Part 7 – Migrating SSH access

Migrate SSH to Wireguard interface

> Notice that I am running SSH on a non-standard port. The default is 22 and is often changed to reduce the number of bots spamming it on public servers. When running over VPN, it is safe to return back to the default 22 for simpler configuration.

Configure SSH on all interfaces

Currently my `iptables` firewall accepts SSH traffic on all interfaces on the correct port. SSH server configuration resides in `/etc/ssh/ssdh_config`. In this file `ListenAddress` is currently pointed to the public IP only. To bind both the public and Wireguard IP, replace the old value of `ListenAddress` with `0.0.0.0` and restart the service (your SSH connection won't be droped)

```
....
Port 7985
#AddressFamily any,inet
ListenAddress 0.0.0.0
#ListenAddress ::
....
```

```
$ sudo systemctl restart sshd
```

```
$ systemctl status sshd


sudo systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2021-09-21 20:02:01 CEST; 4s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 3503 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3504 (sshd)
```

```
      CPU: 55ms
   CGroup: /system.slice/ssh.service
           └─3504 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Sep 21 20:02:00 hostname systemd[1]: ssh.service: Succeeded.
Sep 21 20:02:00 hostname systemd[1]: Stopped OpenBSD Secure Shell server.
Sep 21 20:02:00 hostname systemd[1]: Starting OpenBSD Secure Shell server...
Sep 21 20:02:01 hostname sshd[3504]: Server listening on 0.0.0.0 port 7985.
Sep 21 20:02:01 hostname systemd[1]: Started OpenBSD Secure Shell server.
```

Connect SSH using the Wireguard server IP

Open another terminal windows on the machine that you use to connect to the server (Windows clients in my case):

```
> ssh username@10.20.20.1 -p 7985
```

It will give you the classic warning about unknown ECDSA fingerprint, type `yes` to procceed.

```
The authenticity of host '[10.20.20.1]:7985 ([10.20.20.1]:7985)' can't be established.
ECDSA key fingerprint is SHA256:&UIhigdanUYdfs/wF56atgafd851jL4w9uT564sg6133.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.20.20.1]:7985' (ECDSA) to the list of known hosts.
```

Voila...you should be in. Before binding SSH to the Wireguard interface only, **edit SSHd service to start after wg-quick.** Currently, SSHd doesn't care if wg-quick already started or not and might try to bind to an interface that doesn't exist yet. See the config now:

```
$ sudo systemctl cat sshd

[Unit]
Description=OpenBSD Secure Shell server
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
```

```
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
RuntimeDirectory=sshd
RuntimeDirectoryMode=0755

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Edit the config like this: (changes on the `Requires` and `After` line)

```
$ sudo systemctl edit --full sshd

[Unit]
Description=OpenBSD Secure Shell server
Documentation=man:sshd(8) man:sshd_config(5)
Requires=wg-quick@wg0.service
After=network.target auditd.service wg-quick@wg0.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
RuntimeDirectory=sshd
RuntimeDirectoryMode=0755

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Reload systemctl daemon to make the new configuration active:

```
$ sudo systemctl daemon-reload
```

I am now going to reboot the server for the last time I hope to see whether all services started up in a correct order and everything works. Surprise, surprise – it does.

Limit SSH to the Wireguard interface only

Go back to the sshd config file in `/etc/ssh/sshd_config` and set the `ListenAddress` field to `10.20.20.1`. Restart `sshd` to apply.

```
$ sudo systemctl restart sshd
```

Notice `sshd` binding to the Wireguard interface.

```
$ systemctl status sshd
...
Sep 21 22:53:20 hostname sshd[301]: Server listening on 10.20.20.1 port 7985.
...
```

Check `netstat` to confirm that everything is running on the Wireguard interface:

```
$sudo netstat -tulpn

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:5000          0.0.0.0:*               LISTEN      76/python3
tcp        0      0 10.20.20.1:7985         0.0.0.0:*               LISTEN      301/sshd: /usr/sbin
tcp        0      0 10.20.20.1:80           0.0.0.0:*               LISTEN      149/nginx: master p
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      71/systemd-resolved
tcp        0      0 127.0.0.1:8888          0.0.0.0:*               LISTEN      230/uwsgi
tcp        0      0 10.20.20.1:443          0.0.0.0:*               LISTEN      149/nginx: master p
tcp        0      0 127.0.0.1:4004          0.0.0.0:*               LISTEN      80/filtron
tcp        0      0 127.0.0.1:4005          0.0.0.0:*               LISTEN      80/filtron
udp        0      0 127.0.0.53:53           0.0.0.0:*                           71/systemd-resolved
udp        0      0 0.0.0.0:51895           0.0.0.0:*                           -
udp6       0      0 :::51895                :::*                                -
```

Adjust iptables

Make a last slight change to the iptables configuration in `/etc/iptables/rules.v4` by restricting SSH to the `wg0` interface

Replace the line

```
-A INPUT -p tcp -m tcp --dport 7985 -j ACCEPT
```

with

```
-A INPUT -i wg0 -p tcp -m tcp --dport 7985 -j ACCEPT
```

Apply `iptables` using `iptables-restore`

```
$ sudo iptables-restore /etc/iptables/rules.v4
```

> This should be it. Turned out to be way more complicated than expected, but that's exactly how it usually works, but unexpected things occur :)

Recap what we have managed to do today:

- Setup Wireguard with 3 peers (Linux server, Windows and Android client)
- Setup web services and SSH on correct interfaces
- Setup a DNS server for our clients
- Adjust network firewall and make it persistent

Doesn't seems like a lot, though when you look through this entire article, it's clear that it took some serious research and troubleshooting (just like any other IT thing, right?).

Revision #3
Created 22 September 2021 01:53:21 by Marek
Updated 22 September 2021 01:59:55 by Marek