Wicked Quick Linux Firewalls

Create a simple host firewall for a Linux server that whitelists the processes that can make outgoing connections. The whitelist will be based on a new group you create. Processes in the new group are the only ones which may make new outgoing connections. By monitoring the processes in this group, you can be sure to stop any red team processes from connecting back to C&C.

Requirements

- Nftables
  - `sudo apt install -y nftables && sudo systemctl enable --now nftables && sudo nft flush ruleset` # on Ubuntu
  - `sudo yum install -y nftables && sudo systemctl enable --now nftables && sudo nft flush ruleset` # on CentOS
- Sudo
- Git (opt)
- Tmux (opt)

Instructions

1. Remove/disable any other firewall on the box. The two might conflict.
   a. If you are using Docker, don't disable IPtables, just make sure the input and output chains are set allow all (to view: iptables -L).
   b. On CentOS, run `> systemctl disable --now firewalld`
   c. On Ubuntu, `run > systemctl disable --now ufw`
2. Create a new group. It should have a short, but unique, name.
   a. `> sudo groupadd <group>`
3. Edit `` `/etc/sudoers` `` so that you can run commands with your new group. You need to use the `` `visudo` `` command to do this.
   a. Do b. for the sudoers group that you have enabled (sudo or wheel), and **also** for the root user.
   b. Change the line that reads "`%<group or user> ALL=(ALL) ALL`" to "`%<group or user> ALL=(ALL:ALL) ALL`"
   c. If there is an error when you exit the editor, try again but more carefully
   d. Here's an example:
      i. `BEFORE: %sudo ALL=(ALL) ALL`
      ii. `AFTER: %sudo ALL=(ALL:ALL) ALL`
   e. Now you can run a process with a specific group with the command `> sudo -g <group> <command>`

4. Add ports that handle **incoming** connections to the firewall rules. Run `> sudo ss -tulpn` to checking for active incoming connections. If anything is suspicious, document it and **do not** add that port to the firewall. Add trusted services' ports under "#to allow incoming ports …" in the firewall rules.
5. Make sure services run as the correct group. Some services may need to make **outgoing** connections. This may include: DNS, Docker, Filebeat, Apache (PHP). These services need to run with the group you created. Services that only listen, like Nginx, sshd, etc. DO NOT need this config.
    a. `> systemctl edit <service>` `# opens an editor where you can write text`
    ```
    [Service]
    Group=fw_out
    ```
    b. For each service that needs to send out traffic, copy the file at `/usr/lib/systemd/system/filebeat.service` (or whatever) to `/etc/systemd/system/filebeat.service`.
    c. Add the entry "`Group=fw_out`" under the "Service" section of the file.
    d. Run `systemctl daemon-reload`
    e. Restart each service
6. Write the firewall file
    a. See the appendix for some examples.
    b. Hopefully you can pull it in with `git` once someone else has written one.
7. Apply the firewall (best done in tmux)
    a. If you have local access, simply run `sudo nft -f <filename>`. If not, continue to next step
    b. Run > sudo nft -f <filename> && sleep 30 && sudo nft flush ruleset &
    c. Wait a few seconds
    d. Attempt to cancel the command
    e. If you cannot cancel it, wait until the firewall drops after 30 seconds and debug your configuration.

(note that if this appears to fail, make sure the kernel module isn't disabled in /etc/modprobe.d)

8. Test networking
    a. Any command that make an outgoing connection need to be run with `sudo -g <group>`
    b. Try `` `sudo -g <group> dig @1.1.1.1 cnn.com +short` ``.
    c. Try `` `sudo -g <group> curl www.google.com` ``.
    d. Try sshing into another one of your servers.
    e. If you have a GUI, try running a web browser
9. Make the nftables rules permanent. Copy your firewall file to `/etc/nftables.conf`
10. Define an alias for convenience. Saves you typing
    a. `> alias out="sudo -g <group>"`
    b. Now you can run "out <command>" instead of "sudo -g <group> <command>
    c. If you like this alias, put it in your `~./bashrc` file
11. This firewall does NOT stop existing connections! Check all the active connections

Troubleshooting

Yum doesn't work
Run your yum command as the fw_out group, and as the root user via
`sudo -g fw_out -u root <yum cmd>`

APT doesn't work
By default, APT will not work with this configuration as it sandboxes itself by dropping its process to a different user. APT uses the account "_apt" to restrict its privileges, so we can whitelist this account in our firewall. Add the following to the output chain:

```
skuid _apt log prefix "APT_OUT " flags all accept;
```

Another solution to this problem is to change _apt's primary group to the output group, and make it's old primary a secondary group:

```
usermod -g <group> _apt
usermod -aG nogroup _apt
```

Note that this could allow compromised code to run as the "_apt" account. Watch the logs for weird traffic with the "APT_OUT" prefix.

Docker or other services don't work
In the service file, (/lib/systemd/system/ for docker), under [Service], add Group = <group>

I need to add another service!
Under input or output add:
udp dport { <first port>, <second port>, <third port …> } accept;
tcp dport { <first port>, <second port>, <third port …> } accept;

Security

See what users are in the group you created
There should be no users in the group you created, only active processes. To verify this, try
> `sudo lid -g <group>`
Or try
> `grep <group> /etc/group`

See what processes are in the group you created
Find the id of your group
> `getent group <group>`
> `ps -eo user,pid,gid,cmd`

Or as a one-liner
`> ps -eo user,pid,gid,cmd | grep $(getent group <group>)`

See logs of what has been blocked by nftables
Any time an Nftables rule has the statement “log” the kernel will generate a log entry. In most configurations, this log entry will be picked up by syslog. There are no logs generated by default, so be sure to add the “log” statements to your firewall config.

In CentOS, check /var/log/messages.
In Ubuntu, check /var/log/syslog
In Kibana with Filebeat, run the query: “system.syslog.program:kernel MAC”

Note: tcpdump will NOT see dropped packets

Example Firewalls

Save me as firewall.nft (You can skip the commented lines (<u>but not the first one!</u>). Tabs don’t matter.)

```
#!/usr/bin/nft -f
flush ruleset;
table inet filter {
	chain input {
		type filter hook input priority 0; policy drop;
		iifname "lo" accept;

		ct state { established, related } accept;
		ct state invalid drop;
		#allow ping
		 ip protocol icmp accept;
		ip6 nexthdr icmpv6 accept

		#to allow incoming ports for services you are running
		tcp dport { 22 } limit rate 15/minute log prefix “new_ssh “ flags all accept;
		tcp dport { 5901 } accept; #example for web server
	}

	chain output {
		type filter hook output priority 0; policy drop;
		oifname "lo" accept;

		ct state invalid drop;
		ct state { established, related } accept;
```

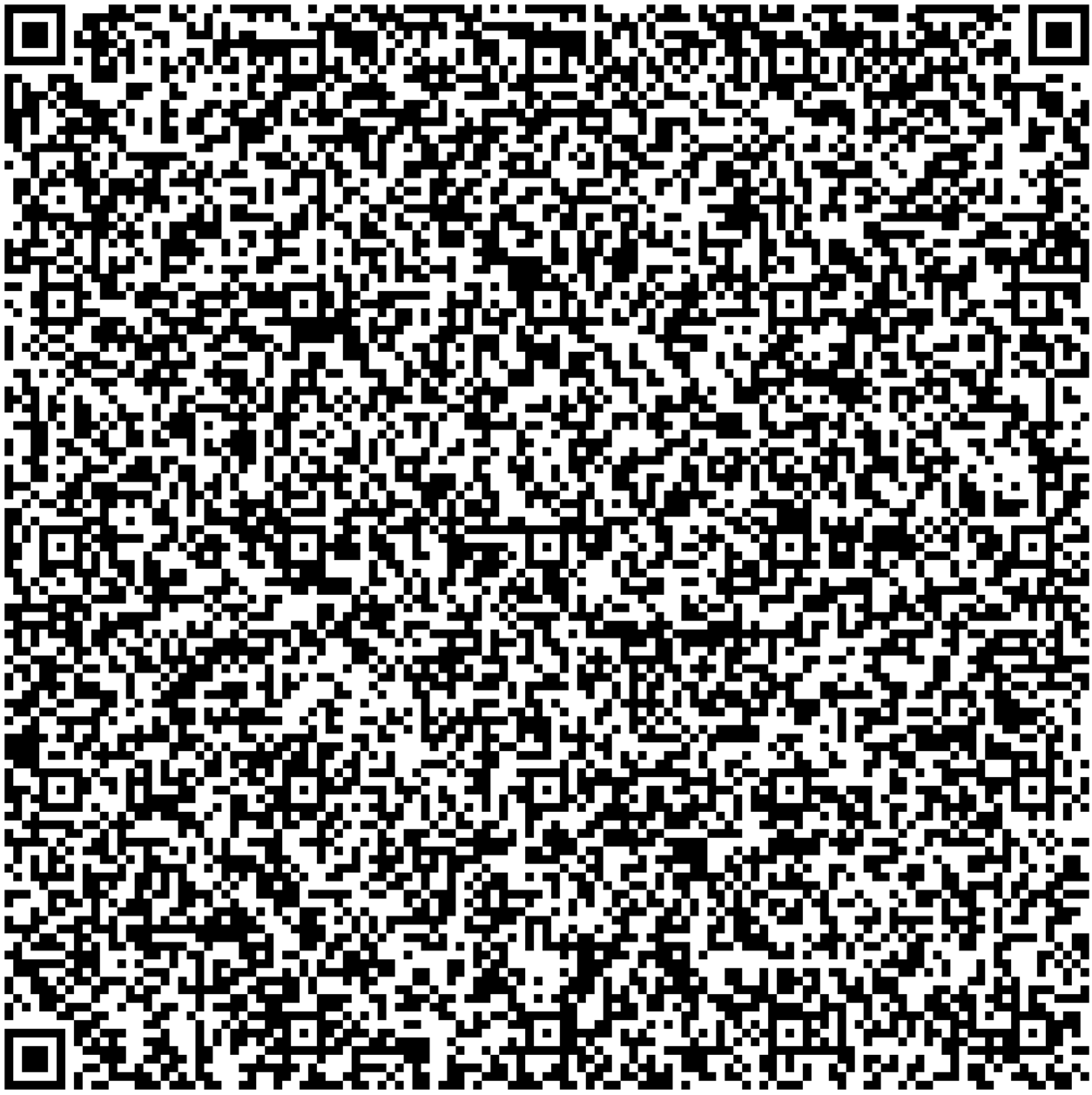
```
        #allow ping
        ip protocol icmp accept;
        ip6 nexthdr icmpv6 accept;

        # Assumes your firewall group is called “fw_out”. Replace with a different
        # name as needed.
        skgid fw_out jump allowed_output;

        # uncomment and add your DNS servers if you have resolution problems
        # necessary if using systemd-resolved
        ip daddr { 1.1.1.1, 8.8.8.8 } accept;
        log prefix “dropped_output “ flags all;
        udp dport { 53 } accept;
        tcp dport { 53 } accept;
    }

    chain allowed_output {
        log prefix “allowed_output “ flags all;
        udp dport { 53 } accept;
        tcp dport { 22, 53, 80, 443, 8080, 5601, 9200, 9300 } accept;
        drop;
    }

    chain forward {
        # needs to be accept for docker to work
        type filter hook forward priority 0; policy accept;
    }
}
```



The firewall as a QR code (01/22/20), thanks https://www.the-qrcode-generator.com/