

Forum:[Hack Forums](#)

Thread: [\[Socialites\] Run your own Caching non-Logging/Censoring, DNSSEC/DNSCrypt DNS Resolver](#)

SUI-3183372 - April 16, 2017, 1:10 am

Download: <https://mega.nz/#!os4TzYgC!mmTbPpOBOAbkud-OXDguvelo7hH5SHDBprFh2ohA5jY>

DNSCrypt server Docker image

Run your own caching, non-censoring, non-logging, DNSSEC-capable, [DNSCrypt](#)-enabled DNS resolver virtually anywhere!

If you are already familiar with Docker, it shouldn't take more than 5 minutes to get your resolver up and running.

Installation

Think about a name. This is going to be part of your DNSCrypt provider name. If you are planning to make your resolver publicly accessible, this name will be public. It has to look like a domain name (example.com), but it doesn't have to be a registered domain.

Let's pick example.com here.

Download, create and initialize the container, once and for all:

```
Code $ docker run --name=dnsencrypt-server -p 443:443/udp -p 443:443/tcp --net=host \
jedisct1/unbound-dnscrypt-server init -N example.com
```

This will only accept connections via DNSCrypt on the standard port (443).

Code --net=host

provides the best network performance, but may have to be removed on some shared containers hosting services.

Now, to start the whole stack:

```
Code $ docker start dnscrypt-server
```

Done.

To check that your DNSCrypt-enabled DNS resolver is accessible, run the DNSCrypt client proxy on another host:

```
Code # dnscrypt-proxy \
--provider-key=<provider key, as displayed when the container was initialized> \
--resolver-address=<dnscrypt resolver public IP address> \
--provider-name=2.dnscrypt-cert.example.com
```

And try using 127.0.0.1 as a DNS resolver.

Note that the actual provider name for DNSCrypt is 2.dnscrypt-cert.example.com, not just example.com as initially entered. The full name has to start with 2.dnscrypt-cert. for the client and

the server to use the same version of the protocol.

Let the world know about your server

Is your brand new DNS resolver publicly accessible?

Fork the [dnscrypt-proxy repository](#), edit the [dnscrypt.csv](#) file to add your resolver's informations, and submit a pull request to have it included in the list of public DNSCrypt resolvers!

Customizing Unbound

To add new configuration to Unbound, add files to the `/opt/unbound/etc/unbound/zones` directory. All files ending in `.conf` will be processed. In this manner, you can add any directives to the server: section of the Unbound configuration.

Serve custom DNS records on a local network

While Unbound is not a full authoritative name server, it supports resolving custom entries in a way that is serviceable on a small, private LAN. You can use unbound to resolve private hostnames such as `my-computer.example.com` within your LAN.

To support such custom entries using this image, first map a volume to the zones directory. Add this to your docker run line:

Code `-v /myconfig/zones:/opt/unbound/etc/unbound/zones`

The whole command to create and initialize a container would look something like this:

Code `$ docker run --name=dnscrypt-server \`
`-v /myconfig/zones:/opt/unbound/etc/unbound/zones \`
`-p 443:443/udp -p 443:443/tcp --net=host \`
`jedisct1/unbound-dnscrypt-server init -N example.com`

Create a new `.conf` file:

Code `$ touch /myconfig/zones/example.conf`

Now, add one or more unbound directives to the file, such as:

Code `local-zone: "example.com." static`
`local-data: "my-computer.example.com. IN A 10.0.0.1"`
`local-data: "other-computer.example.com. IN A 10.0.0.2"`

Troubleshooting

If Unbound doesn't like one of the newly added directives, it will probably not respond over the network. In that case, here are some commands to work out what is wrong:

Code `$ docker logs dnscrypt-server`

`$ docker exec dnscrypt-server /opt/unbound/sbin/unbound-checkconf`

DetailsCaching resolver: [Unbound](#), with DNSSEC, prefetching, and no logs. The number of threads and memory usage are automatically adjusted. Latest stable version, compiled from source. qname minimisation is enabled.

[LibreSSL](#) - Latest stable version, compiled from source.

[libsodium](#) - Latest stable version, minimal build compiled from source.

[dnscrypt-wrapper](#) - Latest stable version, compiled from source.

[dnscrypt-proxy](#) - Latest stable version, compiled from source.

Keys and certificates are automatically rotated every 12 hour.

Download: <https://mega.nz/#!os4TzYgC!mmTbPpOBOAbkud-OXDguvelo7hH5SHDBprFh2ohA5jY>

Source: <https://github.com/jedisct1/dnscrypt-server-docker>

Malicious Intent - April 16, 2017, 1:19 am

Thanks for this tut. Ill be testing this later on today.

SUI-3183372 - April 16, 2017, 2:30 pm

(Apr 16, 2017 01:19 AM)Malicious Intent Wrote: [\[CITE\]](#) Thanks for this tut. Ill be testing this later on today.

It's a great alternative to using default ISP defined DNS servers.

There are already some existing DNSSEC / DNSCrypt alternatives, but this is nice if you want to control your own.

Malicious Intent - April 16, 2017, 2:46 pm

(Apr 16, 2017 02:30 PM)Previously Known As.... Wrote: [\[CITE\]](#) It's a great alternative to using default ISP defined DNS servers.

There are already some existing DNSSEC / DNSCrypt alternatives, but this is nice if you want to control your own.

It certainly is. Running one instance might be risky without ddos protection but its a great option for some uses.

SUI-3183372 - April 16, 2017, 2:49 pm

(Apr 16, 2017 02:46 PM)Malicious Intent Wrote: [\[CITE\]](#) It certainly is. Running one instance might be risky without ddos protection but its a great option for some uses.

If it's used for private usage, you don't run a very high risk of it being attacked as no one should know it exists. However, you're right if there is an unrelated attack against your server provider, an attacker could unknowingly kill your DNS resolver.

Malicious Intent - April 16, 2017, 2:53 pm

(Apr 16, 2017 02:49 PM)Previously Known As.... Wrote: [\[CITE\]](#) If it's used for private usage, you don't run a very high risk of it being attacked as no one should know it exists. However, you're right if there is an unrelated attack against your server provider, an attacker could unknowingly kill your DNS resolver.

Yep. Depends on the usage ofc. Running two of those would be really cheap anyway. Just host in separate data centers and should be all good.

Orgy - April 19, 2017, 3:57 pm

dang this is a great tutorial

Evanescent - April 20, 2017, 3:23 am

Looks interesting, subscribed to the thread to come back.

B?ker - April 24, 2017, 6:37 am

Thank you for this man! This is awesome

Malicious Intent - April 24, 2017, 7:07 am

I'm surprised this didnt get more attention. Its a really good write up.

Mr. Robot - April 24, 2017, 7:08 am

(Apr 24, 2017 07:07 AM)Malicious Intent Wrote: [\[CITE\]](#) I'm surprised this didnt get more attention. Its a really good write up.

Thanks for keeping it alive, it slipped past me.

SUI-3183372 - *July 17, 2017, 7:36 am*

Thread has been updated with new information.

SUI-3183372 - *October 17, 2017, 12:43 am*

I've begun working with the Socialites usergroup to better maintain my privacy, security, and anonymity threads. This will help me push out thread updates sooner and include more content. This thread has been updated with new information.

New support for kubernetes

Kubernetes

Kubernetes configurations are located in the "kube" directory. Currently these assume a persistent disk named "dnscrypt-keys" on GCE. You will need to adjust the volumes definition on other platforms. Once that is setup, you can have a dnscrypt server up in minutes. Create a static IP on GCE. This will be used for the LoadBalancer.

Edit "kube/dnscrypt-init-job.yml" and change "example.com" to your desired hostname.

Edit "kube/dnscrypt-srv.yml" and change "loadBalancerIP" to your static IP.

Run "kubectl create -f kube/dnscrypt-init-job.yml" to setup your keys.

Run "kubectl create -f kube/dnscrypt-deployment.yml" to deploy the dnscrypt server.

Run "kubectl create -f kube/dnscrypt-srv.yml" to expose your server to the world.

To get your public key just view the logs for the "dnscrypt-init" job. The public IP for your server is merely the dnscrypt service address.