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

December 1, 2023

Contents

1 Minimum system requirements OS

1.1 Debian 11

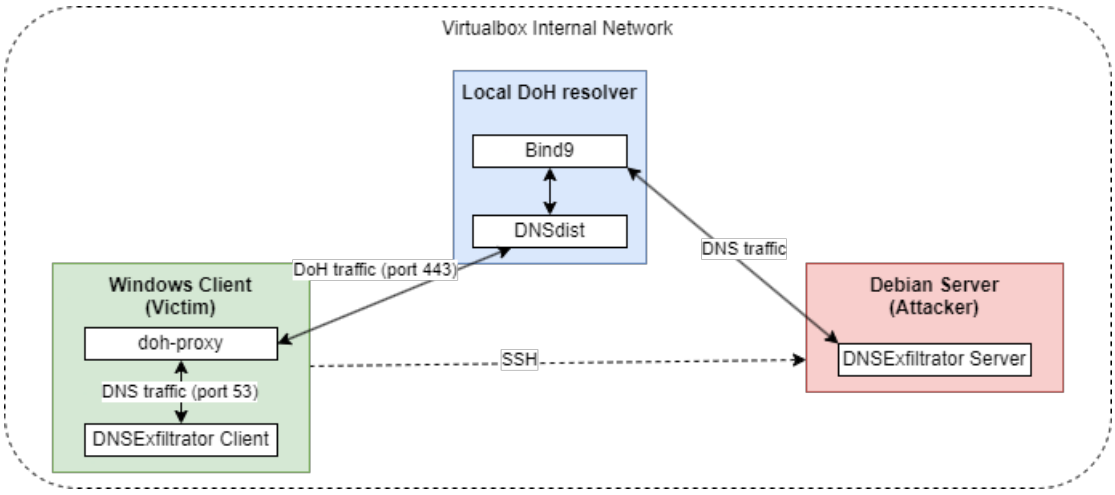
Install Type	RAM (minimum)	RAM (recommended)	Hard Drive
No desktop	128 megabytes	512 megabytes	2 gigabytes
With Desktop	256 megabytes	1 gigabyte	10 gigabytes

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.

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 DNSExfiltrator server installation

3.3.1 Anaconda environment installation

```
$ su -
# 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 create -n det python=2.7.18
```

3.3.2 DNSExfiltrator installation

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

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.

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 DNSExfiltrator (Windows 10)

5.1 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

5.2 Basic configuration (mandatory)

5.2.1 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:

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

```

- 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.

5.2.2 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 Admininistrator 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**:

```
> Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
> Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Server*'
> Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Client*'
> netsh advfirewall firewall add rule name="SSHD service" dir=in action=allow protocol=TCP localport=22
> Start-Service sshd
> Get-Service -Name *ssh*
> Set-Service -Name sshd -StartupType 'Automatic'
> Set-Service -Name 'ssh-agent' -StartupType 'Automatic'

```

- **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-wind`
- `https://networking.grok.lsu.edu/Article.aspx?articleid=17573`

5.3 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:

```
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:

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

5.4 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:

```
C:> conda update conda
C:> conda create --name DoH_proxyConda python=3.5
C:> conda activate DoH_proxyConda
(DoH_proxyConda) C:> git clone https://github.com/LoicCaudron/doh-proxy (or download it manually)
(DoH_proxyConda) C:> cd doh-proxy
(DoH_proxyConda) C:> pip install .
(DoH_proxyConda) C:> python -m pip install --upgrade pip
(DoH_proxyConda) C:> pip install aioh2==0.2.2

```

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

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

```

Example of command to verify and run the proxy:

```
(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:

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

5.5 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

5.6 Retrieve certification

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

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

5.7 Download automation code

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