Security

- Setting up DNS over TLS on pfSense
- Acme Certificates with letsencrypt and cloudflare

Setting up DNS over TLS on pfSense

Source: https://medium.com/@davetempleton/setting-up-dns-over-tls-on-pfsense-bd96912c2416

DNS is a protocol woefully in need of confidentiality and integrity checks. The traditional service running over port 53 can be trivially eavesdropped upon to see what hosts you're visiting, and it's routinely intercepted and its responses altered for domains without DNSSEC (which is most domains). Many VPN clients do not by default route DNS queries over their tunnel, and thus so-called "DNS leaks" allow for hostname resolutions to disclose the nature of tunneled traffic.

There are two competing standards: DNS over TLS, and DNS over HTTPS. DNS over TLS is what pfSense most easily supports using its built-in resolver Unbound. Here's what I've done to set up DNS over TLS on pfSense 2.4.4p3.

Choosing your DNS servers

pfSense's implementation of DNS over TLS only allows connections to upstream resolvers on port 853. If you'd like to test if your resolver of choice allows connections on this port, you can run the following Nmap command:

```
nmap -Pn -sT -p 853 1.2.3.4
```

Where 1.2.3.4 is the server. You're looking for a state of "open" and not "closed" or "filtered." I use the following Google and Cloudflare servers which support DNS over TLS on port 853:

- 8.8.8.8
- 1.1.1.1
- 8.8.4.4
- 1.0.0.1

Feel free to add IPv6 addresses as well if that works on your ISP. You should edit your list of DNS servers in **System > General Setup** before continuing, as all listed servers *must* support DNS over TLS on port 853.

Setting up the DNS Resolver service

pfSense offers two competing DNS services: DNS Forwarder (dnsmasq) and DNS Resolver (Unbound). You *must* use the DNS Resolver, and the DNS Forwarder must be disabled. If you're using the DNS Forwarder currently, you must transition over to the DNS Resolver service; this would include manually copying over any Host Overrides and Domain Overrides if you have any. If you're excepting any domains from DNS rebinding protection, you'd use the following syntax in the DNS Resolver under Custom Options:

```
server:
private-domain: "example.com"
private-domain: "example.org"
```

with as many "private-domain" lines as you need. You must disable the DNS Forwarder service before you enable the DNS Resolver service, as only one service at a time can listen on port 53.

Here are some settings you should make sure you set in **Services > DNS Resolver**:

1. For Network Interfaces, shift-click to select multiple interfaces you'd like to offer DNS services on. Do not use the default of all interfaces, as you do not want to offer DNS services to the internet.
2. Outgoing Network Interfaces should be "WAN" or whatever interface(s) you use for your ISPs.
3. DNSSEC, DNS Query Forwarding, and "Use SSL/TLS for outgoing DNS Queries to Forwarding Servers" should all be enabled.
4. On the Advanced tab, I recommend enabling Prefetch Support, Prefetch DNS Key Support, and Harden DNSSEC Data.

Once enabled, make sure DNS services work on your LAN clients.

Blocking outbound DNS from LAN clients

It would be smart at this point to block outgoing connections on port 53, to make sure all services are using encrypted DNS. To do this, go to **Firewall > Rules > Floating** and click Add. Use the

following settings:

- Action: Reject (or Block)
- Quick: enabled
- Interface: WAN or whatever interface(s) you use for your ISPs
- Direction: out
- Address Family: IPv4 + IPv6
- Protocol: TCP/UDP
- Source: invert match, This Firewall
  (note: previous directions here said “any,” however that prevented the DNS Resolver service from restarting correctly)
- Destination: any
- Destination Port: 53

If you want to disable LAN clients from using DNS over TLS directly, perhaps so you can log all resolutions, you can block them from using port 853 as well. Unfortunately, a single floating rule wouldn’t work here, as blocking port 853 in this manner would also prevent the DNS Resolver service from working. You’d have to create a rule for each LAN/VLAN interface. These rule would look similar to the floating rule above, except:

- Destination: invert match, This Firewall (or “any” if you’ll never enable the DNS over TLS serving features of the DNS Resolver, which are off by default)
- Destination Port Range: 853

You’d be wise at this point to test resolution on LAN clients to make sure things work. You can also verify that LAN clients cannot access outbound DNS services by running the following Nmap command as root:

```
sudo nmap -Pn -sT -sU -p 53,853 8.8.8.8
```

No states should say “open” (a state of “open|filtered” is okay).

Redirecting outbound port 53 traffic

At it’s set up now, any client trying to reach an outside DNS server over port 53 will fail. Some software and devices might have been hard-coded to use these services for a variety of benign, lazy, or malicious reasons; if you’d like for them not to fail, these queries can be re-directed to your DNS Resolver service. Go to **Firewall > NAT > Port Forward** and click Add. Use the following settings:

- Interface: LAN (you'll need to make duplicate rules for each LAN/VLAN interface)
- Protocol: TCP/UDP
- Destination: invert match, This Firewall
- Destination port range: DNS
- Redirect target IP: 127.0.0.1
- Redirect target port: DNS
- NAT reflection: Disable

You can now test that LAN clients think this port is open on outside IPs:

```
sudo nmap -Pn -sT -sU -p 53 1.2.3.4
dig @1.2.3.4 example.com +short
```

Nmap should show a state of "open" for all non-local, non-bogon IPs, and Dig should quickly return an IP.

What protections have we gained?

What we've done here will greatly increase the reliability of your DNS resolutions by keeping them from being modified, and we've stopped the possibility of DNS leaks for any VPN clients.

What we haven't done is stop clients from using insecure protocols like HTTP. The browser extension/addon HTTPS Everywhere with its "Encrypt All Sites Eligible" mode enabled can greatly help here; I've been using this for years, and there are fewer sites without HTTPS support than you might expect. For sites that don't support HTTPS, you can individually allow them, or you can keep a Tor Browser window open and browse HTTP sites in there. In practice, the links I click that most commonly don't support HTTPS are the "Unsubscribe" links in commercial email.

I should also note here that, even with encrypted DNS and HTTPS web traffic, eavesdroppers can typically still know what hostname you're visiting because of Server Name Indication, which allows multiple websites to be hosted on the same web server. Even if your connection doesn't use SNI, if the destination server serves only one website, an eavesdropper would likely know what website you're loading (by knowing the destination IP). For privacy in all of these scenarios, you'd want to tunnel your connections past an eavesdropper using, for example, a VPN or Tor.

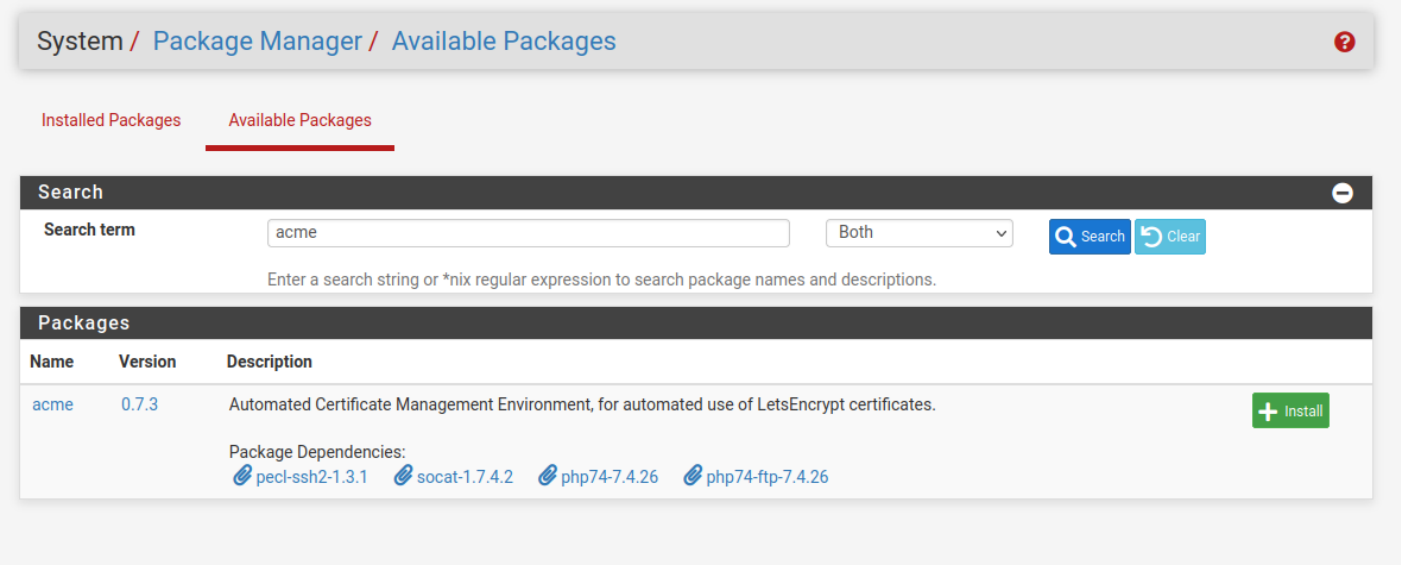
We also haven't stopped LAN clients from using outside DNS services that either run over non-standard ports, use DNS over HTTPS, or use other encrypted/obfuscated protocols.

Acme Certificates with letsencrypt and cloudflare

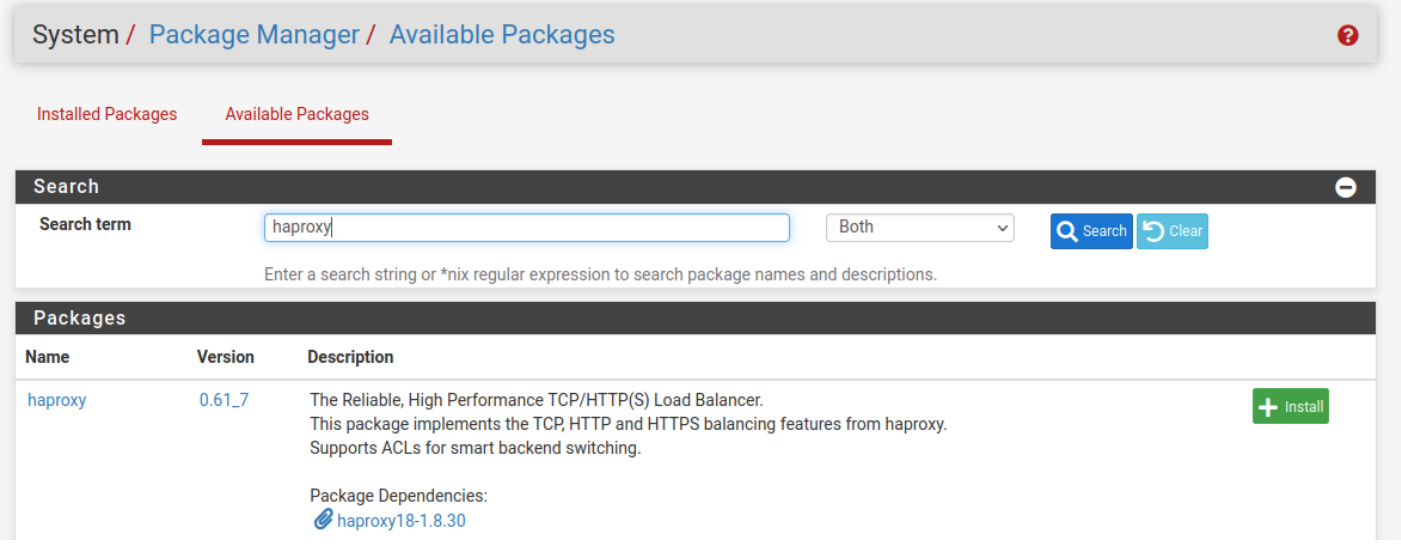
This is to install and configure Acme certs with Letsencrypt as well as deploying HA Proxy for access to our internal services and add SSL services, we're using cloudflare for our cert verification.

On PFsense navigate to system/package manager/available packages

Search for acme and click install then on confirm



do the same for haproxy and install then click on confirm.

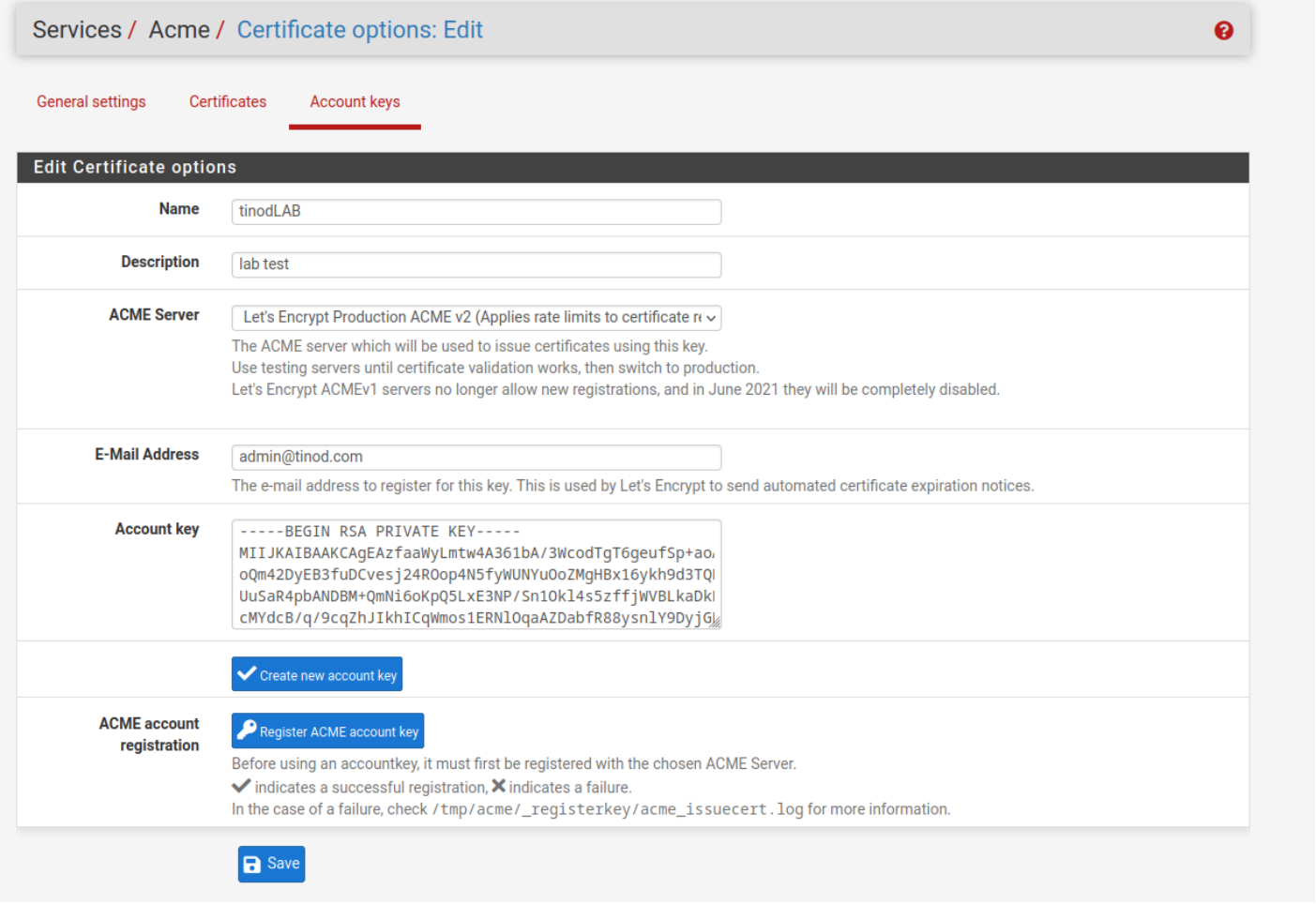


> if you are installing this via the wan interface access, make sure you go back and disable packet filter as doing the installation will re enable, go to the shell and type pfctl -d to disable and gain access via WAN interface

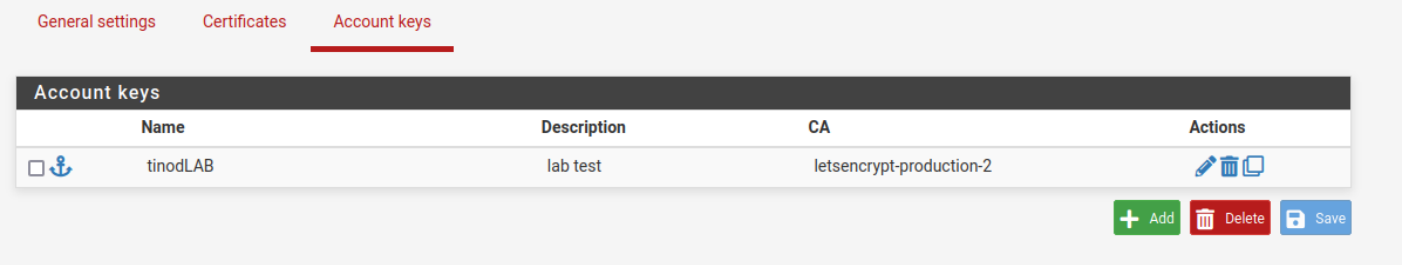
Navigate to Services/Acme

go to Account keys and click add, fill out the information below, for ACME Server you can use production or test, its the same

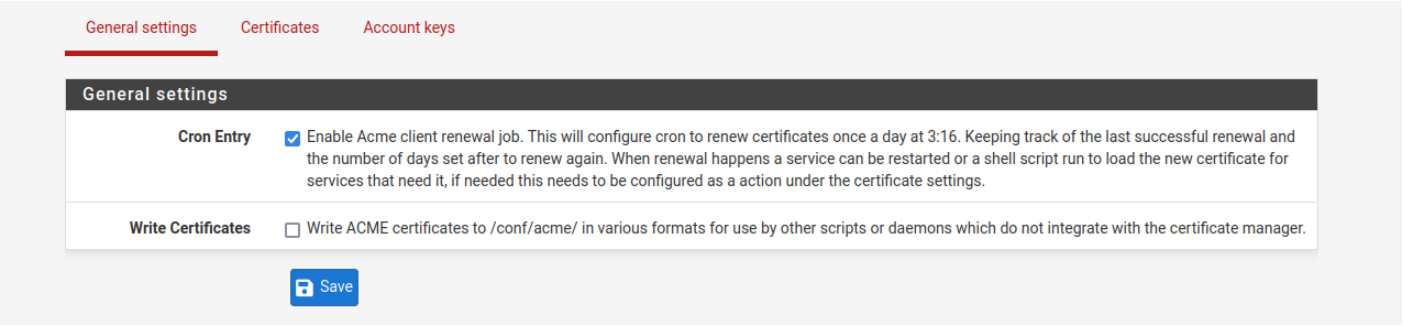
click on create new account key then register acme account key, after you are done click save



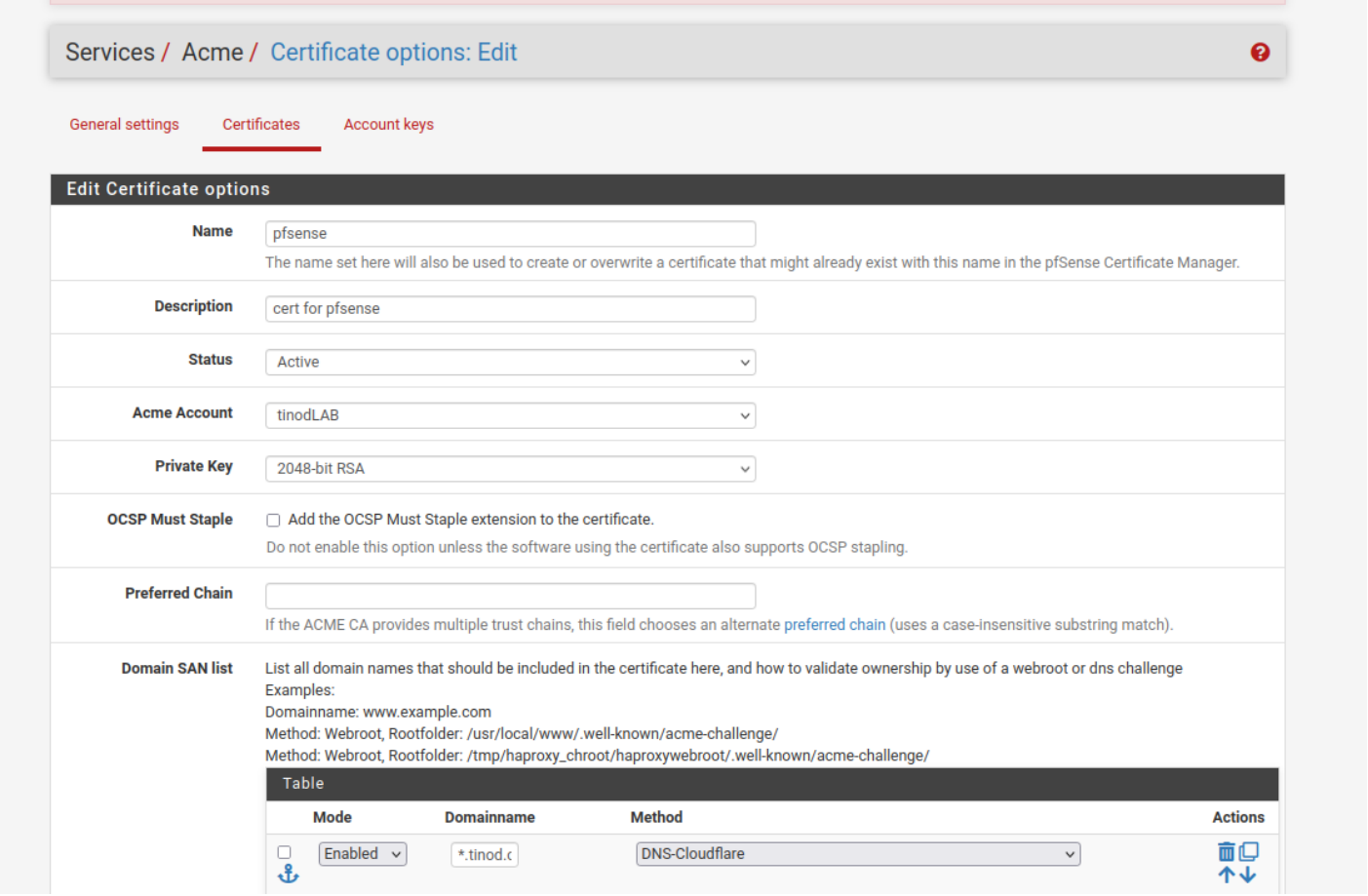
you have now your account key.



Navigate to General Settings and select Enable Acmee client renewal jojb to auto renew your certificates.

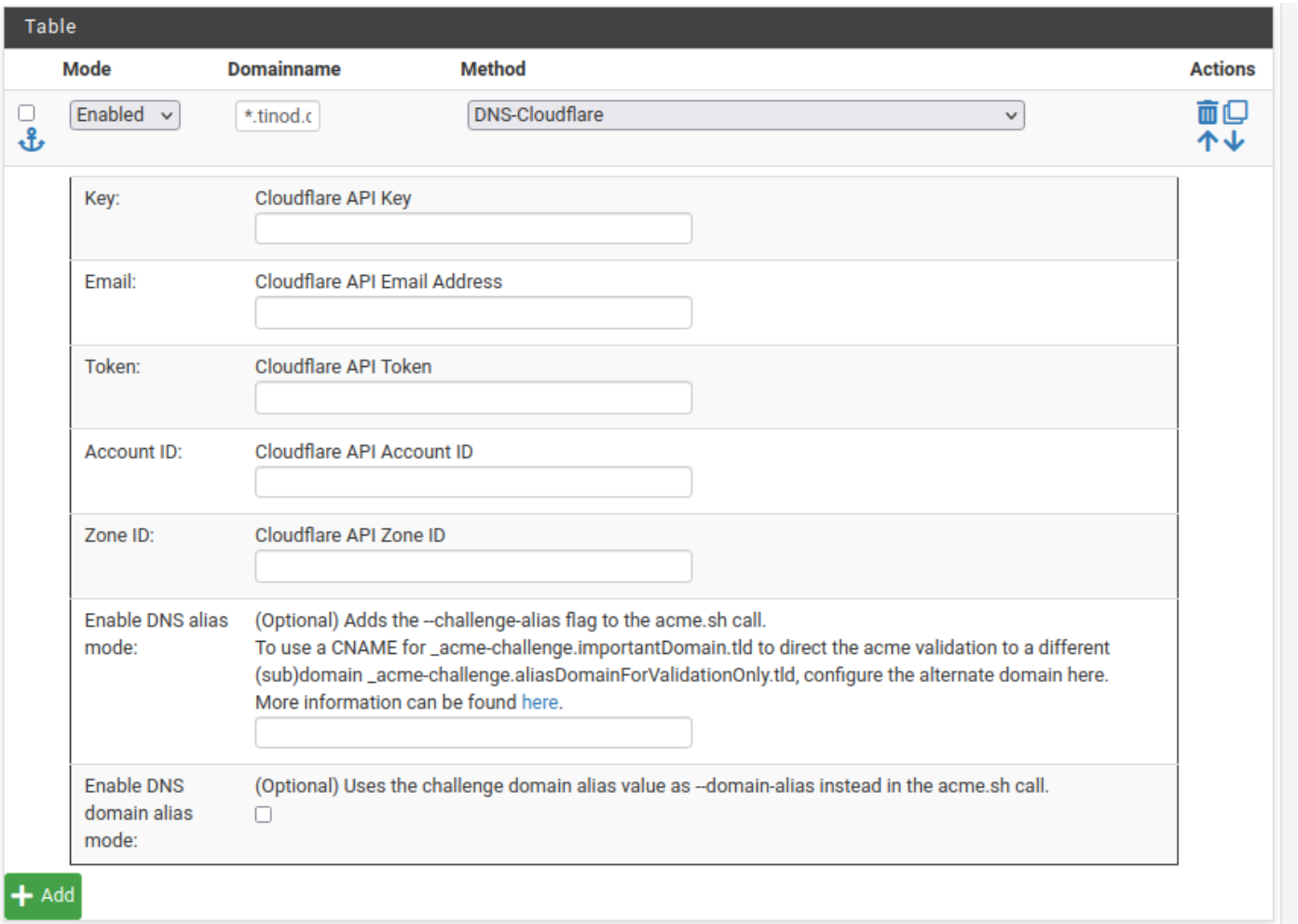


now you can create your first certificate, navigate to certificates and click add, we're using cloudflare



For this lab we're using a wildcard certificate, for our domain we will do *.tinod.com

you need to get your key, token, accound ID and Zone ID from cloudflare



click add and your cert will be ready and it will renew automatically