Design

This script has been designed with extensibility in mind. It is clearly commented with sections for TCP, UDP code under each chain.
At the top we define the input rules for TCP and UDP individually and then the output rules. We label the chains:

- TCP_INPUT_RULES
- UDP_INPUT_RULES
- TCP_OUTPUT_RULES
- UDP_OUTPUT_RULES

The accounting is set up by implementing:

- SSH_TRAFFIC
- WWW_TRAFFIC
- OTHER_TRAFFIC

Chains which are called from our previously defined rules.

Testing

The testing procedure is simple. I will disable iptables on the firewall machine and scan it with nmap. Then I will enable iptables and run the scan again. This should show which ports are available and if my configuration is correct.

Case	Description	Tool	Expectation	Result
1	Scan a machine with no firewall enabled.	Nmap, iptables	The machine will have either no ports open or many ports open depending on configuration.	1716/tcp was open on the machine. Result is in line with expectations.
2	Scan machine with iptables script enabled.	Nmap, iptables	The machine will have web ports open, SSH port open as well as DHCP and DNS ports open.	Web ports were open. SSH ports detected once but never again. Error logs displayed sshd process crashing. It was assumed that the testing caused the crash.

3	Wireshark testing of web traffic.	Wireshark, iptables, web browser.	The firewall will block all web traffic on ports other than 80 and 443. Some sites may not load correctly.	In the included pcap file we can see the firewall block packet 1211 because it does not match the outbound rules.

Disabling firewall:

```
~/Documents/c8006/a1(master) » sudo iptables -F

~/Documents/c8006/a1(master) » sudo iptables -X

~/Documents/c8006/a1(master) » sudo iptables -P INPUT ACCEPT

~/Documents/c8006/a1(master) » sudo iptables -P FORWARD ACCEPT

~/Documents/c8006/a1(master) » sudo iptables -P OUTPUT ACCEPT

~/Documents/c8006/a1(master) » sudo iptables -L -x -n -v
Chain INPUT (policy ACCEPT 21 packets, 5218 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 24 packets, 3206 bytes)
    pkts      bytes target     prot opt in     out     source               destination

~/Documents/c8006/a1(master) »
```

First scan:

```
~ » nmap -p- --min-parallelism 100 -v 192.168.1.64

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-31 23:49 PST
Initiating Ping Scan at 23:49
Scanning 192.168.1.64 [2 ports]
Completed Ping Scan at 23:49, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:49
Completed Parallel DNS resolution of 1 host. at 23:49, 0.03s elapsed
Initiating Connect Scan at 23:49
Scanning 192.168.1.64 [65535 ports]
Discovered open port 1716/tcp on 192.168.1.64
Completed Connect Scan at 23:49, 10.45s elapsed (65535 total ports)
Nmap scan report for 192.168.1.64
Host is up (0.041s latency).
Not shown: 65534 closed ports
PORT     STATE SERVICE
1716/tcp open  xmsg

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 10.56 seconds
```

Running firewall / port forwarding rules via "*sudo bash ipt.sh*":

```
Chain FORWARD (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 TCP_OUTPUT_RULES  all  --  *      *       0.0.0.0/0            0.0.0.0/0
       0        0 UDP_OUTPUT_RULES  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain OTHER_TRAFFIC (0 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     all  --  *      *       0.0.0.0/0           0.0.0.0/0

Chain SSH_TRAFFIC (6 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0           0.0.0.0/0
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0           0.0.0.0/0

Chain TCP_INPUT_RULES (1 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0           tcp dpt:0
       0        0 DROP       tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:80
       0        0 DROP       tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0           tcp spts:0:1023 dpt:443
       0        0 SSH_TRAFFIC  tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             tcp dpt:22
       0        0 WWW_TRAFFIC  tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             tcp dpt:80
       0        0 WWW_TRAFFIC  tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             tcp dpt:443
       0        0 SSH_TRAFFIC  tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             tcp spt:22
       0        0 WWW_TRAFFIC  tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             tcp spt:80
       0        0 WWW_TRAFFIC  tcp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             tcp spt:443

Chain TCP_OUTPUT_RULES (1 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 SSH_TRAFFIC  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22 dpt:22
       0        0 WWW_TRAFFIC  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
       0        0 WWW_TRAFFIC  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443
       0        0 WWW_TRAFFIC  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:80
       0        0 WWW_TRAFFIC  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:443

Chain UDP_INPUT_RULES (1 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 DROP       udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0           udp dpt:0
       0        0 DROP       udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0           udp spts:0:1023 dpt:80
       0        0 DROP       udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0           udp spts:0:1023 dpt:443
       0        0 SSH_TRAFFIC  udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             udp dpt:22
       0        0 WWW_TRAFFIC  udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             udp dpt:80
       0        0 WWW_TRAFFIC  udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             udp dpt:443
       0        0 SSH_TRAFFIC  udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             udp spt:22
       0        0 WWW_TRAFFIC  udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             udp spt:80
       0        0 WWW_TRAFFIC  udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0             udp spt:443
       0        0 ACCEPT     udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0           udp dpt:53
       0        0 ACCEPT     udp  --  enp0s31f6 *       0.0.0.0/0            0.0.0.0/0           udp spts:67:68 dpts:67:68

Chain UDP_OUTPUT_RULES (1 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 SSH_TRAFFIC  udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:22 dpt:22
       0        0 WWW_TRAFFIC  udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:80
       0        0 WWW_TRAFFIC  udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp dpt:443
       0        0 WWW_TRAFFIC  udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:80
       0        0 WWW_TRAFFIC  udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:443
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0           0.0.0.0/0           udp spt:53

Chain WWW_TRAFFIC (16 references)
    pkts      bytes target     prot opt in     out     source               destination
       0        0 ACCEPT     tcp  --  *      *       0.0.0.0/0           0.0.0.0/0
       0        0 ACCEPT     udp  --  *      *       0.0.0.0/0           0.0.0.0/0
```

Here is the scan after running the script ssh is disallowed on my system:

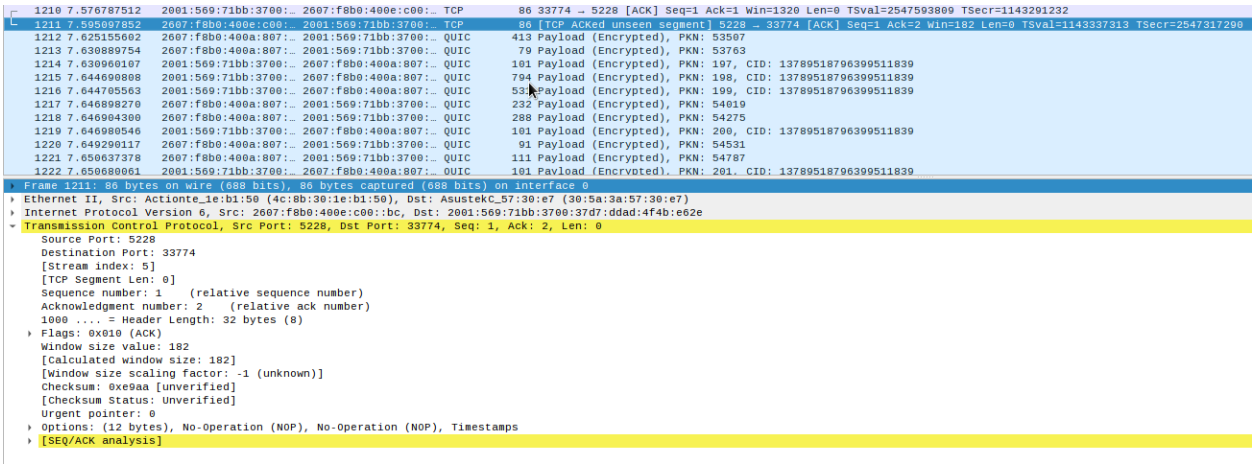
```
~ » nmap -p- --min-parallelism 100 -v 192.168.1.64

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-31 23:59 PST
Initiating Ping Scan at 23:59
Scanning 192.168.1.64 [2 ports]
Completed Ping Scan at 23:59, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:59
Completed Parallel DNS resolution of 1 host. at 23:59, 0.02s elapsed
Initiating Connect Scan at 23:59
Scanning 192.168.1.64 [65535 ports]
Connect Scan Timing: About 25.79% done; ETC: 00:01 (0:01:29 remaining)
Connect Scan Timing: About 58.28% done; ETC: 00:01 (0:00:44 remaining)
Completed Connect Scan at 00:01, 93.39s elapsed (65535 total ports)
Nmap scan report for 192.168.1.64
Host is up (0.0053s latency).
Not shown: 65533 filtered ports
PORT    STATE  SERVICE
80/tcp  closed http
443/tcp closed https

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 93.47 seconds
```

This shows that the config file allows WWW traffic through the machine. I was unable to test on a machine that has ssh correctly configured but the rules are configured inside the script.

Wireshark

The final test is done via Wireshark. Here I browse the web and see if there are any unexpected ports accessed.

In this capture we can see that a TCP packet #1211 from my machine to a server was dropped because it was not on port 443 or 80.



The capture file is included in the submitted package.