

DNS-over-HTTPS local simulation environment

Master thesis document

Caudron Loïc



Course under the supervision of
Dr. DEBATTY Thibault

Academic year
2022 – 2023

Master in Cybersecurity
Cryptanalysis and Forensics

Contents

1	Minimum system requirements OS	3
1.1	Debian 11	3
1.2	Windows 10	3
2	VirtualBox Network setting	3
3	Server (Debian 11)	4
3.1	Installation configuration	4
3.2	Basic configuration	4
3.2.1	Power setting (Mandatory)	4
3.2.2	User configuration	4
3.2.3	Install packages	4
3.2.4	Modify firewall rules	5
3.2.5	Allow SSH connection as Root	5
3.3	DNSCat2 server installation	5
3.3.1	Install Ruby and RubyGems	5
3.3.2	DNSCat2 server	5
3.4	DNSExfiltrator server installation	5
3.4.1	Anaconda environment installation	5
3.4.2	DNSExfiltrator installation	6
3.4.3	Checking the reception of DNS requests on "without desktop" mode	6
4	Local DoH resolver	6
4.1	Installation configuration	6
4.2	Basic configuration	6
4.2.1	Power setting (Mandatory)	6
4.2.2	User configuration	7
4.2.3	Install packages	7
4.2.4	Modify firewall rules	7
4.3	Bind9 installation	7
4.4	DNSDist 1.8 installation	8
4.4.1	Configure the dnsdist server	8
5	Client DNSCat2 (Debian 11)	9
5.1	Installation configuration	9
5.2	Basic configuration	9
5.2.1	Power setting (Mandatory)	9
5.2.2	User configuration	10
5.2.3	Install packages	10
5.3	DNSCat2 client installation	10
5.4	DoH-proxy installation	10

6	Client DNSExfiltrator (Windows 10)	11
6.1	Installation configuration	11
6.2	Basic configuration (mandatory)	11
6.2.1	Disabling security protection	11
6.2.2	Machine configuration	12
6.3	WinDump installation	13
6.4	doh-proxy installation	14
6.5	DNSExfiltrator installation	15
6.6	Retrieve certification	15
6.7	Download automation code	15
7	Controller	15
7.1	Installation configuration	15
7.2	Basic configuration	15
7.2.1	Power setting (Mandatory)	15
7.2.2	User configuration	16
7.2.3	Install packages	16
7.2.4	Install python modules	16
7.3	Download automation code	16
8	DoH proxy (If need of a dedicated machine)	16
8.1	Installation configuration	16
8.2	Basic configuration	17
8.2.1	Power setting (Mandatory)	17
8.2.2	User configuration	17
8.2.3	Install packages	17
8.2.4	Modify firewall rules	17
8.3	Port forwarding configuration	17
8.4	DoH-proxy installation	18
A	Install VirtualBox on Debian 11	18
B	Install VirtualBox Guest Additions	19
C	Debian without desktop	19
C.1	Change keyboard language	19
C.2	Request NAT network address	19

1 Minimum system requirements OS

1.1 Debian 11

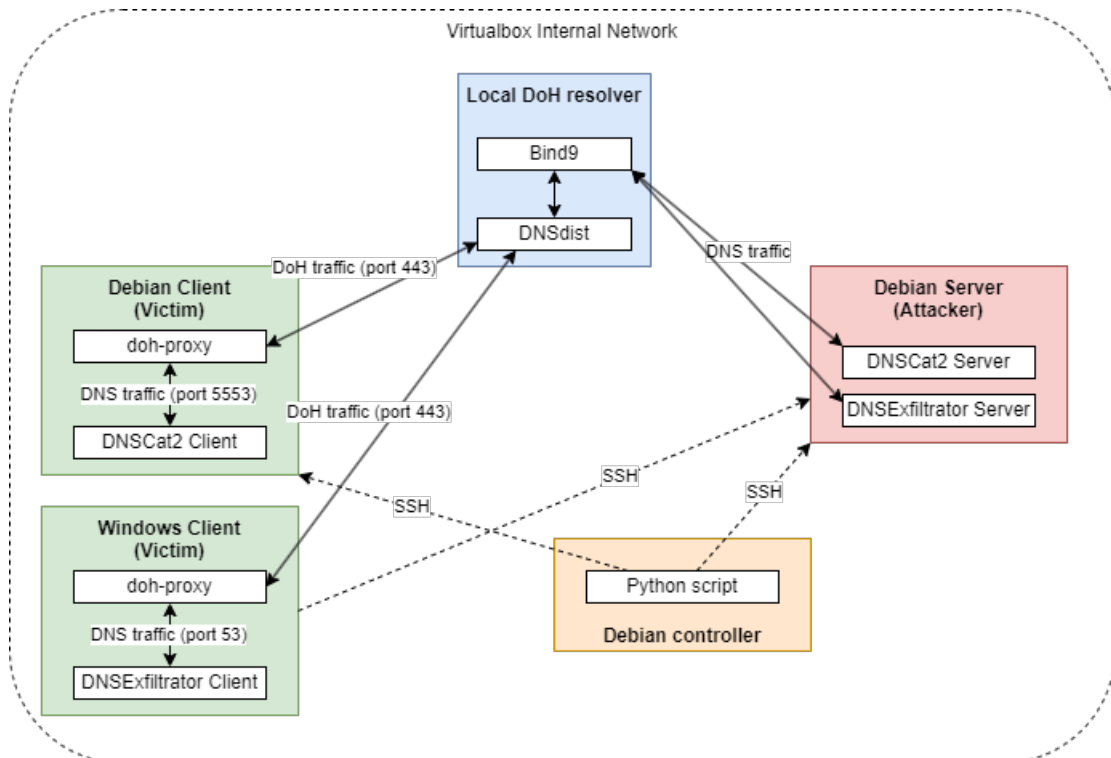
Install Type	RAM (minimum)	RAM (recommended)	Hard Drive
No desktop	128 megabytes	512 megabytes	2 gigabytes
With Desktop	256 megabytes	1 gigabyte	10 gigabytes

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):

```
1 $ su -
2 # apt install sudo
3 # usermod -a -G sudo attacker
```

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

3.2.3 Install packages

```
1 $ sudo apt install git make gcc dtach wireshark tcpdump ufw psmisc avahi-daemon
```

3.2.4 Modify firewall rules

```
1 $ sudo ufw enable
2 $ sudo ufw allow 53/udp
3 $ sudo ufw allow 22/tcp
4 $ sudo ufw reload
5 $ 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

```
1 $ sudo apt install ruby-dev
```

3.3.2 DNSCat2 server

```
1 $ git clone https://github.com/iagox86/dnscat2.git
2 $ cd dnscat2/server/
3 $ sudo apt-get install build-essential
4 $ sudo gem install salsa20 -v '0.1.1'
5 $ sudo gem install bundler
6 $ 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/>

3.4 DNSExfiltrator server installation

3.4.1 Anaconda environment installation

```
1 $ su -
2 # wget https://repo.anaconda.com/archive/Anaconda3-2023.03-1-Linux-x86_64.sh
3 # bash Anaconda3-2023.03-1-Linux-x86_64.sh
4 # sudo shutdown -h now
5 # conda config --set auto_activate_base false
6 # conda create -n det python=2.7.18
```

3.4.2 DNSExfiltrator installation

```
1 $ su -
2 # git clone https://github.com/Arno0x/DNSExfiltrator.git
3 # conda activate det
4 # cd DNSExfiltrator
5 # pip2 install -r requirements.txt --user
```

3.4.3 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):

```
1 $ su -
2 # apt install sudo
3 # usermod -a -G sudo resolver
```

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

4.2.3 Install packages

```
1 $ sudo apt install build-essential curl ufw wireshark avahi-daemon
```

4.2.4 Modify firewall rules

```
1 $ sudo ufw enable
2 $ sudo ufw allow 53/udp
3 $ sudo ufw allow 443/tcp
4 $ sudo ufw allow 22/tcp
5 $ sudo ufw reload
6 $ sudo ufw status
```

4.3 Bind9 installation

```
1 $ sudo apt install bind9 bind9utils bind9-doc
2 $ sudo systemctl status named
3 $ sudo nano /etc/bind/named.conf.local
```

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

```
1 zone "testlab.lan." {
2     type forward;
3     forward only;
4     forwarders {[server_IP] port 53;};
5 };
```

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

Write these instructions:

```
1 options {
2     directory "/var/cache/bind";
3
4     dnssec-validation no;
5     listen-on-v6 {any; };
6 };
```

Check that the modifications are correct:

```
1 $ sudo named-checkconf
2 $ sudo systemctl restart bind9
```

4.4 DNSDist 1.8 installation

```
1 $ 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`:

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

Then, execute the following commands:

```
1 $ curl https://repo.powerdns.com/FD380FBB-pub.asc | sudo apt-key add -
2 $ sudo apt-get update
3 $ sudo apt-get install dnsdist
```

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

4.4.1 Configure the dnsdist server

Modify the configuration file of dnsdist:

```
1 $ 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.

```
1 addLocal ( '0.0.0.0:5300 ' )
2 addACL( '0.0.0.0/0 ' )
3 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 \Rightarrow -nodes):

```
1 $ cd /opt/
2 $ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -out dohpub.pem
-keyout dohpvt.pem
```

Fill in the form (Here is an example):

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

Change the permission of the files created:

```
1 $ sudo chmod 777 dohp*
```

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

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

Restart the dnsmasq service and check the configuration:

```
1 $ sudo dnsmasq --check-config  
2 $ sudo systemctl restart dnsmasq
```

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):

```
1 $ su -
2 # apt install sudo
3 # usermod -a -G sudo client
```

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

5.2.3 Install packages

```
1 $ sudo apt install git make gcc curl dtach wireshark tcpdump avahi-daemon
psmisc php net-tools
```

5.3 DNSCat2 client installation

```
1 $ git clone https://github.com/iagox86/dnscat2.git
2 $ cd dnscat2/client
3 $ 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:

```
1 $ wget https://repo.anaconda.com/archive/Anaconda3-2023.03-1-Linux-x86_64.sh
sh
2 $ bash Anaconda3-2023.03-1-Linux-x86_64.sh
3 $ sudo shutdown -h now
4 $ conda config --set auto_activate_base false
5 $ conda update conda
6 $ conda create --name DoH_proxyConda python=3.5
7 $ conda activate DoH_proxyConda
8 (DoH_proxyConda)$ git clone https://github.com/LoicCaudron/doh-proxy
9 (DoH_proxyConda)$ cd doh-proxy
10 (DoH_proxyConda)$ pip install .
11 (DoH_proxyConda)$ pip install --upgrade pip
12 (DoH_proxyConda)$ pip install aiohttp==0.2.2
```

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

```
1 (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:

```
1 $ scp resolver@doh.local:/opt/dohpub.pem /home/client/Documents/dohpub.pem
```

6 Client DNSExfiltrator (Windows 10)

6.1 Installation configuration

- **CPU thread:** 2
- **RAM:** 2048 MB
- **Storage:** 50 GB
- **Network:** NAT Network
- **Root Password:** root
- **Non-admin username:** Client
- **Non-admin password:** notroot
- **Hostname:** exfilClient
- **Enable SSH server**

In order to create a local account in Windows 10:

- Don't connect the machine to Internet
- Say you don't have activation key
- Specify you don't have Internet connection and continue with the limited installation

6.2 Basic configuration (mandatory)

6.2.1 Disabling security protection

- Go to "*Start → Settings → Privacy & security → Windows Security.*"
- Select "Virus & threat protection." If you don't see this option, select "Open Windows Security," then select "Virus & threat protection."
- Select "Manage Settings."

- Toggle off “Tamper Protection.”
- Toggle off “Real-Time Protection.”
- Deactivate Windows Defender permanently by modifying the registry key
 - Make "Win+R"
 - Type "regedit"
 - Navigate to: HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender
 - Add a "DWORD (32-bit) Value" called "*DisableRealtimeMonitoring*"
 - Modify the value of the key to 1
 - Add a new key (folder) to "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender" called "Real-Time Protection"
(HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection)
 - Add a "DWORD (32-bit) Value" called "*DisableRealtimeMonitoring*" and set value to 1
- Run on Powershell with administrator rights: *Set-ExecutionPolicy ByPass*. Answer *T* for *Yes for all*.
- Allow traffic in port 53 with powershell:


```

1 PS > New-NetFirewallRule -DisplayName "Allow Port 53 UDP" -
    Direction Inbound -LocalPort 53 -Protocol UDP -Action Allow
2
```
- On **Microsoft Edge**:
 - Select *Settings and more (...)* → *Settings* → *Privacy, search, and services*.
 - Under *Security*, turn *Microsoft Defender SmartScreen* off.

WARNING: The Windows 10 update may eventually modify the values of added registry keys. This can lead to the detection and deletion of certain files considered malicious by the operating system, even if protections have been disabled. In this case, please check that no added registry key values have been modified.

6.2.2 Machine configuration

- Go in the power option and modify the parameters of the mode. Put the shutdown screen option on **Never** for "*On battery*" and "*On power supply*".
- **(Optional if hostname already configured)** Change the hostname with Powershell as Administrator with the command: *Rename-Computer -NewName "exfilClient"* and restart the computer for the changes to take effect.
- **Enable the SSH Server with Powershell as Administrator:**

```

1         > Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
2         > Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Server*'
3         > Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client*'
4         > netsh advfirewall firewall add rule name="SSH service" dir=in
    action=allow protocol=TCP localport=22
5         > Start-Service sshd
6         > Get-Service -Name *ssh*
7         > Set-Service -Name sshd -StartupType 'Automatic'
8         > Set-Service -Name 'ssh-agent' -StartupType 'Automatic'
9

```

• Install Python

- Add Python to Path when installing
- Install the Fabric library: `C:> pip install fabric`

• Disable IPv6:

- Type "*Control panel*" in the search bar and open it
- Open *Network and Internet*
- Open *Network and Sharing Center*
- Click on *Change Adapter Settings*
- Right-click your connection and go to *Properties*
- Uncheck the box next to *Internet Protocol Version6 (TCP/IPv6)* to disable it
- Select *OK* to confirm the change

Sources:

- <https://kb.nomachine.com/AR03S01117>
- https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse?tabs=powershell
- <https://www.hanselman.com/blog/how-to-ssh-into-a-windows-10-machine-from-linux-or-windows>
- <https://networking.grok.lsu.edu/Article.aspx?articleid=17573>

6.3 WinDump installation

- Download the last version on <https://www.winpcap.org/>
- Execute the installation program as Administrator
- Follow the installation assistant. By default, we can let default options.
- Download the WinDump executable on <https://www.winpcap.org/windump/> and put it on the Downloads folder.

To **capture traffic with a particular filter**, such as HTTPS traffic, simply run the following command:

```
1 C:> WinDump.exe -n -XX -s 0 -w C:\Users\[User]\Documents\capture.pcap -i 1
port 443
```

You can **list the various network** interfaces with the command:

```
1 C:> WinDump.exe -D
```

6.4 doh-proxy installation

- Download the last version of Anaconda on <https://www.anaconda.com/>
- Run the Anaconda executable and add Anaconda to the PATH during the installation.
- Execute the following command to install the tool and its environment:

```
1 C:> conda update conda
2 C:> conda create --name DoH_proxyConda python=3.5
3 C:> conda activate DoH_proxyConda
4 (DoH_proxyConda) C:> git clone https://github.com/LoicCaudron/doh-
proxy (or download it manually)
5 (DoH_proxyConda) C:> cd doh-proxy
6 (DoH_proxyConda) C:> pip install .
7 (DoH_proxyConda) C:> python -m pip install --upgrade pip
8 (DoH_proxyConda) C:> pip install aiohttp==0.2.2
9
```

- If not already done, on Powershell, in administrator mode:

```
1 PS > New-NetFirewallRule -DisplayName "Allow Port 53 UDP" -
Direction Inbound -LocalPort 53 -Protocol UDP -Action Allow
2
```

Example of command to verify and run the proxy:

```
1 (DoH_proxyConda) C:> doh-stub --listen-address 127.0.0.1 --listen-port 53
--domain dns.google
```

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)*".

Command to kill the doh-proxy process:

```
1 taskkill /IM doh-stub.exe /F
```


6.5 DNSExfiltrator installation

- Download the tool on the GitHub repository: <https://github.com/Arno0x/DNSExfiltrator>
- Unzip it
- Put it in the Documents folder
- It is ready to use

6.6 Retrieve certification

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

```
1 $ scp resolver@doh.local:/opt/dohpub.pem C:\Users\Client\Documents\dohpub.pem
```

6.7 Download automation code

```
1 $ git clone https://github.com/LoicCaudron/Local_Malicious_DoH_Dataset_Automation (or download it manually)
```

7 Controller

7.1 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

7.2 Basic configuration

7.2.1 Power setting (Mandatory)

In Settings > Power > Power Saving, put:

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

7.2.2 User configuration

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

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

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

7.2.3 Install packages

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

7.2.4 Install python modules

```
1    $ pip install fabric
2    $ pip install decorator
```

7.3 Download automation code

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

8 DoH proxy (If need of a dedicated machine)

8.1 Installation configuration

- **CPU thread:** 2
- **RAM:** 1024 MB (512 MB if without desktop)
- **Storage:** 15 GB
- **Network:** NAT Network
- **Root Password:** root
- **Non-admin username:** dohproxy
- **Non-admin password:** notroot
- **Hostname:** dohproxy
- **Enable SSH server**

8.2 Basic configuration

8.2.1 Power setting (Mandatory)

In Settings > Power > Power Saving, put:

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

8.2.2 User configuration

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

```
1 # su -
2 # usermod -a -G sudo resolver
```

Restart the machine.

8.2.3 Install packages

```
1 $ sudo apt install ufw dtach avahi-daemon
```

8.2.4 Modify firewall rules

```
1 $ sudo ufw enable
2 $ sudo ufw allow 53/udp
3 $ sudo ufw allow 5553/tcp
4 $ sudo ufw allow 22/tcp
5 $ sudo ufw reload
6 $ sudo ufw status
```

8.3 Port forwarding configuration

Source: <https://www.baeldung.com/linux/ufw-port-forward>

First of all, we will enable the port forwarding by modifying `/etc/ufw/sysctl.conf` and uncomment the line `net/ipv4/ip_forward=1`.

In order to set up a port forward from the 53 port to the 5553 port of the DoH proxy, we must edit the `/etc/ufw/before.rules` file. We will add a NET table after the filter table:

```
1 *nat
2 :PREROUTING ACCEPT [0:0]
3 -A PREROUTING -p tcp --dport 53 -j REDIRECT --to-port 5553
4 COMMIT
```

Restart UFW:

```
1 $ sudo systemctl restart ufw
```

8.4 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:

```
1 $ wget https://repo.anaconda.com/archive/Anaconda3-2023.03-1-Linux-x86_64.sh
sh
2 $ bash Anaconda3-2023.03-1-Linux-x86_64.sh
3 $ conda config --set auto_activate_base false
4 $ conda update conda
5 $ conda create --name DoH_proxyConda python=3.5
6 $ conda activate DoH_proxyConda
7 (DoH_proxyConda)$ git clone https://github.com/LoicCaudron/doh-proxy (or
download it manually)
8 (DoH_proxyConda)$ cd doh-proxy
9 (DoH_proxyConda)$ pip install .
10 (DoH_proxyConda)$ python -m pip install --upgrade pip
11 (DoH_proxyConda)$ pip install aiohttp==0.2.2
```

Example of command to verify and run the proxy:

```
1 (DoH_proxyConda) C:> doh-stub --listen-address 127.0.0.1 --listen-port 53
--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:

```
1 $ scp resolver@doh.local:/opt/dohpub.pem /home/client/Documents/dohpub.pem
```

A Install VirtualBox on Debian 11

```
1 $ wget -O- -q https://www.virtualbox.org/download/oracle_vbox_2016.asc |
sudo gpg --dearmor -o /usr/share/keyrings/oracle_vbox_2016.gpg
2 $ echo "deb [arch=amd64 signed-by=/usr/share/keyrings/oracle_vbox_2016.gpg]
http://download.virtualbox.org/virtualbox/debian bullseye contrib" | sudo
tee /etc/apt/sources.list.d/virtualbox.list
3 $ sudo apt update
4 $ sudo apt install virtualbox-7.0
```

```
5 $ wget https://download.virtualbox.org/virtualbox/7.0.6/  
Oracle_VM_VirtualBox_Extension_Pack-7.0.6.vbox-extpack  
6 $ sudo vboxmanage extpack install Oracle_VM_VirtualBox_Extension_Pack  
-7.0.6.vbox-extpack  
7 $ sudo usermod -a -G vboxusers $USER
```

source: <https://linuxiac.com/how-to-install-virtualbox-on-debian-11-bullseye/>

B Install VirtualBox Guest Additions

```
1 $ sudo apt -y install dkms build-essential linux-headers-$(uname -r)  
2 $ sudo reboot  
3 $ sudo mkdir /mnt/cdrom  
4 $ sudo mount /dev/cdrom /mnt/cdrom  
5 $ cd /mnt/cdrom  
6 $ sudo sh ./VBoxLinuxAdditions.run --nox11
```

C Debian without desktop

C.1 Change keyboard language

```
1 $ sudo dpkg-reconfigure keyboard-configuration
```

Restart the machine in order to have effect.

C.2 Request NAT network address

```
1 $ sudo systemctl restart networking  
2 $ sudo dhclient
```