How to Secure CENTOS 7.1
Part 2

Motivation

This paper is part of a multi-part series on securing CentOS 7.1. This paper focuses on implementing firewalld as a host based firewall. If you need an enterprise class firewall, I would recommend first reading Next Generation Firewall for Dummies and researching as a starting place this company's products: (http://www.paloaltonetworks.com/products/overview/). The intended audience for this paper is laymen and journeymen system administrators that need a quick reference and solid solutions for common host based firewalls. If you have comments or feedback on how I can represent this better, please email me your ideas to the email address listed on the website (securityhardening.com).

Principles

Effectiveness of a firewall

Firewalls are only one part of the security of a system. Implemented correctly, they reduce the risk of a system or network being remotely compromised, but they do not remove all threat. Insiders are one of the greatest threats to an organization because of their ability to circumvent policy that is weakly implemented. Do not dismiss Firewalls, in this case we want to look at what a host based firewall can do to reduce the threat of remote attacks. Implemented correctly, the firewall can prevent direct connections from remote attacks.

Is it a router or a Firewall

For early Firewall's, they were performing the same functionality as a router. They routed packets and blocked based off of the implemented rules. The idea was to offload from the routing device and have one central location where most rules were located. This idea is at leat 20+ years old and does not keep up with the threats coming from the Internet today.

Top Down or Bottom Up reading of rules

This is an important point to understand. Most of the older firewalls read from top-down and used the first rule that it matched. Today's firewalls have optimization rules that put them into the most efficient order. For IPtables, it will be reading from top-down, so putting the deny-all rule at the bottom make the most sense.

Block all by default and allow only the known through the firewall

Allow only those services that you expressly permit. Think of a service as the end TCP or UDP port that you wish to connect to from a source IP address. Do not allow any direct TCP/IP connections between applications on internal systems and servers on the Internet. The notion that I operate under is that I never trust connections from un-trusted systems. Be aware that another system can impersonate a secure system. When attackers use this type of attack, they impersonate a trusted known host on your network. This impersonation, which is also called IP spoofing, allows the host to bypass your security controls to connect to your network. Finally, the routing table update that you receive from a neighboring router may redirect your network traffic to an unintended destination.

Point-to-Point connections

Establish a controlled link from trusted IP addresses. If I am writing a rule on a database server, I only wish to accept the database port as a destination port from the IP address of the web server(s). I would also then only accept SSH connections from the group of workstations that I use to manage and maintain the database server. What other ports would you need to have open? I would suggest none, but you know your organization better than I do and therefore must adapt to meet your organizational goals.

Deploying applications in a split zone

DMZ and Trusted zones with a firewall protecting the zones. DMZ and Trusted zones are still a worthwhile thought process at this point in time.

Using a screening router to offload some rules

If your organization has the money, I recommend to deploy a screening router on the outside of the firewall that allows known bad traffic to be discarded before getting to the main firewall. Such as preventing ICMP traffic originating from the internet to the internal network or firewall -- but allow ICMP outbound. Using an internal router to only allow trusted protocols in the Trusted zone (this can be performed on smart switches that understand layer 2-4 traffic).   Securing the switches to handle layer 4 traffic and limit only trusted ports and protocols (if a system is infected, it can't get past the switch port).

KISS -- Keep it Simple Stupid!

Keep the rules minimal and straight forward to meet your organizations goals while being usable. One of the stupidest things I've ever done was loading 600,000+ rules into 16MB of memory. The rules loaded and worked, but every packet going out had to be compared against this colossal set of rules, top-down and match one of the rules. My connections for an application like Internet Explorer would take at least 3-8 minutes just to start and connect to Google.com. This is unacceptable in my opinion. Minimize the rules and rely on the default block to protect the system. Use a holistic approach, such as hardening the Operating System, host based firewalls, anti-virus protection, regular log reviews, and back ground checks on all employees. It really comes down to a trust issue; how much do you really trust the people that maintain your systems? Keep the rules simple and do the right thing every day!

Implementation

Copy the below rules examples into /etc/firewalld/services/<service>.xml.

Restart the service with:

```
systemctl restart firewalld
```

Make sure that the service restarts on reboot:

```
systemctl enable firewalld
```

Commands you need to know:

State of the firewall:

```
# firewall-cmd -state

running
```

Show active zones

```
# firewall-cmd --get-active-zones

public
  interfaces: enp0s3 enp0s8
```

Show a zones rules

```
# firewall-cmd --zone=public --list-all

public (default, active)
  interfaces: enp0s3 enp0s8
  sources:
  services: dhcpv6-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

Show all services

```
# firewall-cmd --get-services

RH-Satellite-6 amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client
dns ftp high-availability http https imaps ipp ipp-client ipsec kerberos
kpasswd ldap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nfs ntp
openvpn pmcd pmproxy pmwebapi pmwebapis pop3s postgresql proxy-dhcp radius
rpc-bind samba samba-client smtp ssh telnet tftp tftp-client transmission-
client vnc-server wbem-https
```

Panic Mode (in case you are hacked):

```
# firewall-cmd --panic-on
```

Panic Mode (exit/quite):

```
# firewall-cmd --panic-off
```

Enable SSH:

```
# firewall-cmd --zone=public --add-port=22/tcp
```

Set Default Zone:

```
# firewall-cmd --set-default-zone=public

22/tcp
```

Show all Zones:

```
# firewall-cmd --list-all-zones | less

block
  interfaces:
  sources:
  services:
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

dmz
  interfaces:
  sources:
  services: ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

drop
  interfaces:
  sources:
  services:
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

external
  interfaces:
  sources:
  services: ssh
  ports:
  masquerade: yes
  forward-ports:
  icmp-blocks:
  rich rules:

home
  interfaces:
```

```
  sources:
  services: dhcpv6-client ipp-client mdns samba-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

internal
  interfaces:
  sources:
  services: dhcpv6-client ipp-client mdns samba-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

public (default, active)
  interfaces: enp0s3 enp0s8
  sources:
  services: dhcpv6-client ssh
  ports: 22/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

trusted
  interfaces:
  sources:
  services:
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:

work
  interfaces:
  sources:
  services: dhcpv6-client ipp-client ssh
  ports:
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

Firewall Lockdown

Local applications or service s are able to change the fire wall configuration if the y are running as root (for example , libvirt). With this feature, the administrator can lock the firewall configuration so that either no applications , or only applications that are added to the lockdown white list, are able to re quest firewall changes . The lockdown settings default to disabled. If enabled, the user can be sure that there are no unwanted configuration changes made to the firewall by local applications or services .

Edit /etc/firewalld/firewalld.conf and append:

```
Lockdown=yes
```

Reload the firewall:

```
# systemctl firewalld restart
```

Changing the Zone of your Interface Permanently

Interfaces will always revert to the default zone if they do not have an alternative zone defined within their configuration. On CentOS, these configurations are defined within the /etc/sysconfig/network-scripts directory with files of the format ifcfg-<interface>.

To define a zone for the interface, open up the file associated with the interface you'd like to modify. We'll demonstrate making the change we showed above permanent:

```
sudo vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

At the bottom of the file, set the ZONE= variable to the zone you wish to associate with the interface. In our case, this would be the "home" interface:

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DNS1=8.8.4.4
DNS2=8.8.8.8
ZONE=public
```

Restart the services:

```
sudo systemctl restart network.service
sudo systemctl restart firewalld.service
```

Add Services

You can enable a service for a zone using the --add-service= parameter. The operation will target the default zone or whatever zone is specified by the --zone= parameter. By default, this will only adjust the current firewall session. You can adjust the permanent firewall configuration by including the --permanent flag.

For instance, if we are running a web server serving conventional HTTP traffic, we can allow this traffic for interfaces in our "public" zone for this session by typing:

```
sudo firewall-cmd --zone=public --add-service=http
```

Once you have tested that everything is working as it should, you will want to modify the permanent firewall rules so that your service will still be available after a reboot. We can make our "public" zone change permanent by typing:

```
sudo firewall-cmd --zone=public --permanent --add-service=http
```

Testing and Troubleshooting

Testing is absolutely essential when dealing with firewall rules. When you deploy a new application on CENTOS 7.1, turn off the firewall rules to make sure the application works 100% correctly, then turn the firewall back on and test again. If something doesn't work, that means you need to add a new port number into the rules. Use the command, "netstat -an" to get a list of open ports with the firewall turned off and the application running. I like using the command "watch" to help with this, as in:

```
watch -d -n 5 netstat -pan -A inet,inet6
```

Output:

```
Every 5.0s: /bin/netstat -pan -A inet,inet6                                   Sun Nov 22 20:47:07
2015

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State  PID/Program name
tcp        0     0 0.0.0.0:22              0.0.0.0:*               LISTEN  1231/sshd
tcp        0     0 127.0.0.1:25            0.0.0.0:*               LISTEN  1360/master
tcp        0     0 10.20.10.33:22          10.20.10.1:42675        ESTABLISHED 1369/sshd: masterf
tcp6      0      0 :::22                   :::*                    LISTEN  1231/sshd
tcp6      0      0 ::1:25                  :::*                    LISTEN  1360/master
raw6      0      0 :::58                   :::*                    7           738/NetworkManager
raw6      0      0 :::58                   :::*                    7           738/NetworkManager
```

I find it imperative when troubleshooting firewall rules to know exactly how the system is handling and using the rules. Monitor your log files, /var/log/messages, to understand what's being blocked and adjust the rules to fit your organizational policy.

Conclussion

Deploying the local firewall will prevent almost all unwanted remote connections. There is a slight management overhead introduced with these rules, but the benefit outweighs the cost. These are in no way the magic silver bullet to protect everything and need to again be combined with "a holistic approach, such as hardening the Operating System, host based firewalls, anti-virus protection, regular log reviews, and back ground checks on all employees". Failure to implement a full set of deterrence's will result in a weaker target that will cost the organization time, money and possibly customers or embarrassing data loss highlighted on the nightly news and worst case, in front of congress.