C&C environment installation and configuration for DoH communication with DNSCat2

December 1, 2023

Contents

1 Minimum system requirements OS **2**

1.1 Debian 11 2

1.2 Windows 10 2

2 VirtualBox Network setting **2**

3 Server (Debian 11) **3**

3.1 Installation configuration 3

3.2 Basic configuration 3

3.2.1 Power setting (Mandatory) 3

3.2.2 User configuration 4

3.2.3 Install packages 4

3.2.4 Modify firewall rules 4

3.2.5 Allow SSH connection as Root 4

3.3 DNSCat2 server installation 4

3.3.1 Install Ruby and RubyGems 4

3.3.2 DNSCat2 server 4

3.3.3 Checking the reception of DNS requests on ”without desktop” mode . . . 5

4 Local DoH resolver **5**

4.1 Installation configuration 5

4.2 Basic configuration 5

4.2.1 Power setting (Mandatory) 5

4.2.2 User configuration 5

4.2.3 Install packages 5

4.2.4 Modify firewall rules 6

4.3 Bind9 installation 6

4.4 DNSDist 1.8 installation 6

4.4.1 Configure the dnsdist server 7

5 Client DNSCat2 (Debian 11) **8**

5.1 Installation configuration 8

5.2 Basic configuration 8

5.2.1 Power setting (Mandatory) 8

5.2.2 User configuration 8

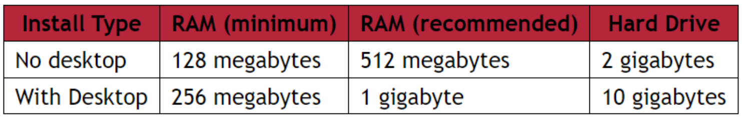
5.2.3 Install packages 8

5.3 DNSCat2 client installation . . . 8
5.4 DoH-proxy installation . . . 9

6 Controller **9**
6.1 Installation configuration . . . 9
6.2 Basic configuration . . . 10
6.2.1 Power setting (Mandatory) . . . 10
6.2.2 User configuration . . . 10
6.2.3 Install packages . . . 10
6.2.4 Install python modules . . . 10
6.3 Download automation code . . . 10

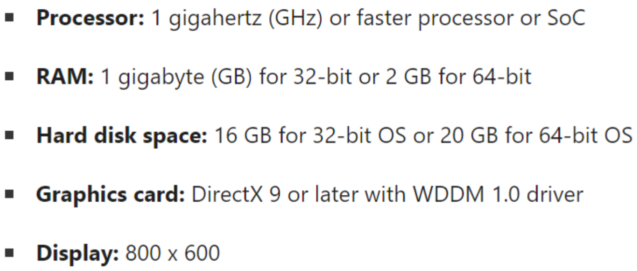
1 Minimum system requirements OS

1.1 Debian 11



Install Type	RAM (minimum)	RAM (recommended)	Hard Drive
No desktop	128 megabytes	512 megabytes	2 gigabytes
With Desktop	256 megabytes	1 gigabyte	10 gigabytes

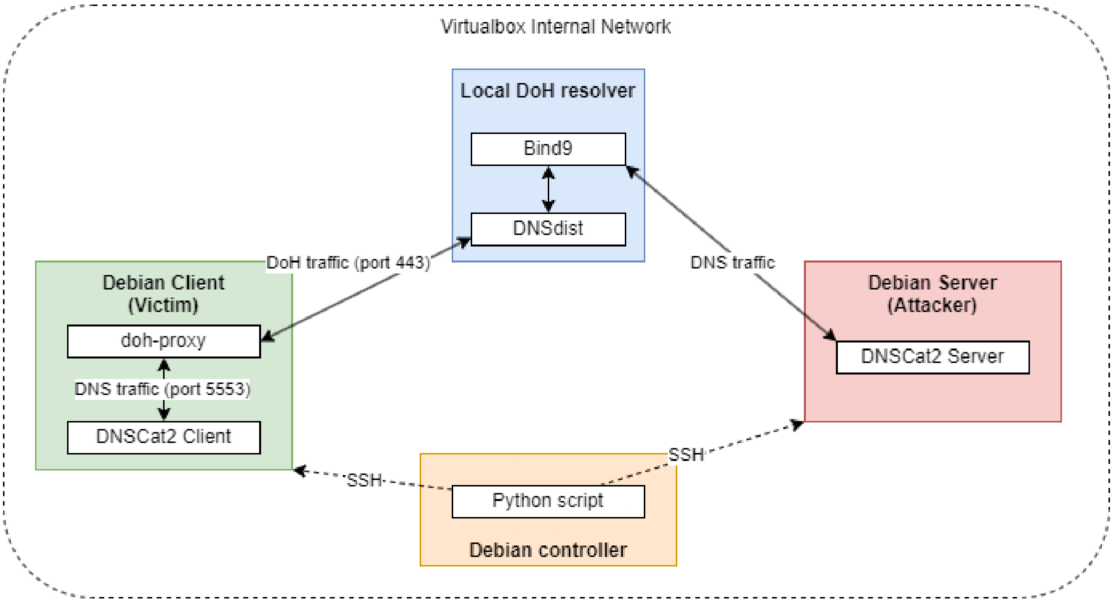
1.2 Windows 10



- **Processor:** 1 gigahertz (GHz) or faster processor or SoC
- **RAM:** 1 gigabyte (GB) for 32-bit or 2 GB for 64-bit
- **Hard disk space:** 16 GB for 32-bit OS or 20 GB for 64-bit OS
- **Graphics card:** DirectX 9 or later with WDDM 1.0 driver
- **Display:** 800 x 600

2 VirtualBox Network setting

Create a LAN network which is named by default ”NATNetwork” and set all the created machine to this network during the configuration before installation.



3 Server (Debian 11)

3.1 Installation configuration

- **CPU thread**: 2
- **RAM**: 1024 MB (512 MB if without desktop)
- **Storage**: 30 GB
- **Network**: NAT Network
- **Root Password**: root
- **Non-admin username**: attacker
- **Non-admin password**: notroot
- **Hostname**: dohserver
- **Enable SSH server**

3.2 Basic configuration

3.2.1 Power setting (Mandatory)

In Settings > Power > Power Saving, put:

- Blank Screen: Never
- Automatic Suspend:
  - On Battery Power: OFF
  - Plugged In: OFF

3.2.2 User configuration

Add the user to sudoers (if without desktop version):

```
$ su -
# apt install sudo
# usermod -a -G sudo attacker
```

Restart the machine. (if without desktop, use: sudo shutdown -h now)

3.2.3 Install packages

```
$ sudo apt install git make gcc dtach wireshark tcpdump ufw psmisc avahi-daemon
```

3.2.4 Modify firewall rules

```
$ sudo ufw enable
$ sudo ufw allow 53/udp
$ sudo ufw allow 22/tcp
$ sudo ufw reload
$ sudo ufw status
```

3.2.5 Allow SSH connection as Root

Edit the following file: $ sudo nano /etc/ssh/sshd_config

- Comment the following line: PermitRootLogin prohibit-password
- Add below: PermitRootLogin yes
- Restart the SSH service: $ sudo service ssh restart

3.3 DNSCat2 server installation

3.3.1 Install Ruby and RubyGems

```
$ sudo apt install ruby-dev
```

3.3.2 DNSCat2 server

```
$ git clone https://github.com/iagox86/dnscat2.git
$ cd dnscat2/server/
$ sudo apt-get install build-essential
$ sudo gem install salsa20 -v '0.1.1'
$ sudo gem install bundler
$ sudo bundle install
```

Verify that the installation was well done with: $ sudo ruby ./dnscat2.rb

Possible errors:

- https://github.com/iagox86/dnscat2/issues/173
- https://www.logsec.cloud/posts/dnscat2-demo-on-aws/

3.3.3 Checking the reception of DNS requests on ”without desktop” mode

To check that DNS requests are being received correctly on a system without a graphical user interface, use the following tcpdump command: $ sudo tcpdump -i any -vv port 53

4 Local DoH resolver

4.1 Installation configuration

- **CPU thread**: 2
- **RAM**: 1024 MB (512 MB if without desktop)
- **Storage**: 10 GB
- **Network**: NAT Network
- **Root Password**: root
- **Non-admin username**: resolver
- **Non-admin password**: notroot
- **Hostname**: doh.local
- **Enable SSH server**

4.2 Basic configuration

4.2.1 Power setting (Mandatory)

In Settings > Power > Power Saving, put:

- Blank Screen: Never
- Automatic Suspend:
  - On Battery Power: OFF
  - Plugged In: OFF

4.2.2 User configuration

Add the user to sudoers (if without desktop version):

```
$ su -
# apt install sudo
# usermod -a -G sudo resolver
```

Restart the machine. (if without desktop, use: sudo shutdown -h now)

4.2.3 Install packages

```
$ sudo apt install build-essential curl ufw wireshark avahi-daemon
```

4.2.4 Modify firewall rules

```
$ sudo ufw enable
$ sudo ufw allow 53/udp
$ sudo ufw allow 443/tcp
$ sudo ufw allow 22/tcp
$ sudo ufw reload
$ sudo ufw status
```

4.3 Bind9 installation

```
$ sudo apt install bind9 bind9utils bind9-doc
$ sudo systemctl status named
$ sudo nano /etc/bind/named.conf.local
```

Define a zone in which we need to forward all the requests containing the domain ”testlab.lan”:

```
zone ”testlab.lan.” {
    type forward;
    forward only;
    forwarders {[server_IP] port 53;};
};
```

Modify the options of Bind: $ sudo nano /etc/bind/named.conf.options

Write these instructions:

```
options {
    directory ”/var/cache/bind”;

    dnssec-validation no;
    listen-on-v6 {any; };
};
```

Check that the modifications are correct:

```
$ sudo named-checkconf
$ sudo systemctl restart bind9
```

4.4 DNSDist 1.8 installation

```
$ echo ”deb [arch=amd64] http://repo.powerdns.com/debian bullseye-dnsdist-18 main” | sudo tee /etc/apt/sources.list.d/pdns.list
```

Modify the file in path /etc/apt/preferences.d/dnsdist:

```
Package: dnsdist*
Pin: origin repo.powerdns.com
Pin-Priority: 600
```

Then, execute the following commands:

```
$ curl https://repo.powerdns.com/FD380FBB-pub.asc | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install dnsdist
```

Documentation source: https://repo.powerdns.com/

4.4.1 Configure the dnsdist server

Modify the configuration file of dnsdist:

```
$ sudo nano /etc/dnsdist/dnsdist.conf
```

We will first ask to dnsdist to listen on a particular port since by default it is listening on port 53 which is also the port on which Bind9 is listening. To do this, we will add "addLocal" option.

Then, we will add an Access Control List with the "addACL" option. We want that any IP of any subnet will be able to query the dnsdist server.

The next part is to add a recursive resolver to dnsdist because it does not have its own recursive resolver by default since it is a loadbalancer. To this end, we will use the "newServer" option with the loopback address because the recursive resolver is the system itself.

Finally, we will add the capability to manage DoH. But for this, we need to generate a private and a public key for the encryption of the traffic and a self-signed certificate. We will use OpenSSL to generate that.

```
addLocal ('0.0.0.0:5300')
addACL('0.0.0.0/0')
newServer({address="127.0.0.1:53"})
```

Generation of the encryption key pair and the certificate (we will not encrypt the private key in this case ⇒ -nodes):

```
$ cd /opt/
$ sudo openssl req −x509 −nodes −days 365 −newkey rsa:2048 −out dohpub.pem −
keyout dohpvt.pem
```

Fill in the form (Here is an example):

```
Country Name (2 letter code) [AU]:BE
State or Province Name (full name) [Some−State]:Belgium
Locality Name (eg, city) []:Bruxelles
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ULB
Organizational Unit Name (eg, section) []:MSECUC
Common Name (e.g. server FQDN or YOUR name) []:doh.local
Email Address []:admin@doh.local
```

Change the permission of the files created:

```
$ sudo chmod 777 dohp*
```

We go back to the dnsdist configuration file: `$ sudo nano /etc/dnsdist/dnsdist.conf`
Add the line to allow DoH communications:

```
addDOHLocal('0.0.0.0:443', '/opt/dohpub.pem','/opt/dohpvt.pem','/dns−query', {
doTCP=true, reusePort=true, sessionTimeout=43200})
```

Restart the dnsdist service and check the configuration:

```
$ sudo dnsdist −−check−config
$ sudo systemctl restart dnsdist
```

5 Client DNSCat2 (Debian 11)

5.1 Installation configuration

- **CPU thread**: 2
- **RAM**: 1024 MB (512 MB if without desktop)
- **Storage**: 50 GB
- **Network**: NAT Network
- **Root Password**: root
- **Non-admin username**: client
- **Non-admin password**: notroot
- **Hostname**: dohclient
- **Enable SSH server**

5.2 Basic configuration

5.2.1 Power setting (Mandatory)

In Settings > Power > Power Saving, put:

- Blank Screen: Never
- Automatic Suspend:
  - On Battery Power: OFF
  - Plugged In: OFF

5.2.2 User configuration

Add the user to sudoers (if without desktop version):

```
$ su -
# apt install sudo
# usermod -a -G sudo client
```

Restart the machine. (if without desktop, use: sudo shutdown -h now)

5.2.3 Install packages

```
$ sudo apt install git make gcc curl dtach wireshark tcpdump avahi-daemon
psmisc php net-tools
```

5.3 DNSCat2 client installation

```
$ git clone https://github.com/iagox86/dnscat2.git
$ cd dnscat2/client
$ make
```

5.4 DoH-proxy installation

In order to execute the DoH proxy, we need to install a Conda environment. Take the most recent Anaconda version on the Website and download it.Then, execute the following commands:

```
$ wget https://repo.anaconda.com/archive/Anaconda3-2023.03-1-Linux-x86_64.sh
$ bash Anaconda3-2023.03-1-Linux-x86_64.sh
$ sudo shutdown -h now
$ conda config --set auto_activate_base false
$ conda update conda
$ conda create --name DoH_proxyConda python=3.5
$ conda activate DoH_proxyConda
(DoH_proxyConda)$ git clone https://github.com/LoicCaudron/doh-proxy
(DoH_proxyConda)$ cd doh-proxy
(DoH_proxyConda)$ pip install .
(DoH_proxyConda)$ pip install --upgrade pip
(DoH_proxyConda)$ pip install aioh2==0.2.2
```

Try if the installation works with the following command and observe the connection with Wireshark ($ sudo wireshark).

```
(DoH_proxyConda)$ doh-stub --listen-address 127.0.0.1 --listen-port 5553 --
domain dns.google
```

You can test a DNS request through the DoH proxy with: `$ dig @127.0.0.1 -p 5553 twitch.tv`

If your host system is a **Linux system**, there should be no problem with TLS certificate verification and, by extension, HTTPS connection.

If your host system is a **Windows system** running antivirus software, there may be a TLS certificate verification error. Please check your antivirus settings to disable scanning of HTTPS encrypted connections.

Example with Kaspersky: Go to Configuration -¿ Network parameters. Check ”*Do not analyze encrypted connections*” in ”*Analyze encrypted connections (HTTPS)*”.

Retrieve the self-signed certificate

If you are on the ”without desktop” version, create a folder named ”Documents” on `/home/client` with: `$ mkdir Documents`

Retrieve the certificate generated on the DoH server (DNSDist machine) by using:

```
$ scp resolver@doh.local:/opt/dohpub.pem /home/client/Documents/dohpub.pem
```

6 Controller

6.1 Installation configuration

- **CPU thread**: 2
- **RAM**: 1024 MB (512 MB if without desktop)
- **Storage**: 30 GB
- **Network**: NAT Network
- **Root Password**: root

- **Non-admin username**: controller
- **Non-admin password**: notroot
- **Hostname**: controller
- **Enable SSH server**

6.2 Basic configuration

6.2.1 Power setting (Mandatory)

In Settings > Power > Power Saving, put:

- Blank Screen: Never
- Automatic Suspend:
  - On Battery Power: OFF
  - Plugged In: OFF

6.2.2 User configuration

Add the user to sudoers (if without desktop version):

```
$ su -
# apt install sudo
# usermod -a -G sudo controller
```

Restart the machine. (if without desktop, use: sudo shutdown -h now)

6.2.3 Install packages

```
$ sudo apt install python3-pip
$ sudo apt install git avahi-daemon
```

6.2.4 Install python modules

```
$ pip install fabric
$ pip install decorator
```

6.3 Download automation code

```
$ git clone https://github.com/LoicCaudron/
Local_Malicious_DoH_Dataset_Automation
```