Caddy Jail

We will ultimately change `PF` to direct all web traffic to this jail. This jail will run `caddy` as a reverse proxy for the other jails. Web request SSL terminations happen at the `caddy` web server, and the traffic is then passed transparently to the respective jails. A great benefit of `caddy` is the built-in Let's Encrypt feature for initial certs and renewals.

Preamble

The beginning steps are mostly the same across the jails. Before jumping in, if you haven't already, remember to run `custom_cshrc.sh caddy_jail`, and then probably/possibly run `bastille caddy_jail tzsetup` and choose your time zone. Actually, it would make the most sense for the reverse proxy to be on the host's time, which it should be already, so ignore that.

Next, we may update the jail. If you just created or updated your base jail, or if this is a thin jail, then there is actually no reason for this. But if you do need/want to do an update, refer to a prior page that talks about initial jail setup.

Setup Specific to this Jail

We install what we need from `pkg`.

```
bastille pkg caddy_jail install -y caddy vim-console curl
```

You should read the message spit out by `pkg` because it tells you all you need to know, pretty much. In particular, pay attention to the version of `caddy`. This write-up centers around `v1`. This write-up will not work well with `v2`.

Config for the Jail

We'll need to give `caddy` the ability to "authenticate" us with Let's Encrypt.

```
bastille sysrc caddy_jail caddy_cert_email="your.email@example.org"
```

And then we'll need the `Caddyfile`, which hopefully works how we think it will.

But wait! Save yourself some time and run this:

```
bastille cmd caddy_jail caddy -version
```

Your config/Caddyfile will be different depending on v1 or v2. The quarterly FreeBSD package is v1 right now (as of the time of this original write-up).

```
bastille console caddy_jail
```

```
cd /usr/local/
```

```
mkdir www && cd www
```

```
vim Caddyfile
```

> Depending on V1 or V2, mind the `Caddyfile` location. V2 moves the `Caddyfile` location from `/usr/local/www` to `/usr/local/etc/caddy/Caddyfile`, so be sure its location matches the location listed in the `rc` file (and is preferably in the standard location according to V1 or V2).

For `v1`:

```
mydomain.tld, www.mydomain.tld {
  proxy / 10.101.10.140:80 {
    transparent
  }
}

bookstack.mydomain.tld {
  proxy / 10.101.10.110:80 {
    transparent
  }
}
```

For `v2`:

```
mydomain.tld, www.mydomain.tld {
  reverse_proxy 10.101.10.140
}

bookstack.mydomain.tld {
```

```
  reverse_proxy 10.101.10.110
}
```

Then `exit` out of the jail's console. And then we enable `caddy` and start it (almost).

```
bastille sysrc caddy_jail caddy_enable="YES"
```

Grand Finale

Now we adjust `/etc/pf.conf` to forward `http` and `https` traffic to the `caddy` jail.

```
# the macro
caddy_ip = "10.101.10.100"

# and the port forward
rdr pass inet proto tcp from any to any port {80, 443} -> $caddy_ip
```

And then we test that the config doesn't have an errors, and then reload `PF`. (Reload w/ just `-f`.)

```
pfctl -vnf /etc/pf.conf
```

And now let's start `caddy` and hope that it grabs certs and starts serving our two existing jails.

```
bastille service caddy_jail caddy start
```

And either check the URL in your browser, or also check:

```
bastille service caddy_jail caddy status
```

That was easy.

Revision #3
Created 5 August 2020 03:46:24 by scoob
Updated 1 January 2021 20:39:59 by scoob