Stacki Tutorial #2: Disabling Host Firewalls

Introduction

By default, Stacki installs a host firewall on each backend server, and configures the firewall to allow traffic on the private subnet, and only connections associated with the private subnet on the public subnet. And ssh is allowed anywhere, so you can administer the server.

Of course if you have added a network card for eth1 and put it on the public network, this might not be your optimal configuration.

What you will learn

This short tutorial will focus on disabling the host firewall on backend servers.
Specifically, when done with this tutorial, you will be able to:

- Disable Firewalls on an installed host.
- Disable Firewalls on all installed hosts.
- Disable activation of firewalls on an installed host.
- Disable activation of firewalls for all installs.

While there is a ton more possible, and we touch on several of the other parts of the Stacki command line, in keeping with the goals of this tutorial series we are going to stay lazer focused on disabling the firewall in Stacki, so you have all the information you need about how Stacki manages firewalls and disabling them, and we can keep this tutorial at a reasonable length.

What Stacki Does For Host Firewalls

Stacki wraps iptables, so that single firewall configuration commands can be used across machines to configure firewalls correctly and consistently. To do this, Stacki presents a command line that you interact with to change firewall configuration by adding rules to iptables.

The default configuration is to allow all traffic within the private network, and only allow traffic on the public network that is associated with a private connection. This is a secure and reliable configuration, but may not meet the needs of your organization. The default configuration for CentOS/RHOS 6.X allows SSH into both public and private, while for CentOS/RHOS 7.X allows both SSH and ICMP echo-request/echo-reply.

Stop That!

The first, and easiest thing to learn is how to disable firewall configuration in Stacki. On the stacki server, type the command:

```
stack list attr attr=firewall
```

Will give you the global firewall attribute and its value.

```
[root@stackidon ~]# stack list attr=firewall
-------------------- firewall       true           G
```

As you can see, firewalling hosts is turned on by default. The "G" stands for "global", which we'll talk more about in a bit.

If that doesn't suit the needs of your environment, turning it off *before* installing is as simple as setting the global firewall attribute to false:

```
[root@stackidon ~]# stack set attr attr=firewall value=false
```

This command line (and all set attribute command lines) does not display any results unless there is an error.
Once the above command line has been executed, *every* backend server install that stacki does from that point forward will have firewalls disabled. This includes reinstalls. The only exceptions to this rule are (a) if you've over-ridden the value for a particular host (see below), and (b) obviously if you ever change the value back to true, firewalls will begin installing again.

If you prefer to know that you got it right, simply get the value again from stacki:

```
[root@stackidon ~]# stack list attr | attr=firewall
-------------------- firewall       false          G
```

As you can see, the global firewall attribute is now set to false.
What this means is that on install, iptables will not be configured and run at all.

If you have already installed your servers and want to disable firewalling, you still can. First follow the steps above to disable the firewall on future installs. Then execute the following on the stacki server:

CentOS 6.X

```
[root@stackidon ~]# stack run host backend command='/etc/init.d/iptables stop;chkconfig iptables
off'
```

CentOS 7.X

```
[root@stackidon ~]# stack run host backend command='systemctl stop iptables:systemctl disable iptables'
```

In both cases, we are simply visiting each host and stopping iptables, then telling the system to disable iptables so that on future boots it does not start automatically.

The run command finds all servers that match the host name (for host name discussions, see the call-out box at this link), and executes the *command* on each server.

That's it! If you've followed these steps, firewalls should be disabled on all backend nodes, and any new nodes you install will have firewalls disabled by default.

Disabling the Firewall on a Single Server

Of course, you can disable firewalling on a single system while leaving it in place for all others by using the following command:

```
[root@stackidon ~]# stack set host attr backend-0-0 attr=firewall value=false
```

If this is done before the host is installed, that's all you need to do. *To disable on a single server do not change the value of the global firewall attribute. That disables for all future installations.*

If the host is already installed, you still need to execute the above command so that if Stacki ever re-installs the host it knows how to treat the firewall, but you also need to stop the already-configured firewall from running on the host:

CentOS 6.X

```
[root@stackidon ~]# stack run host backend-0-0 command='/etc/init.d/iptables stop;chkconfig iptables off'
```

CentOS 7.X

```
[root@stackidon ~]# stack run host backend-0-0 command='systemctl stop iptables:systemctl disable iptables'
```

Where you can substitute the hostname for whichever machine needs the firewall disabled in the place of "backend-0-0" in the above commands.

This works because the global value for firewall is a default, and an individually configured value at the host level will over-ride the default. This is the design of Stacki overall, and you will see more of this theme in coming tutorials.

Summary

Host-based firewalls, and if they are used, tends to be a very local decision. There is no real standard for deployment of host firewalls, many organizations feeling network firewalls provide enough protection, and others not wanting the internal servers locked down so tightly that they cause problems. This tutorial showed you just the tip of the iceberg in terms of firewall configuration. Turning firewall activation on and off is just one of many Stacki commands relevant to firewalls, but if your organization has a policy of not allowing host firewalls, it is the most important. In future installments we'll cover other firewall functionality, so that you can get the most from Stacki's functionality.