Aaron Toponce

{ 2016 05 20 }

Weechat Relay With Let's Encrypt Certificates

I've been on IRC for a long time. Not as long as some, granted, but likely longer than most. I've had my hand in a number of IRC clients, mostly terminal-based. Yup, I was (shortly) using the ircII client, then (also shortly) BitchX. Then I found irssi, and stuck with that for a *long* time. Search irssi help topics on this blog, and you'll see just how long. Then, after getting hired at XMission in January 2012, I switched full-time to WeeChat. I haven't looked back. This IRC client is amazing.

One of the outstanding features of WeeChat is the relay, effectively turning your IRC client into a bouncer. This feature isn't unique- it's in irssi also. However, the irssi proxy does not support SSL (2009). The WeeChat relay does. And with Let's Encrypt certificates freely available, this is the perfect opportunity to use TLS with a trusted certificate.

This post assumes that you are running WeeChat on a box that you can control the firewall to. In my case, I run WeeChat on an externally available SSH server behind tmux. With Let's Encrypt certificates, you will need to provide a FQDN for your Common Name (CN). This is all part of the standard certificate verification procedure. I purchased a domain that points to the IP of that server, and you will need to do the same.

The official Let's Encrypt "certbot" package used for creating Let's Encrypt certificates is already available in Debian unstable. A simple "apt install certbot" will get that up and running for you. Once installed, you will need to create your certificate.

```
$ certbot certonly --standalone -d weechat.example.com -m aaron.toponce@gmail.com
```

Per Let's Encrypt documentation, you needs ports 80 and 443 open to the world when creating and renewing your certificate. The execution will create four files:

```
# ls -l /etc/letsencrypt/
total 24
```

```
drwx------ 3 root root 4096 May 19 12:36 accounts/
drwx------ 3 root root 4096 May 19 12:39 archive/
drwxr-xr-x 2 root root 4096 May 19 12:39 csr/
drwx------ 2 root root 4096 May 19 12:39 keys/
drwx------ 3 root root 4096 May 19 12:39 live/
drwxr-xr-x 2 root root 4096 May 19 12:39 renewal/
# ls -l /etc/letsencrypt/live/weechat.example.com/
total 0
lrwxrwxrwx 1 root root 43 May 19 12:39 cert.pem -> ../../archive/weechat.example.com/cert1.pem
lrwxrwxrwx 1 root root 44 May 19 12:39 chain.pem -> ../../archive/weechat.example.com/chain1.pem
lrwxrwxrwx 1 root root 48 May 19 12:39 fullchain.pem -> ../../archive/weechat.example.com/fullchain1.pem
lrwxrwxrwx 1 root root 46 May 19 12:39 privkey.pem -> ../../archive/weechat.example.com/privkey1.pem
```

The "cert.pem" file is your public certificate for your CN. The "chain.pem" file in the Let's Encrypt intermediate certificate. The "fullchain.pem" file is the "cert.pem" and "chain.pem" files combined. Of course, the "privkey.pem" file is your private key. For the WeeChat relay, it needs the "privkey.pem" and "fullchain.pem" files combined into a single file.

Because the necessary directories under "/etc/letsencrypt/" are accessible only by the root user, you will need root access to copy the certificates out and make them available to WeeChat, which hopefully isn't running as root. Also, Let's Encrypt certificates need to be renewed no sooner than every 60 days and no later than every 90 days. So, not only will you want to automate renewing the certificate, but you'll probably want to automate moving it into the right directory when the renewal is complete.

As you can see from above, I setup my certificate on a Thursday at 12:39. So weekly, on Thursday, at 12:39, I'll check to see if the certificate needs to be nenewed. Because it won't renew any more frequently than every 60 days, but I have to have it renewed every 90 days, this gives be a 30-day window in which to get the certificate updated. So, I'll keep checking weekly. If a renewal isn't needed, the certbot(1) tool will gracefully exit. If a renewal is needed, the tool will update the certificate. Unfortunately, certbot(1) does not provide a useful exit code when renewals aren't needed, so rather than parsing text, I'll just copy the new certs into my WeeChat directory, regardless if they get updated or not.

So, in my root's crontab, I have the following:

```
39 12 * * 4 /usr/local/sbin/renew.sh
```

Where the contents of "/usr/local/sbin/renew.sh" are:

```
#!/bin/bash

certbot renew -q
cat /etc/letsencrypt/live/weechat.example.com/privkey.pem \
    /etc/letsencrypt/live/weechat.example.com/fullchain.pem > \
    ~aaron/.weechat/ssl/relay.pem
chown aaron.aaron ~aaron/.weechat/ssl/relay.pem
```

Now the only thing left to do is setup the relay itself in WeeChat. So, from within the client:

```
/relay sslcertkey
/relay add ssl.weechat 8443
```

You will need port 8443 open in your firewall, of course.

That's it. I have had some problems with certificate caching in WeechatAndroid it seems. So far, I have had to manually restart the relay in WeeChat, and flush the cache in WeechatAndroid and restart it to get the new certificate (I was previously using a self-signed certificate). Hopefully, this can also be automated, so I don't have to manually keep restarting the relay in WeeChat and flushing the cache in WeechatAndroid.

Regardless, this is how you use Let's Encrypt certificates with WeeChat SSL relay. Hopefully this is beneficial to someone.

Posted by Aaron Toponce on Friday, May 20, 2016, at 9:40 am. Filed under Debian, General, Linux, Scripting. Follow any responses to this post with its comments RSS feed. You can post a comment or trackback from your blog. For IM, Email or Microblogs, here is the Shortlink.

{ 6 } Comments

1. Chris Hills | October 28, 2016 at 3:46 am | Permalink

   How do you signal to weechat to reload the ssl cert automatically?