Manual for

Installation and Configuration of DoH (DNS over HTTPS) and DoT (DNS over TLS)

December 2020

Centre of Excellence in DNS SECURITY

System Requirements

1) OS: Ubuntu 20.04 LTS
2) Internet connection.
3) A valid IP address.
4) Hostname: doh.local
5) Software Packages Required: bind v9.16.1+, dnsdist v1.4+

Manual at Glance:

I. Prepare the system for installation Update the OS

```
#sudo su
#apt update
#apt upgrade
#apt autoremove
```

II. Install and Verify Bind Installation

1) Install Bind 9

```
# apt install bind9
```

2) Verify installation

```
# nslookup www.cdac.in localhost
```

```
root@ubuntu:/home/anoopmis# nslookup www.cdac.in localhost
Server:         localhost
Address:        127.0.0.1#53

Non-authoritative answer:
www.cdac.in     canonical name = cdac.in.
Name:   cdac.in
Address: 196.1.113.45
Name:   cdac.in
Address: 2405:8a00:6029::45
```

III. Install and Verify dnsdist Installation

dnsdist is a highly DNS-, DoS- and abuse-aware loadbalancer. Its goal in life is to route traffic to the best server, delivering top performance to legitimate users while shunting or blocking abusive traffic. For more information visit: https://dnsdist.org/.

1) Install dnsdist

```
# apt install dnsdist
```

2) Verify dnsdist installation

```
# dnsdist --version
```

```
root@ubuntu:/home/anoopmis# dnsdist --version
dnsdist 1.4.0 (Lua 5.2.4)
Enabled features: cdb dns-over-tls(gnutls openssl) dns-over-https(DOH) dnscrypt ebpf
fstrm ipcipher libsodium lmdb protobuf re2 recvmmsg/sendmmsg snmp systemd
```

The version must be 1.4.0+.

IV. Generate TLS Certificate

If the system has a valid public domain name, then a free TLS certificate can be obtained from Let's Encrypt. The steps for obtaining the certificate are listed at: https://certbot.eff.org/

1) To generate a self-signed certificate, run the following command:

```
# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout doh.local.key -out doh.local.crt
```

```
root@ubuntu:/home/anoopmis# openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout doh.local.key -out doh.local.crt
Generating a RSA private key
..........+++++
.....................................................................................
..........+++++
writing new private key to 'doh.local.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:Karnataka
Locality Name (eg, city) []:Bangalore
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CDAC
Organizational Unit Name (eg, section) []:CSG
Common Name (e.g. server FQDN or YOUR name) []:doh.local
Email Address []:anoop@cdac.in
```

2) Copy the key and certificate to respective TLS directories.

```
# cp doh.local.key /etc/ssl/private/

# cp doh.local.crt /etc/ssl/certs/
```

V. Configure dnsdist for DoH and DoT

1) Open dnsdist configuration file:

```
# nano /etc/dnsdist/dnsdist.conf
```

Append following line to change the listening port of dnsdist. By default, it listens on port 53 which conflicts with bind listening port. Now it will listen on all interfaces on port 5300.

```
addLocal('0.0.0.0:5300', {doTCP=true, reusePort=true, tcpFastOpenSize=0})
```

*Optionally for IPv6, add following line

```
addLocal('[::]:5300', {doTCP=true, reusePort=true, tcpFastOpenSize=0})
```

Add Permissive ACL

```
addACL('0.0.0.0/0')
```

*Optionally for IPv6, add following line

```
addACL('::/0')
```

Add Local Recursive Resolver to which DNS queries will be forwarded. The IP can be the local IP, or localhost IP or the Public IP.

```
newServer({address="127.0.0.1",qps=1, name="resolver1"})
```

*Optionally, multiple recursive DNS servers can be added by repeating the line above and changing the IP address and name.
Add TLS resolver as follow. The certificate and key also need to be specified.

```
addTLSLocal('0.0.0.0','/etc/ssl/certs/doh.local.crt, '/etc/ssl/private/doh.local.key')
```

*Optionally for IPv6, add following line

```
addTLSLocal('[::]','/etc/ssl/certs/doh.local.crt, '/etc/ssl/private/doh.local.key')
```

Add DoH resolver as follow. The DoH will be listening to all interfaces (0.0.0.0) on port 443 and can be queried on the context "/dns-query".

```
addDOHLocal("0.0.0.0:443", "/etc/ssl/certs/ doh.local.crt", "/etc/ssl/private/
doh.local.key", "/dns-query", { doTCP=true, reusePort=true, tcpFastOpenSize=0 })
```

Now you file should look like this:

```
-- dnsdist configuration file, an example can be found in /usr/share/doc/dnsdist/examples/

-- disable security status polling via DNS
setSecurityPollSuffix("")

addLocal('0.0.0.0:5300', {doTCP=true, reusePort=true, tcpFastOpenSize=0})

addACL('0.0.0.0/0')

newServer({address="127.0.0.1",qps=1, name="resolver1"})

addTLSLocal('0.0.0.0','/etc/ssl/certs/doh.local.crt', '/etc/ssl/private/doh.local.key')

addDOHLocal("0.0.0.0:443", "/etc/ssl/certs/doh.local.crt", "/etc/ssl/private/doh.local.key", "/dns-query", {
doTCP=true, reusePort=true, tcpFastOpenSize=0 })
```

2) Save the dnsdist configuration file and start & check the dnsdist service. The status should show active (running).

```
# systemctl start dnsdist

# systemctl status dnsdist
```

```
● dnsdist.service - DNS Loadbalancer
     Loaded: loaded (/lib/systemd/system/dnsdist.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2020-08-20 17:03:53 IST; 5s ago
       Docs: man:dnsdist(1)
             https://dnsdist.org
    Process: 18457 ExecStartPre=/usr/bin/dnsdist --check-config (code=exited, status=0/SUCCESS)
   Main PID: 18474 (dnsdist)
      Tasks: 13 (limit: 8192)
     Memory: 24.1M
     CGroup: /system.slice/dnsdist.service
             └─18474 /usr/bin/dnsdist --supervised --disable-syslog -u _dnsdist -g _dnsdist

Aug 20 17:03:52 doh.local dnsdist[18457]: Configuration '/etc/dnsdist/dnsdist.conf' OK!
Aug 20 17:03:52 doh.local dnsdist[18474]: Added downstream server 127.0.0.1:53
Aug 20 17:03:52 doh.local dnsdist[18474]: Listening on 0.0.0.0:5300
Aug 20 17:03:52 doh.local dnsdist[18474]: Listening on 0.0.0.0:853 for TLS
Aug 20 17:03:52 doh.local dnsdist[18474]: Listening on 0.0.0.0:443 for DoH
```

3) To check if the DoH service is running, we can use curl. *[Install curl, if not present]*

```
# curl –doh-url https://doh.local/dns-query www.iiref.in
```

```
root@doh:~# curl --doh-url https://doh.local/dns-query
https://www.iiref.in


<!DOCTYPE html>
<html>
<head>

<link type="text/css" rel="stylesheet" href="css/bootst
rap.css" />
<link type="text/css" rel="stylesheet" href="css/custom
.css" />
<link type="text/css" rel="stylesheet" href="css/footer
.css" />
<link type="text/css" rel="stylesheet" href="css/text_h
over.css"/>
```

VI. Install dnslookup tool for verifying DoH and DoT working

[Only if the DoH server has a valid domain name and TLS certificate]

Remember dnslookup will only work if the DoH/DoT server has a valid domain name and TLS certificate. With a self-signed certificate, it will throw error. Skip to Step 8 to check DoT configured with a self-signed certificate.

```
root@dns:/home/anoopmis# dnslookup www.cdac.in https://doh.local/dns-query
dnslookup 1.3.0-4282
2020/08/20 16:16:58 Cannot make the DNS request: couldn't initialize HTTP client or
 transport, cause: couldn't do a POST request to 'https://doh.local:443/dns-query',
 cause: Get "https://doh.local:443/dns-query?dns=_rkBAAABAAAAAAAACGlwdjRvbmx5BGFycG
EAAAEAAQ": x509: certificate signed by unknown authority
```

1) Download and install dnslookup using SNAP

```
# snap install dnslookup
```

2) Add SNAP to your path (temporarily)

```
# export PATH=$PATH:/snap/bin
```

3) Query DoH and DoT [Here, a public DoH/DoT is being used.]

```
# dnslookup www.cdac.in https://doh.iiref.in/dns-query

# dnslookup www.cdac.in tls://doh.iiref.in
```

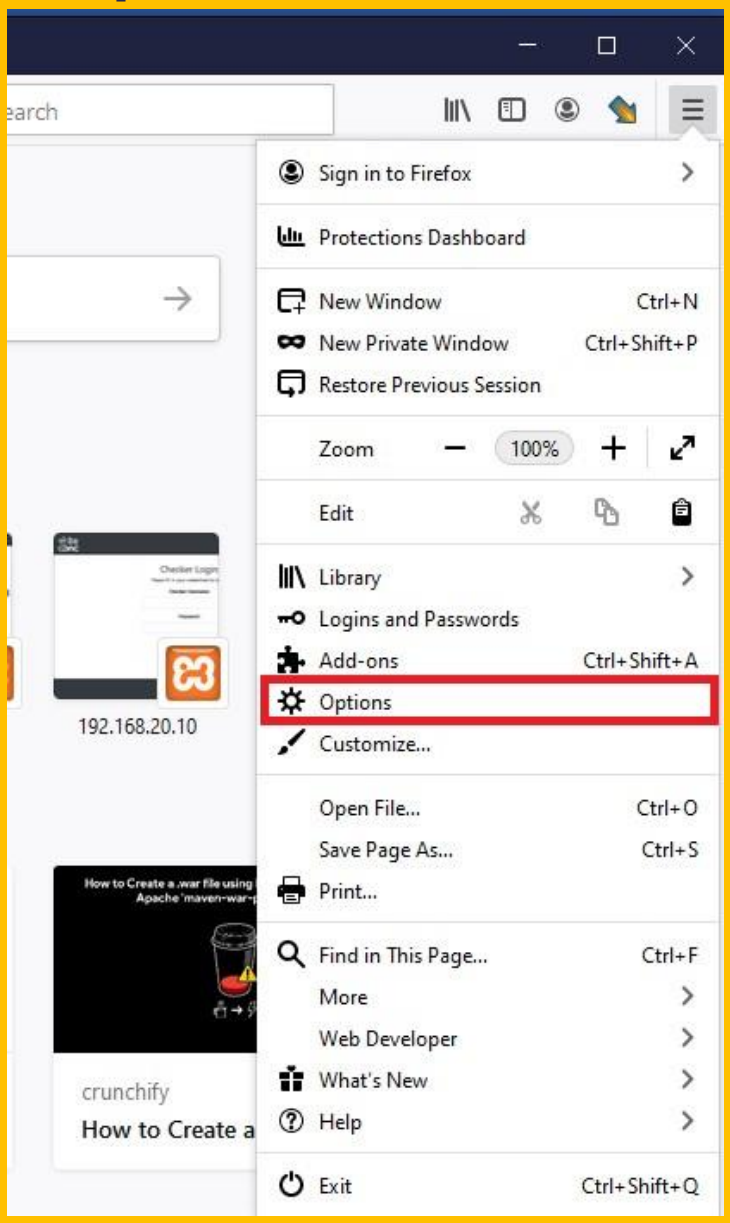
```
root@ubuntu:/home/anoopmis# dnslookup www.cdac.in tls://doh.iiref.in
dnslookup 1.3.0-4282
dnslookup result:
;; opcode: QUERY, status: NOERROR, id: 23753
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.cdac.in.    IN       A

;; ANSWER SECTION:
www.cdac.in.     841      IN       CNAME    cdac.in.
cdac.in.         841      IN       A        196.1.113.45

root@ubuntu:/home/anoopmis# dnslookup www.cdac.in https://doh.iiref.in/dns-query
dnslookup 1.3.0-4282
dnslookup result:
;; opcode: QUERY, status: NOERROR, id: 965
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.cdac.in.    IN       A

;; ANSWER SECTION:
www.cdac.in.     836      IN       CNAME    cdac.in.
cdac.in.         836      IN       A        196.1.113.45
```

VII. Enabling DoH in FireFox

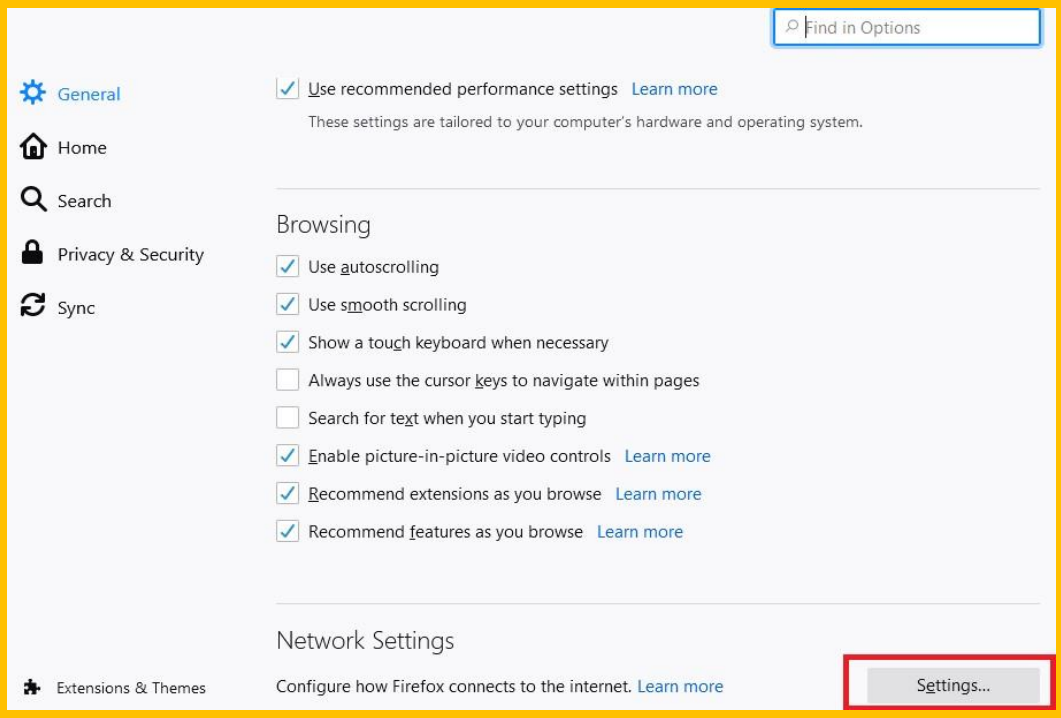
VII. Enabling DoH in Firefox *[Only if the DoH server has a valid domain name and TLS certificate]*

The Firefox browser should be updated to the latest version (currently 73.0+)

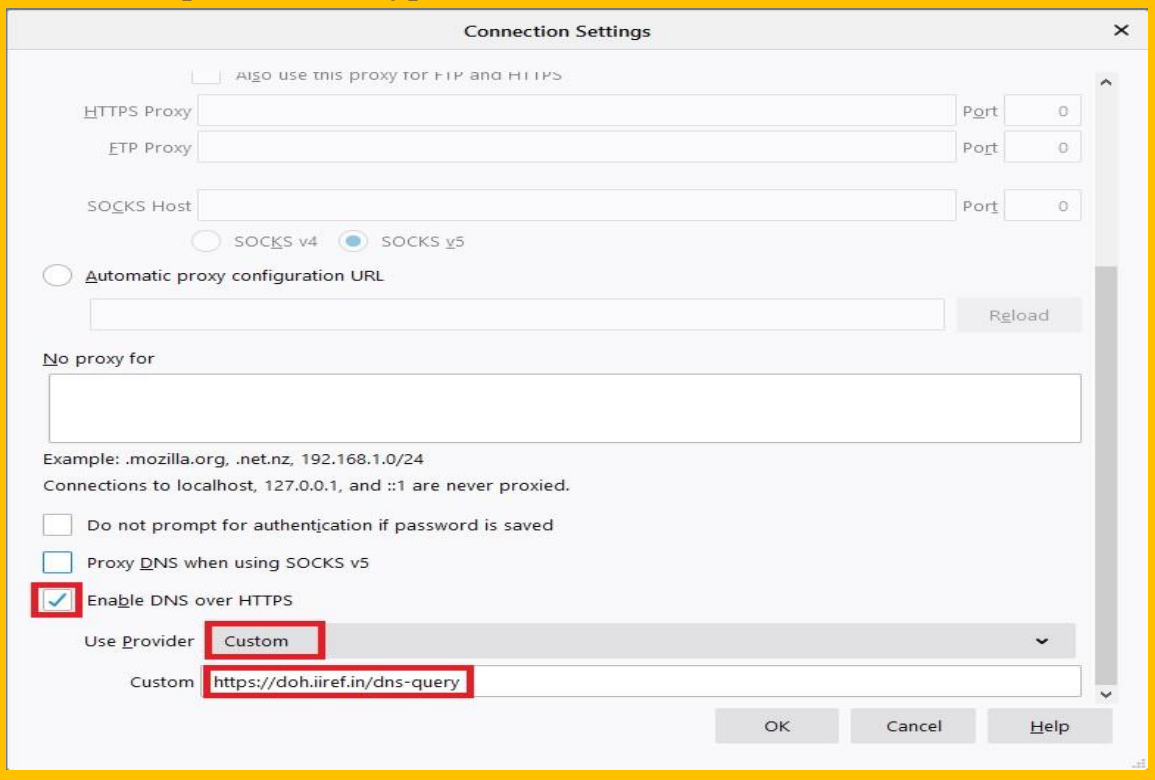
1) **To enable DoH, click the three horizontal bars in the top-right corner of Firefox and then select the "Options" button.**



2) **Search for Network Settings (usually at the bottom of the page) and click on the button.**



3) **Scroll to the bottom of the wizard, click on Enable DNS over HTTPS checkbox, select custom provider and type the URL of the DoH server.**



VIII. Testing DoT using Stubby by enabling SPKI *[This is optional for DoT with a valid TLS certificate.]*

Following steps should be performed on a different client system which will query the DoT server using stubby. We have used another Ubuntu system.

1) Download and install stubby.

```
# apt install stubby
```

2) Generate SPKI for DoT server.

```
# echo | openssl s_client -connect <your_IP_address>:853 2>/dev/null | openssl x509 pubkey -noout | openssl pkey -pubin -outform der | openssl dgst -sha256 -binary | openssl enc -base64
```

```
root@dns:/home/anoopmis# echo | openssl s_client -connect 192.168.10.15:853
2>/dev/null | openssl x509 -pubkey -noout | openssl pkey -pubin -outform der
 | openssl dgst -sha256 -binary | openssl enc -base64
ovx9fT8Z59wBO5vOv5iVIKDmgRUiiYvS+V/akpft57E=
```

3) Configure upstream recursive server on stubby.

```
# nano /etc/stubby/stubby.yml
```

Go to "upstream_recursive_servers:" section and disable default servers beneath by adding a '#' before each line.

Now add the DoT server as follow:

```
- address_data: <your_IP_server>     tls_auth_name: "doh.local"     tls_pubkey_pinset:
- digest: "sha256" value: ovx9fT8Z59wBO5vOv5iVIKDmgRUiiYvS+V/akpft57E=
```

Here value contains the SPKI output generated in step 2.

Your stubby config file should look like this now:

```
upstream_recursive_servers:

# IPv4 addresses
# DoH Server
  - address_data: 192.168.10.15
    tls_auth_name: "doh.local"
    tls_pubkey_pinset:
      - digest: "sha256"
        value: ovx9fT8Z59wBO5vOv5iVIKDmgRUiiYvS+V/akpft57E=
```

4) Restart stubby service

```
# systemctl restart stubby
```

5) Update your resolver to point to localhost (stubby)

```
# nano /etc/resolv.conf

nameserver 127.0.0.1
```

6) Query using nslookup

```
# nslookup www.coednssecurity.in
```

```
root@ubuntu:/home/ant# nslookup www.coednssecurity.in
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   www.coednssecurity.in
Address: 220.156.189.66
Name:   www.coednssecurity.in
Address: 2404:4100:0:3000::189:66
```

7) Check log on DoT server (If logging is enabled)

```
# cat /var/cache/bind/querylog | grep coedns
```

```
root@dns:/home/anoopmis# cat /var/cache/bind/querylog | grep coedns
20-Aug-2020 17:12:25.788 client @0x7f02c002b030 127.0.0.1#43132 (www.coednssecurit
y.in): query: www.coednssecurity.in IN A +E(0)T (127.0.0.1) [ECS 0.0.0.0/0/0]
20-Aug-2020 17:12:26.932 client @0x7f02c002b030 127.0.0.1#43132 (www.coednssecurit
y.in): query: www.coednssecurity.in IN AAAA +E(0)T (127.0.0.1) [ECS 0.0.0.0/0/0]
```

IX. Enable logging in Bind 9

1) Open bind configuration file

```
# nano /etc/bind/named.conf
```

Append the following line

```
include "/etc/bind/named.conf.logging";
```

2) Open named options file and enable querylog

```
# nano /etc/bind/named.conf.options

querylog yes;
```

```
root@dns:/home/anoopmis# cat /etc/bind/named.conf.options
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        // forwarders {
        //      0.0.0.0;
        // };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        querylog yes;

        listen-on-v6 { any; };
};
```

3) Create logging configuration file and add the following

```
# nano /etc/bind/named.conf.logging
logging {
      channel querylog {
            file "/var/cache/bind/querylog";
            severity debug 3;
            print-time yes;     // Print timestamp on logs
      };
      category queries { querylog; };
};
```

4) Create querylog file and change ownership

```
# touch /var/cache/bind/querylog
# chown bind.bind /var/cache/bind/querylog
```

5) Restart Bind

```
# systemctl restart bind
```


Acknowledgements:

We express our sincere thanks to Internet Governance Division of Ministry of Electronics & Information Technology (MeitY) and National Internet Exchange of India (NIXI).