventional firewall. The solution adds dynamic, predefined enabling to the traditionally static ruleset, allowing users

Listing 1: Portsmith Filter

```
Chain INPUT (policy DROP 22 packets, 3590 bytes)
 pkts bytes target     prot opt in     out     source               destination
    7   476 ACCEPT     all  --  eth1   *       192.168.2.0/24       0.0.0.0/0
  200 17460 ACCEPT     all  --  lo     *       127.0.0.1            0.0.0.0/0
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           limit: avg 5/sec burst 5 tcp dpt:80
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           limit: avg 5/sec burst 5 tcp dpt:25
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           limit: avg 5/sec burst 5 tcp dpt:443
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    1    62 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           limit: avg 15/sec burst 5 udp spt:53 state
RELATED,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       192.168.2.0/24       0.0.0.0/0           limit: avg 5/sec burst 5
Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       192.168.2.0/24       0.0.0.0/0
    0     0 ACCEPT     all  --  *      *       127.0.0.1            0.0.0.0/0
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
Chain OUTPUT (policy ACCEPT 233 packets, 21730 bytes)
 pkts bytes target     prot opt in     out     source               destination
```



USERNAME PASSWORD TYPE
chris ************ user
DESCRIPTION USERS STATUS
Christian Ney active
DEFAULT RULE
(select to change)
CURRENT RULES
(check to remove)
□ OPEN tcp 22 : Secure Shell
□ FORWARD tcp 80 => 192.168.2.10:80 : Umleitung auf Webserver
AVAILABLE RULES
(check to add)
□ OPEN tcp 110 : POP/Mail
□ FORWARD tcp 3389 => 192.168.1.4:3389 : RDP Terminal Server

Figure 5: Not so easy for non-technical users: setting enabled services.

to initiate changes. One potential use would be hardening access to an internal web mailer or other services, such as Sun Secure Global Desktop [10]. With encrypted connections, Portsmith could serve as a VPN replacement. If you prefer a genuine VPN solution, you can easily add IPSec software, such as strongSwan [11] or OpenVPN [12].

Simply using Portsmith to grant SSH firewall access to a restricted group of users is very much over the top. Port knocking [13] is a simpler and more secure alternative. Users with dynamic IP addresses need to rely on manual techniques or an external tool such as a DynDNS service to update the configuration with an active, official IP address.

The default ruleset poses a couple of questions and could definitely be improved in places. Fortunately, changes are made easily, although a change does require manual editing of the Portsmith configuration files. Manual editing can cause complications when you update.

With respect to authentication, Portsmith unfortunately relies on its own database-oriented approach. Use of an RSA token system or single sign-on would be preferable. This would not only mean more convenience for the user, but (especially in the case of RSA) also an extra layer of security.

It would also be useful to introduce time-based controls to automatically lock temporary accounts, for example, or restrict logins to normal office hours for some users. One would hope that the system is flexible enough to support extensions of this kind.

INFO

[1] Portsmith: *http://ddbolt.net/portsmith.php*

[2] Ubuntu server download: *http://www.ubuntu.com/getubuntu/download*

[3] Apache Web Server: *http://httpd.apache.org*

[4] Hypertext preprocessor: *http://www.php.net*

[5] PostgreSQL: *http://www.postgresql.org*

[6] Portsmith download: *http://ddbolt.net/downloads/Install_CD*

[7] DynDNS: *http://www.dyndns.org*

[8] Suhosin: *http://www.hardened-php.net/suhosin.127.html*

[9] ModSecurity: *http://www.modsecurity.org*

[10] Sun Secure Global Desktop: *http://www.sun.com/software/products/sgd/index.jsp*

[11] strongSwan: *http://www.strongswan.org*

[12] OpenVPN: *http://openvpn.net*

[13] Port Knocking: *http://en.wikipedia.org/wiki/Port_knocking*

Conclusion

All told, and assuming a working basic configuration and more hardening, Portsmith is a useful extension to any iptables-based firewall that is likely to make life easier for both administrators and users. ■

Listing 2: Source NAT

```
Chain PREROUTING (policy ACCEPT 1603 packets, 117K bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DNAT       tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           tcp dpt:9100 to:192.168.1.250:9100
```

Listing 3: Modified Rules

```
Chain INPUT (policy DROP 122 packets, 14737 bytes)
 pkts bytes target     prot opt in     out     source               destination
   85  7589 ACCEPT     tcp  --  eth0   *       192.168.0.10         0.0.0.0/0           tcp dpt:22
   14   961 ACCEPT     all  --  eth1   *       192.168.2.0/24       0.0.0.0/0
 2497  640K ACCEPT     all  --  lo     *       127.0.0.1            0.0.0.0/0
   23  3652 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           limit: avg 5/sec burst 5 tcp dpt:80
    0     0 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           limit: avg 5/sec burst 5 tcp dpt:25
  262 37622 ACCEPT     tcp  --  eth0   *       0.0.0.0/0            0.0.0.0/0           limit: avg 5/sec burst 5 tcp dpt:443
  114 22241 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    3   204 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           limit: avg 15/sec burst 5 udp spt:53
 state RELATED,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       192.168.2.0/24       0.0.0.0/0           limit: avg 5/sec burst 5
```