Will Let's Encrypt work for me? (Multiple servers serving one domain)

Issuance Tech

2gkc27 2015-12-13 03:41:25 UTC #1

Hello! I have a domain,let's say example.com, that has multiple A records.

```
;; QUESTION SECTION:
;example.com.         IN    A

;; ANSWER SECTION:
example.com.    300    IN    A    198.51.100.42
example.com.    300    IN    A    203.0.113.13
example.com.    300    IN    A    192.0.2.37
```

Would it be possible for each of these server to use Let's Encrypt to get a certificate issued for example.com without issue? Or would multiple A records mess things up?

Osiris 2015-12-13 23:24:49 UTC #2

As far as I know, the ACME server will choose one IP address at random for its challenge, so each of the servers will need to be able to answer to it. I'm guessing they are three different servers? 😛

---

jhass 2015-12-13 23:33:56 UTC #3

Not trivially until the `dns-01` challenge is fully implemented and deployed to production. Currently you have to make sure the `http-01` challenge is distributed to all servers to guarantee it's solved.

In either case you will either need to distribute the generated certificate among the servers or take extra precaution to not hit the 5 certificates per domain per 7 days rate limit.

Osiris 2015-12-13 23:39:24 UTC #4

> jhass:
>
> Currently you have to make sure the `http-01` challenge is distributed to all servers to guarantee it's solved.

~~Shouldn't be *that* hard?~~

Surely, Let's Encrypt is meant to be automated, but if **@2gkc27** is willing to do stuff manually, it wouldn't be that hard to put the challenge in the right spot on the servers? We're talking about three IP's here, so there's probably just three servers.. A terminal with three tabs and three SSH clients, type `echo -n "" >` into all three, start the manual authentication process, Ctrl-C and -V the challenge contents from the Let's Encrypt terminal tab between the two " in all three tabs, Ctrl-C/V the filename after the three >, press enter three times and continue with your manual authentication.. All done in less than ten seconds.

---

jhass 2015-12-13 23:40:13 UTC #5

I never claimed it's hard? That it's easy to realize doesn't make it trivial though.

Osiris 2015-12-13 23:42:19 UTC #6

Ah OK, my bad.. 😄 Guess I'll have to polish my meaning/interpretation of some (English) words 😉

---

xfgpaywx 2015-12-14 00:13:15 UTC #7

(Hi, I'm **@2gkc27** - permanently locked myself out of that account)

Thanks both of you for replying! They are three separate servers and unfortunately I only control one of them.. hmm. Too bad it doesn't connect back to the originating server if it's in an A record.

Maybe I'll need to coordinate some sort of synchronization of .well-known or something.. Hm.

---

souki 2015-12-14 00:25:41 UTC #8

It is actualy really simple if you can change webserver config. I have this rule in site config (nginx):

```
location ~ /\.well-known/acme-challenge/ {
proxy_pass http://letsencrypt.example.org:8081;
}
```

Letsencrypt is running on machine **http://letsencrypt.example.org1** with port mapped from 8081 to 80.
I can now simply verify domain running on different machines with single instance.

---

xfgpaywx 2015-12-14 01:00:13 UTC #9

Thanks for the suggestion!

I was hoping to automate it a bit if at all possible (so each server could auto-renew). Maybe some sort of shared filesystem for the .well-known directory)?

---

souki 2015-12-14 13:08:12 UTC #10

My solution might be little more practical then shared files because you don't need to share anything.

You can have this 3 lines of code in config of every site. Than you can once a day run docker container somewhere and it will renewe/issue certificates. Then you need just some mechanism to deploy new certificates to each server (we are using Puppet)

---

frazzledjazz 2015-12-22 20:19:20 UTC #11

why dont you register it for one domain and let the loadbalance take care of the sync between the ips/filesystem? doent make sence to do this any other way. that is how you implemented, correct? any of the three ips should reply to the domain request. They should all be setup the same so a write to filesystem of one server is same across all three. Pointless waste of linux to do that otherwise.I dont see why youd have an issue.

---

m-maarten 2016-04-21 10:12:00 UTC #12

Does anybody knows had this can work with Apache proxy?

---

jmorahan 2016-04-21 12:17:21 UTC #13

It would be something along the lines of

```
ProxyPass "/.well-known/acme-challenge/" "http://letsencrypt.example.org:8081"
```

Michael_MCP 2016-05-23 00:40:45 UTC #14

Have i understood this right?
The port mapping be done on a firewall for only one of the servers, and the other servers wouldn't accept port 8081 ?
you then add that to the webserver/nginx config (i don't understand why)? the port would have changed so it would simply accept it as a 80 request?

or is this script on the other servers to proxy and repoint the request to the correct server?

Michael_MCP 2016-05-23 00:45:40 UTC #15

A solution i found for my 3 server setup was to stop the web service on the two secondary servers, leaving the webserver only running on the primary letsencrypt server. This forces all traffic to the letsencrypt server.

1) Set a cronjob for the nginx or apache service to stop on the secondary servers
2) Set a cronjob on the letsencrypt server for the renewal a minuet after the other servers stop
3) Give the renewal 3 minuets then sync the new certificates (scp or rsync) and bring the two secondary servers back online.

and whala! you have three servers each with the up to date cert. The only issue is the site will be slow/vulnerable for the window of letsencrypt only server.

---

archon810 2016-12-30 09:54:40 UTC #16

@Michael_MCP That's a terrible solution, especially if you end up with a traffic spike and your single server gets overpowered.

The proposed solution was much better - it only proxies (transparent to the remote party) the location that it hits to verify the domain and nothing else.

---

allo 2016-12-30 21:41:02 UTC #17

What you can try is using the manual mode (try it one time to see how it works) and on the step where you need to place the files in the webroot, you use some script, which places the files on all relevant servers. I think the "manual" part can be automated by own scripts then.

The proxy pass solution sounds good as well. You may want to use TLS on the proxy connection.

---

danday74 2017-01-12 11:13:59 UTC #18

i can confirm this works.

location ~ /.well-known/acme-challenge/ {
proxy_pass **http://ctrl.mydomain.com:80**;
}

using nginx i added this location to ALL server blocks.

You then run lets encrypt on the machine ctrl.mydomain.com (this machine typically is the controller machine, and is not serving web stuff - its pure purpose from a web POV is to handle incoming cert requests - if you don't know what a controller machine is then read up on ansible)

To make it work I had to use the webroot plugin for Let's Encrypt. I could not get standalone mode to work.

my A records look like ..

www01.mydomain.com points to 1.2.3.4
www02.mydomain.com points to 2.3.4.5
ctrl.mydomain.com points to 3.4.5.6
mydomain.com points to 1,2,3,4 and 2,3,4,5 (multiple A records)
www.mydomain.com is an alias (cname) for mydomain.com

NGINX runs on www01 and www02 on port 80 to load balance requests (e.g. www01 load balances between www01 and www02, www02 ALSO load balances between www01 and www02)

the above lets encrypt location block is added to NGINX running on both www01 and www02 for all NGINX server blocks

now run lets encrypt in webroot mode (you will need to standup a web server on your controller machine) and request a single certificate for **www01.mydomain.com www02.mydomain.com** mydomain.com **www.mydomain.com**

when you run this command on your controller machine (ctrl.mydomain.com) it will fireoff a request to each of the 4 domains in return. Every single request will be proxied back to ctrl.mydomain.com via NGINX

bosh!

2 tips

1 - to use webroot mode you will need to have a basic web server running on ctrl.domain.com which can serve content from a specified directory

2 - do not use standalone mode, i could not get it to work

3 - this solution sits very nicely if you are using ansible, since the certs will live on the controller machine and can be copied across to all slave machines with a single command

---