Cloud DNS Proxy Filter Concept

Wednesday, June 17, 2020 8:20 PM

Excellent visual of current Bailey Config -> openVPN+Proxy+PiHole DNS

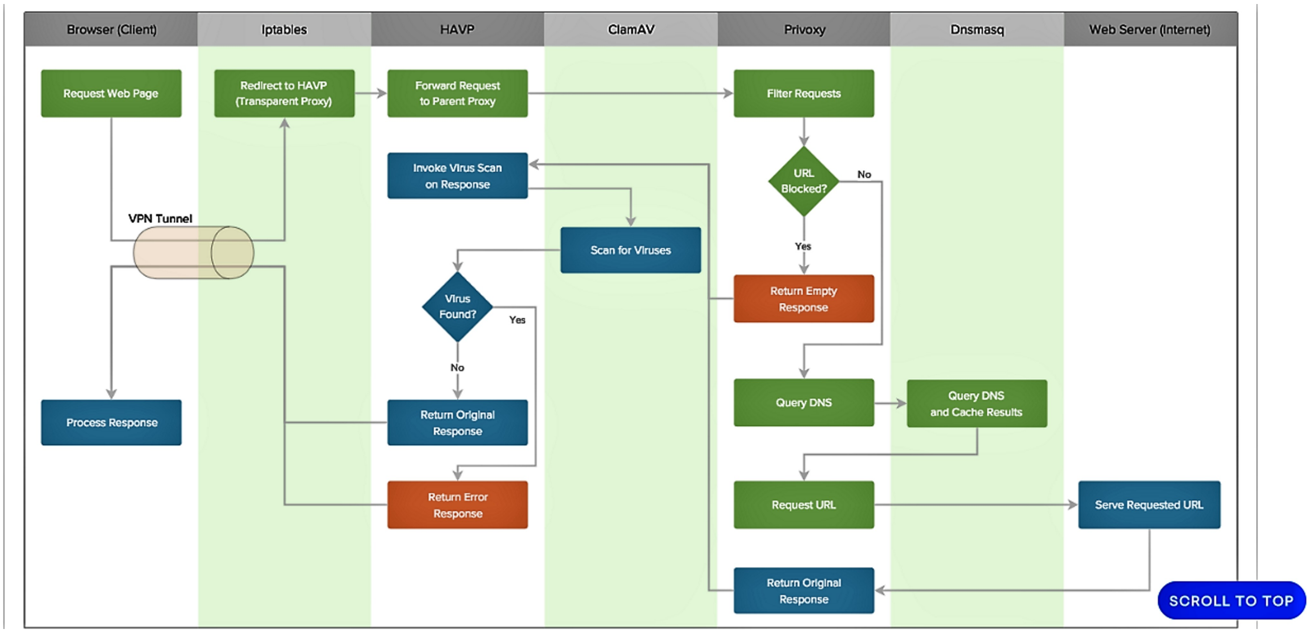
Flow Chart

1.) openVPN - creates a tunnel to accessing the web server so no one can access any of these services

2.) Iptables - firewall rules to protect from rogue incoming packets

3.) Proxy - will filter outgoing http(s) requests based on URL/host patterns

4.) Pi Hole (extension of DNSmasq) DNS - If request is allowed through proxy, Pi Hole DNS will reject domains on its blacklists and protect from incoming bullshit from the web



~~~~~~

The diagram in the link is an excellent visual of how VPN, proxy, and Pi Hole (DNS which includes dnsmasq) all work together

- `sudo vi /etc/privoxy/config`

  Scroll down to Section 6 and set the value of following parameter to 1. This will cause Privoxy to accept http requests redirected to it by firewall.

  ```
  accept-intercepted-requests 1
  ```

  Also remove 127.0.0.1 from privoxy listen address. Now, Privoxy will listen on all interfaces.

  ```
  listen-address    :8118
  ```

  Save and exit. Restart privoxy for changes to take effect.

- `sudo service privoxy restart`

  Now, you need to redirect http requests to Privoxy, remember it is set up to listen on port 8118.
~~~~~~

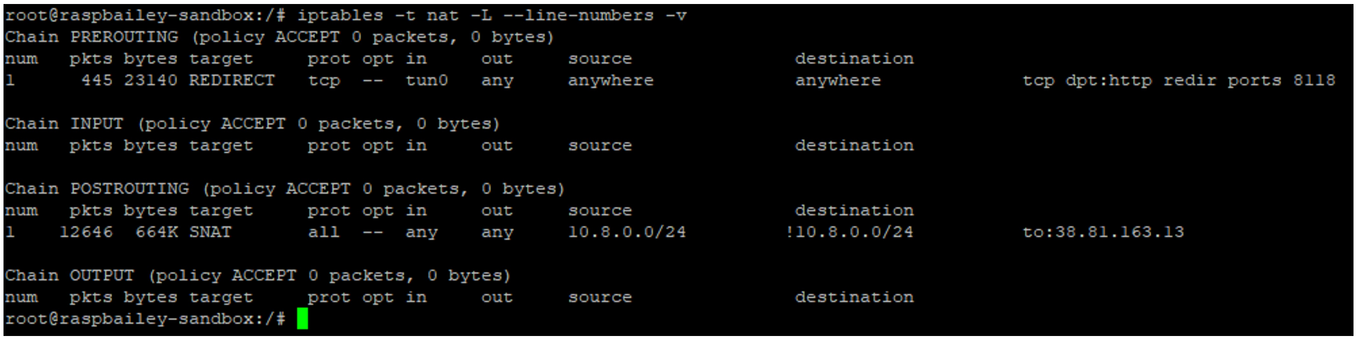
- `sudo vi /etc/ufw/before.rules`
  change the transparent proxy port from 8080 to 8118 it should look as below:

  ```
   -A PREROUTING -i tun+ -p tcp --dport 80 -j REDIRECT --to-port 8118
  ```

  From <https://www.digitalocean.com/community/tutorials/3-ways-to-securely-browse-the-internet-with-openvpn-on-debian-8>

~~~~~~
*** RED HIGHLIGHT = Slight adjustments needed since iptables has changed since this article was writing

- In 2020 Debian, no more before.rules file, iptables does everything.
    - "Iptables -t nat" let's you see nat rules
    - **Sudo iptables -t nat -A PREROUTING -i tun0 -p tcp --dport 80 -j REDIRECT --to-port 8118**

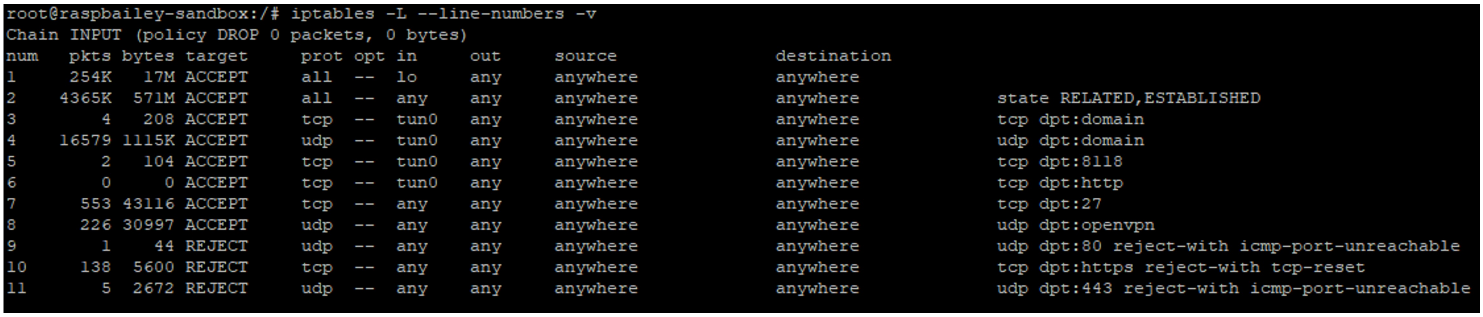


```
root@raspbailey-sandbox:/# iptables -t nat -L --line-numbers -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1      445 23140 REDIRECT   tcp  --  tun0   any     anywhere             anywhere             tcp dpt:http redir ports 8118

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

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1    12646  664K SNAT       all  --  any    any     10.8.0.0/24         !10.8.0.0/24          to:38.81.163.13

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
root@raspbailey-sandbox:/#
```

Result is above rules in NAT Prerouting policy, this will forward all incoming tcp traffic on port 80 from tun0 (VPN interface) and forward to privoxy configured default port 8118

- This will allow you to access the config page, therefore all traffic goes through proxy server
- The post routing rule already existed and is basically your IP masquerading rule. Our LAN that has been created for us to access Pi Hole via VPN interface tun0 all is masked on its way out to the public internet using our dedicated static v4 IP (IP of the cloud server) - SO DON'T TOUCH IT!

BUT WAIT! THIS WILL STILL NOT GET YOU SET UP COMPLETELY!



```
root@raspbailey-sandbox:/# iptables -L --line-numbers -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1     254K   17M ACCEPT     all  --  lo     any     anywhere             anywhere
2    4365K  571M ACCEPT     all  --  any    any     anywhere             anywhere             state RELATED,ESTABLISHED
3        4   208 ACCEPT     tcp  --  tun0   any     anywhere             anywhere             tcp dpt:domain
4    16579 1115K ACCEPT     udp  --  tun0   any     anywhere             anywhere             udp dpt:domain
5        2   104 ACCEPT     tcp  --  tun0   any     anywhere             anywhere             tcp dpt:8118
6        0     0 ACCEPT     tcp  --  tun0   any     anywhere             anywhere             tcp dpt:http
7      553 43116 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:27
8      226 30997 ACCEPT     udp  --  any    any     anywhere             anywhere             udp dpt:openvpn
9        1    44 REJECT     udp  --  any    any     anywhere             anywhere             udp dpt:80 reject-with icmp-port-unreachable
10     138  5600 REJECT     tcp  --  any    any     anywhere             anywhere             tcp dpt:https reject-with tcp-reset
11       5  2672 REJECT     udp  --  any    any     anywhere             anywhere             udp dpt:443 reject-with icmp-port-unreachable
```

The current iptables input chain policy is to drop packets for which no rule applies. There needs to be a specific firewall rule to allow traffic to reach tcp port 8118 that we defined in the nat table, otherwise the packets would get dropped and the privoxy web portal request would timeout

In iptables - ORDER MATTERS, so rather than append (-A) the new firewall rule to the INPUT chain, I want to insert (-I) the new rule up where the tun0 rules are, it just makes sense.

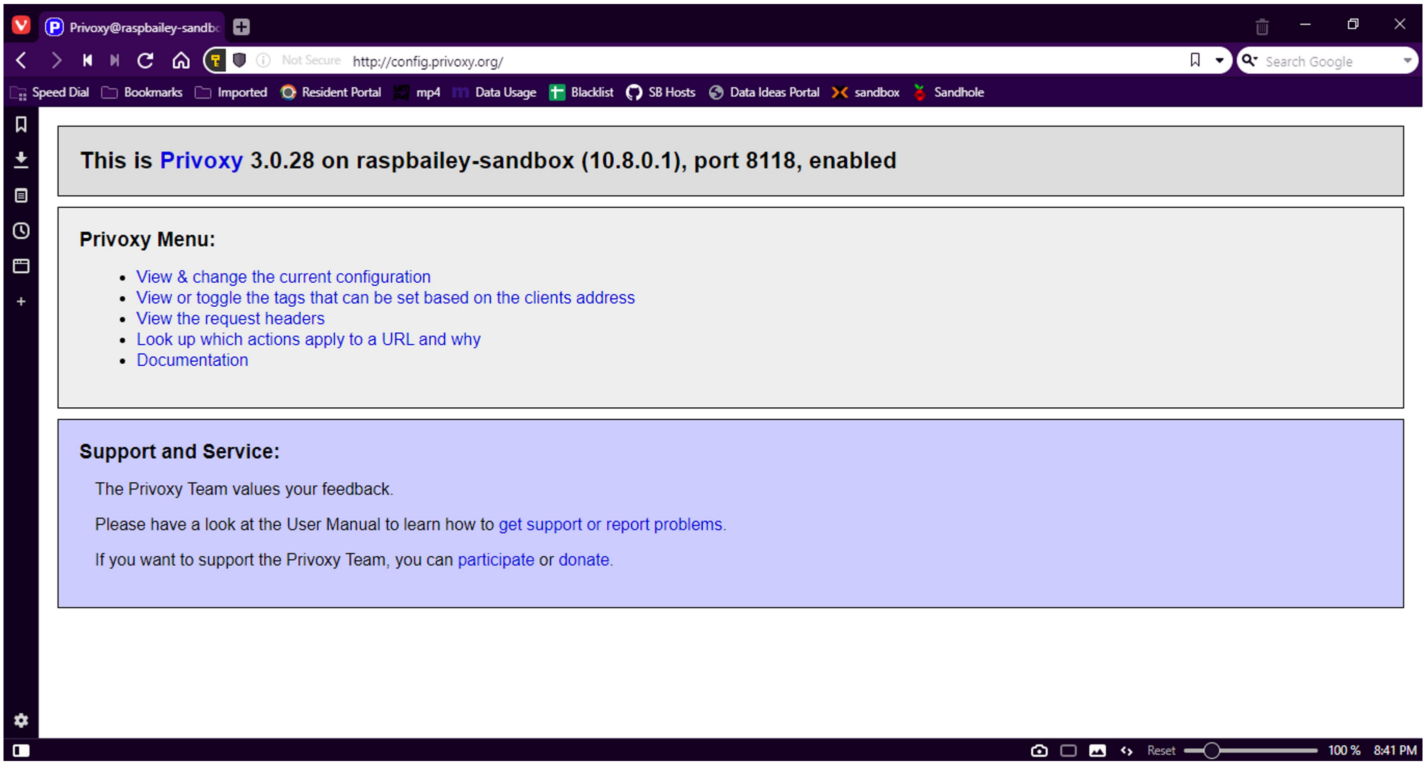
RUN:

iptables -I INPUT 5 -i tun0 -p tcp --destination-port 8118 -j ACCEPT

-I INPUT 5 means insert into position 5. use the --line-numbers argument to see the rules with line numbers like in the picture

Now go here and see if you can access:

http://config.privoxy.org/
~~~~~~



This is Privoxy 3.0.28 on raspbailey-sandbox (10.8.0.1), port 8118, enabled

Privoxy Menu:

- View & change the current configuration
- View or toggle the tags that can be set based on the clients address
- View the request headers
- Look up which actions apply to a URL and why
- Documentation

Support and Service:

The Privoxy Team values your feedback.

Please have a look at the User Manual to learn how to get support or report problems.

If you want to support the Privoxy Team, you can participate or donate.