

**Dark Web
&
DarkNet Recon for the CISO
Lab Guide
V1.05**

<https://darkweb.sh>

Introduction

This guide is a collection of exercises for the O'Reilly Live Training Hands-on

Learning Path: Additional Cybersecurity Training (Free with your O'Reilly Subscription)

This training is part of a learning path that has numerous live training sessions and video courses that are available with your O'Reilly subscription.

Lab Setup

Setup your lab using Kali Linux or Parrot OS Linux systems, and then use the scripts provide at <https://darkweb.sh>

1. Step 1: Download Kali or Parrot and install it on a VM
2. Step 2: After you have installed Kali Linux, run the following command from a terminal window to setup your environment:

Day 1

LAB 1 Demo – Using Shodan and ZoomEye.

Try for yourself:

<https://www.shodan.io/>

Selectors: html:"tor" html:"market"

<https://www.zoomeye.org/>

Selectors: "Dark Web" +"Tor"

Activity 1: Introduction to Dark Nets (surface websites)

Deep Web Search: <https://darkweb.sh/resources/>

Dark Web Terminology: <https://DarkWeb.sh/terminology/>

Dark Web Tools by Category: <https://darkweb.sh/tools/>

Tor Exit Nodes: <https://metrics.torproject.org/> & <https://metrics.torproject.org/rs.html#search/flag:Exit>

Tor Entry - Guard Nodes: <https://metrics.torproject.org/rs.html#search/flag:Guard>

Class Research Machine Template: <https://DarkWeb.sh/research/>

Lab 2: The Tor Network

Installing Tor, Tor Browser, and Tor Tools

The Long way around:

Download the tar file: tor-browser-linux64-11.5.2_en-US.tar.xz

```
tar -xf [tor_filename]
```

```
chmod +x start-tor-browser.desktop
```

Or

Easy mode:

```
apt-get install tor torbrowser-launcher -y
```

Installing Tor Browser on Kali Linux

Install Instructions

Open the terminal then run the following commands:

```
kali@kali:~$ sudo apt update
kali@kali:~$ 
kali@kali:~$ sudo apt install -y tor torbrowser-launcher
kali@kali:~$
```

As user run the following command:

```
kali@kali:~$ torbrowser-launcher
```

First time it will download and install Tor Browser including the signature verification.

Next time it will be used to update and launch Tor Browser.

Reference

- [Debian Wiki: TorBrowser](#)

LAB 3 - Build & Manage your Cloud based Recon Systems

Cloud Based Research Systems:

Management: RDP to Cloud Debian based Systems:

You can RDP from your Windows or Linux host to an Cloud based Linux or windows host.

1. Install xfce and xrdp on your linux host
2. Install RDP on linux
3. Create an account and enable auto start.



- A. [Xfce](#) or KDE is a lightweight desktop environment for UNIX-like operating systems. It aims to be fast and low on system resources, while still being visually appealing and user friendly.
- B. [XRDP/RDP](#) is the Remote Desktop Protocol which operates on Port 3389, xrdp provides a graphical login to remote Windows and Linux machines using Microsoft Remote Desktop Protocol (RDP). xrdp accepts connections from a variety of RDP clients.

Installing XRDP and XFCE on Linux:

Prepare the System

1. `sudo apt-get update -y`
2. `sudo DEBIAN_FRONTEND=noninteractive apt-get -y install xfce4`
3. `sudo apt install xfce4 xfce4-goodies xfce4-session xorg dbus-x11 x11-xserver-utils`
4. `sudo apt update && apt install tasksel -y`

Setup

1. `sudo apt-get -y install xrdp`
2. `sudo systemctl enable xrdp`
3. `sudo adduser xrdp ssl-cert` <- ubuntu requires a certificate
4. `sudo passwd azureuser` <-create a local user, used to access
5. `echo xfce4-session > ~/.xsession` <-tell xrdp what desktop environment to use.

Start xrdp

1. `sudo service xrdp restart`
2. `sudo service xrdp status`

Check the Status:

```
root@researcher001:~# service xrdp status
● xrdp.service - xrdp daemon
   Loaded: loaded (/lib/systemd/system/xrdp.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-10-11 06:11:18 UTC; 36s ago
     Docs: man:xrdp(8)
           man:xrdp.ini(5)
```

RDP on Windows 10 & 11:

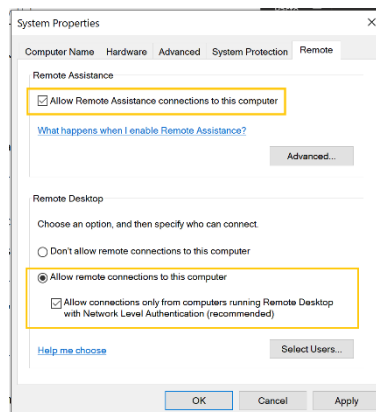
You do not need to setup RDP on your local system, however if your cloud system is windows you will follow these instructions:

- ① Set up the PC you want to connect to so it allows remote connections:
 - Make sure you have Windows 10 Pro. To check, go to **Start** > **Settings** > **System** > **About** and look for **Edition**. For info on how to get it, go to [Upgrade Windows 10 Home to Windows 10 Pro](#).
 - When you're ready, select **Start** > **Settings** > **System** > **Remote Desktop**, and turn on **Enable Remote Desktop**.
 - Make note of the name of this PC under **How to connect to this PC**. You'll need this later.
- ② Use Remote Desktop to connect to the PC you set up:
 - **On your local Windows 10 PC:** In the search box on the taskbar, type **Remote Desktop Connection**, and then select **Remote Desktop Connection**. In Remote Desktop Connection, type the name of the PC you want to connect to (from Step 1), and then select **Connect**.
 - **On your Windows, Android, or iOS device:** Open the Remote Desktop app (available for free from Microsoft Store, Google Play, and the Mac App Store), and add the name of the PC that you want to connect to (from Step 1). Select the remote PC name that you added, and then wait for the connection to complete.

Click here to go to the tab in Windows: [Enable Remote Desktop](#)

You can also enable Remote Desktop using the **System Properties**. Press the Windows Key and Type: **advanced system**. Click View advanced system settings. This gives you similar screens “Remote Assistance” and a check box “Allow Remote Assistance connections to this computer”, and Remote Desktop select radio button “Allow Remote Connections to this Computer” and check Allow Connections.

You can also go to Settings > connection settings in the app and configure the appearance, devices and make other adjustments that work for you



Cloud Based Systems

Demo the Azure Cloud based – Cloud Desktop

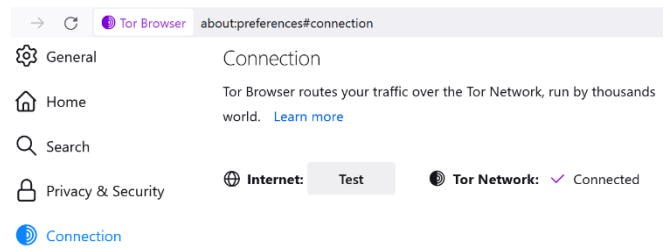
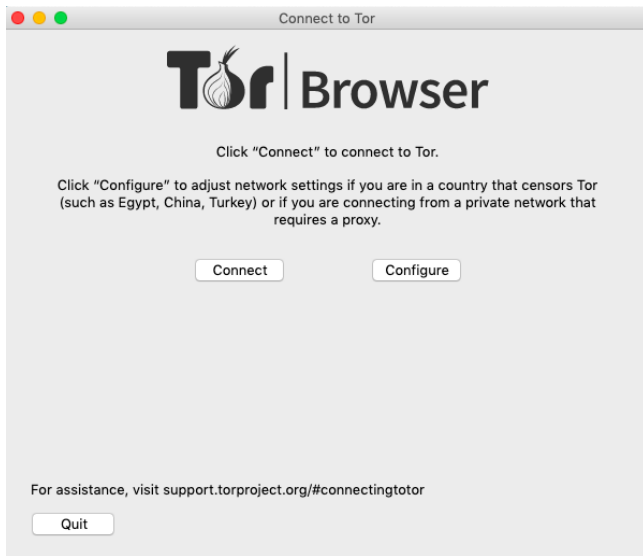
Amazon Workspaces – Amazon Workstation

Google – Cloud Based Virtual Workstation

<https://cloud.google.com/architecture/creating-a-virtual-windows-workstation>

LAB 4 – Configuring your Tor Browser for Optimal Protections

Tor Browser Settings Lab



Security

Security Level

Disable certain web features that can be used to attack your security and anonymity. [Learn more](#)

☐ Standard

All Tor Browser and website features are enabled.

☐ Safer

Disables website features that are often dangerous, causing some sites to lose functionality.

☒ Safest

Only allows website features required for static sites and basic services. These changes affect images, media, and scripts.

- JavaScript is disabled by default on all sites.
- Some fonts, icons, math symbols, and images are disabled.
- Audio and video (HTML5 media), and WebGL are click-to-play.



LAB 5 - Messaging and Email Exercise



- **Tor Mail**

<http://tormailpout6wplxlrrkhhjj2ra7bmqaij5iptdmhnnep3r6f27m2yid.onion/mail/src/login.php>

MEGA Tor - Chat

Dark Chat Platform

- **MegaChat** – <http://ylmjp76zk4ndvgpncbtgzrfsrzpblvlzgtuoduggygwdlexou64iwfqd.onion/>

LAB 5 - Configuring Socks, Sock Proxy's and Browser Socks

Three Options

1. Use [FoxyProxy](#) add to [Firefox](#)
 2. Change manually in each browser setting
 3. Configure in Proxychains
-

<https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/>



2. Browser Proxy Settings

Tor Firefox under: Settings > Connection > Advanced >

Configure how Tor > Settings

Regular Firefox under: Settings > Network Settings > Settings >

Manual Proxy Configuration



SPYS.ONE/EN/	Free proxy list	Proxy list by country	Anonymous free proxy	HTTPS/SSL proxy	SOCKS proxy list
Proxy by ASN/ORG	Proxy by cities	Proxy by ports	HTTP proxy list	Transparent proxy list	IPinfo
HTTPS SSL proxy servers list. Free HTTP proxies with SSL support					
Proxy address:port	Proxy type	Anonymity*	Country (city)	Hostname/ORG	Latency** Speed*** Uptime
200.94.142.220:999	HTTPS (Mikrotik)	NOA	Mexico (Jaipa de Mendez)	static-200-94-142-220.alestra.net.mx (Alestra, S. de R.L. de C.V.)	6.127 new:-
185.252.28.98:9090	HTTPS (Mikrotik)	NOA	Iran	185.252.28.98 (Shabakeh Ertebatat Artak Towseeh LTD)	4.74 1.0%(8)-
188.133.153.227:1256	HTTPS (Mikrotik)	NOA	Russia (Moscow)	227.153.133.188.msk.enforta.com (JSC ER-Telecom Holding)	8.482 33%(9)-
159.65.63.209:8888	HTTPS	HIA	United Kingdom (London)	159.65.63.209 (DIGITALOCEAN-ASN)	12.346 29%(20)-
40.129.203.4:8080	HTTPS	HIA	United States (Holtwood) !!!	h4.203.129.40.static.ip.windstream.net (WINDSTREAM)	1.048 77%(41)+

LAB 6 – Proxy Chains CONFIGURATION, USAGE

A. The proxychains configuration file is located under `/etc/` directory. We will open the **proxychains4.conf** file - first open a terminal - the enter the following sudo (using VI, View or Nano) edit ->

`/etc/proxychains4.conf`

B. **Strict_Chain**, this chains the ip's you list in order so that the final ip address is the last one on your list, if a chain is unresponsive the chain fails and you get nothing. ii. **Dynamic Chain** works similar to strict chain, only it does not require all proxies in the Chain configuration file to work, if a proxy is down then the connection jumps to the next proxy in the list, you will see these on the command line, this lets you remove unresponsive proxies.

C. **Random Chain** adds randomness proxy select ability to the list and the chain will look different each time its used.

```

root@kali: ~
File Edit View Search Terminal Help
GNU nano 2.9.5 /etc/proxychains.conf

# proxychains.conf  VER 3.1
#
# HTTP, SOCKS4, SOCKS5 tunneling proxyifier with DNS.
#
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
#dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain
#
# Strict - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# all proxies must be online to play in chain
# otherwise EINTR is returned to the app
#
#random_chain
#
# Random - Each connection will be done via random proxy
# (or proxy chain, see chain_len) from the list.
# this option is good to test your IDS :)
#
# Make sense only if random_chain
#chain_len = 2

```

```

apt-get install proxychains tor -y
edit /etc/proxychains4.conf
Comment out #strict_chain
Uncomment - dynamic_chain

```

Make sure that your proxy_dns is uncommented

To use one of the three selected methods, comment (#) out two and leave the one you want to use uncommented.

We are going to select **dynamic_chain** and comment out the others. Its the best for speed and skips offline or non responsive proxies

Make sure you uncomment #proxy_dns this protects your dns requests so that they appear to be coming from the same source ip (and not yours)

Next we will locate and place all of our proxies in the file, to find a list of open proxies, there are a couple sites, for this exercise we will use “freeproxylists.net”

<https://www.freeproxylists.net> or <https://spys.one> has a nice layout, there are lots of proxies and a google search can help you find even more.

```
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
# socks5      127.0.0.1      9050
# socks5      213.59.123.90  9050
http 13.57.51.106 5432
http 68.183.181.188 8080
-- INSERT -- W10: Warning: Changing a readonly file
```

Now lets launch proxy chains and see if it works:

From a terminal window enter:

proxychains firefox duckduckgo.com,

You will need a socks5 proxy(s) to perform the nmap scan mentioned below:

proxychains nmap -sT -p 80,443 1.1.1.1

If you see something like this or a yellow box around your page, know that they are trying to snoop on you with a man in the middle. Close, and locate another proxy server.



Now lets add Tor to the configuration and even further anonymize our traffic

You should first stop tor -

LAB 7 – VPN SETUP, CONFIGURATION, USAGE

VPN – Linux, Debian, Kali, Ubuntu (install openvpn)

1. Open the terminal from the menu or `ctrl + Alt + t` at the same time
2. Install OpenVPN network manager by entering (copy/paste) into the terminal:

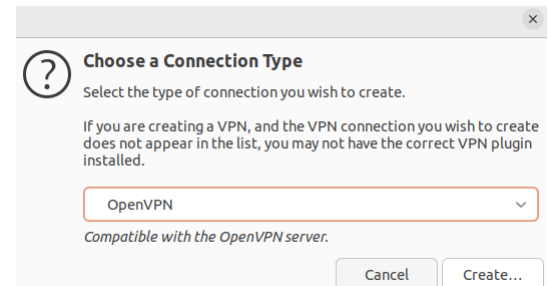
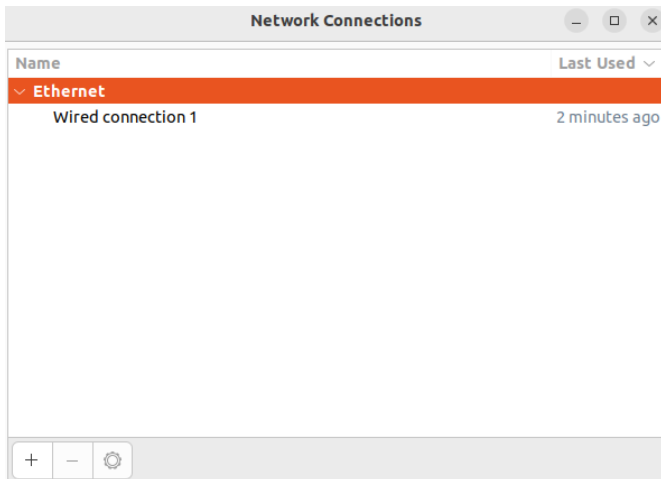
```
sudo apt-get install network-manager-openvpn
```

and hit Return or Enter

```
sudo apt-get install network-manager-openvpn-gnome
```

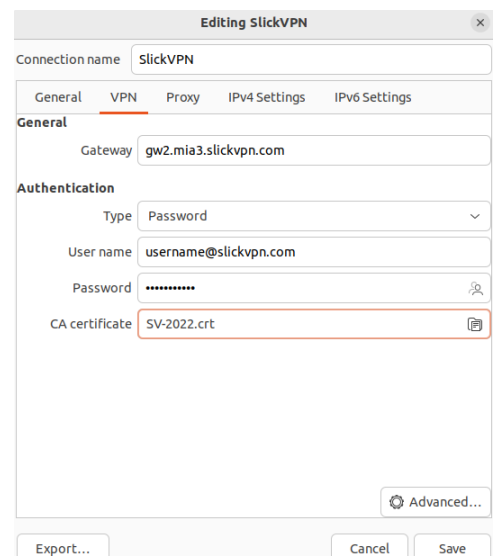
and hit Return or Enter

3. Restart the Network Manager
`systemctl restart NetworkManager`
4. Go to Network Manager, click Network Connections. Click the '+' button in the Network Connections window and choose OpenVPN from the dropdown menu



5. In the Editing VPN connection window, Enter the following details:

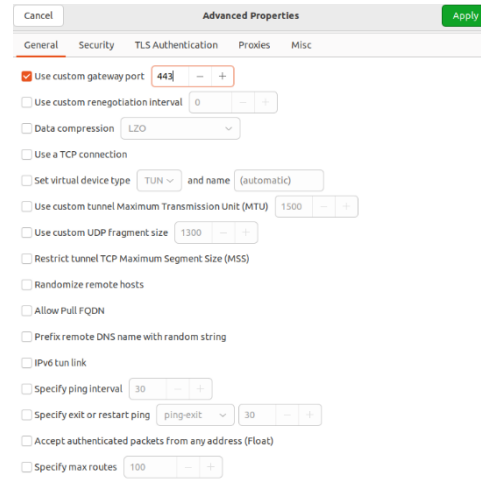
1. **Connection name:** SlickVPN, IPVanish, NordVPN, OVPN, ETC
2. **Gateway:** see.your.provider.com or choose a gateway
3. **Type:** Password
4. **Username:** Username
5. **Password:** Main Password
6. **CA Certificate:** vpn.crt



6. Click Advanced near the bottom of the window.

In the General tab

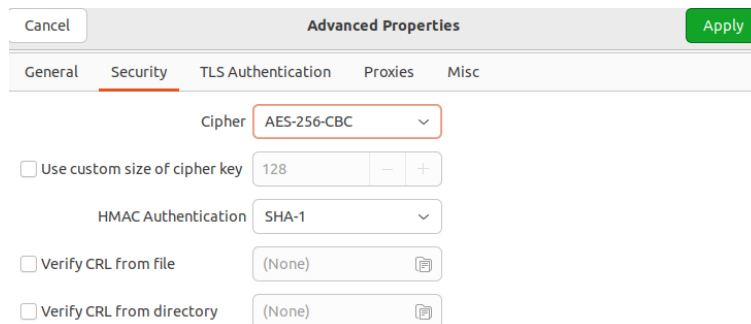
1. **Custom gateway port:** 443 or 8888
2. **[Optional]:** Use TCP connection



The Security tab

Cipher: AES-256-CBC

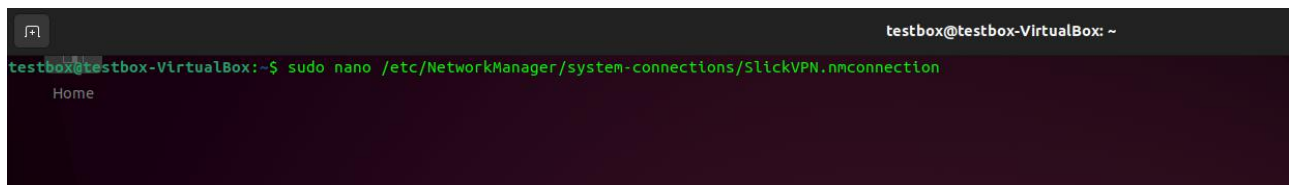
HMAC Authentication: SHA-1



Click Apply; and Click Save

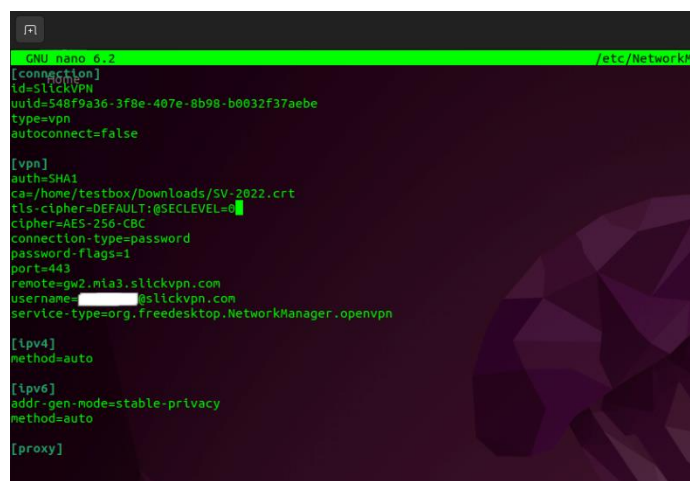
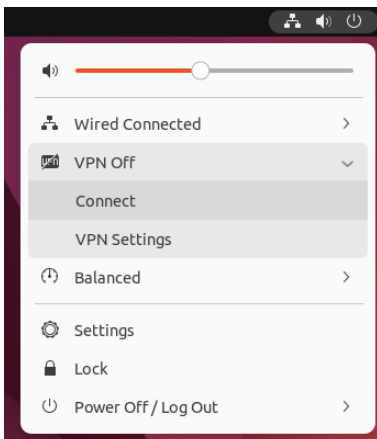
7. Go back to Terminal and enter the following

```
sudo nano /etc/NetworkManager/system-connections/VPN.connection
```



add this line under ca

```
tls-cipher=DEFAULT:@SECLEVEL=0
```



8. Restart Network Manager

```
systemctl restart NetworkManager
```

9. Start the connection to the VPN from Network Manager

LAB 7b – VPN SETUP Using Private VPN

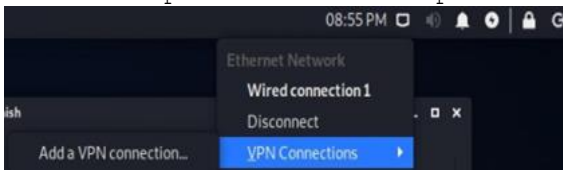
```
sudo apt-get install -y openvpn network-manager-openvpn sudo apt-get install -y network-manager-openvpn-gnome
```

Select your VPN of your choice, I selected ipvanish (not a recommendation)

1. `mkdir ~/ipvanish`

```
wget https://www.ipvanish.com/software/configs/configs.zip
```

3. Unzip the file "unzip" configs.zip



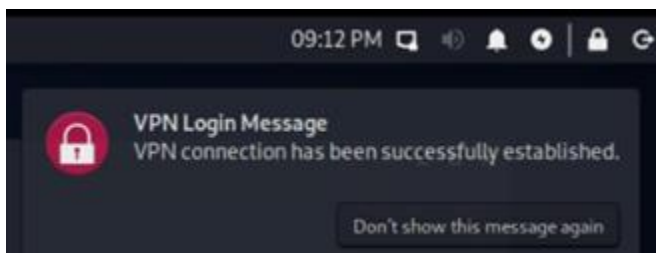
4. Select import a saved VPN configuration and then next, create, and locate the file. And save.



5. In the Add VPN window appears that requires you to enter your username and password for your account



6. Next select the network icon again, and you will see the profile for the site you selected, **simply click on it** and you will connect and shortly see the following message confirming you are connected



Note: You do not have an "ipvanish" account so this will not work for you.

LAB 8 – BUILDING A DARK WEB SERVER

1. Lets configure our workstation to host a website on the darkweb, step 1 is already done, installing "TOR" and making sure you can run it as a service, lets check that out.
 - a. From a terminal run "Service tor status" if its not running you can start it with running "service tor start" or "service tor restart" or to stop it service tor stop.
 - b. Next lets edit the Tor RC file located under /etc/tor/torrc using view, leafpad or nano which ever your comfortable using: From a terminal window enter:
 - i. `sudo /etc/tor/torrc`

```
File Actions Edit View Help
## HiddenServicePort x y:z says to redirect requests on port x to the
## address y:z.

#HiddenServiceDir /var/lib/tor/hidden_service/
#HiddenServicePort 80 127.0.0.1:80

#HiddenServiceDir /var/lib/tor/other_hidden_service/
#HiddenServicePort 80 127.0.0.1:80
#HiddenServicePort 22 127.0.0.1:22
```

- You should scroll down until you see the #HiddenServiceDir notice the path to those services /var/lib/tor/other_hidden_service/ Next we will edit the file making the following changes:

```
File Actions Edit View Help
## HiddenServicePort x y:z says to redirect requests on port x to the
## address y:z.

HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:80

HiddenServiceDir /var/lib/tor/other_hidden_service/
HiddenServicePort 80 127.0.0.1:80
HiddenServicePort 22 127.0.0.1:22
```

- Now lets change to the directory we created to host our website,

[cd /var/www/onion](#) And create a simple webpage touch index.html

view index.html (or which ever editor you prefer)

Add content similar to below and save the file

```
File Actions Edit View Help
<html>
  <title> UNDERWORLD </title>
<body>
  <center>
<h1>UNDERWORLD CRIME CINDIKATE </h1>
<BR><BR>
<h3> for access contact an admin </h3>
</center>
</body>
</html>
```

- Next we can use php to create a simple web server
- Where the page is execute: [php -S 127.0.0.1:80](#)
- Or install nginx as show in the power point

```
(root@TH21) ~[/var/www/onion]
# php -S 127.0.0.1:80
[Tue Feb 23 23:18:49 2021] PHP 7.4.11 Development Server (http://127.0.0.1:80) started
```

You should receive feedback similar to the above output, showing the service started.

- Next lets open firefox and browse to <http://127.0.0.1:80> with or without the 80 should work, since its the default port, but this is a simple php server so there is no https. You should see a similar page as shown below to confirm your system is working.
- Now lets see if its accessible from the “Dark Web” so whats our onion address and how can people find it ?

- c. Just run this command from a new terminal window, we do not want to close the one hosting the php server so open a new terminal and type in
- d. `cat /var/lib/tor/hidden_service/hostname`
- e. Or you can change to the directory and see the key's and the hostname file as shown.

```
(root@TH21) [/var/lib/tor/hidden_service]
# ls -la
total 24
drwx--S--- 3 debian-tor debian-tor 4096 Feb 23 23:25 .
drwx--S--- 5 debian-tor debian-tor 4096 Feb 23 23:25 ..
drwx--S--- 2 debian-tor debian-tor 4096 Feb 23 23:25 authorized_clients
-rw----- 1 debian-tor debian-tor   63 Feb 23 23:25 hostname
-rw----- 1 debian-tor debian-tor   64 Feb 23 23:25 hs_ed25519_public_key
-rw----- 1 debian-tor debian-tor   96 Feb 23 23:25 hs_ed25519_secret_key

(root@TH21) [/var/lib/tor/hidden_service]
# more hostname
6tfegfupndrl6ftzp6rhtqpt4rmbocszvlbbyc7o3znyeyjamo3ghlqd.onion
```

There you have it, your now a webhoster on the Darkweb. Now this is extremely unsafe the way we ran php, and didnt setup rights and such, you can install/run apache or nginx for more robust and secure hosting.

Another option would be to run this: `python3 -m http.server --bind 127.0.0.1 80`

LAB 8 – A Vanity Tor URL

A **vanity url**, instead of all these weird combo of letters and numbers, no problem there is a tool for that, it's called **mkp224o** and it is a vanity address generator for Tor Onion v3 Hidden Services, created by [cathugger](#) and available [on Github](#).

See the Power Point for the full options.,,

How do I make tor use of generated keys?

Copy key folder (though technically only `hs_ed25519_secret_key` is required) to where you want your service keys to reside: `sudo cp -r onionsite54....onion /var/lib/tor/onionsite`

You many need to adjust ownership permissions:

```
sudo chmod -R u+rwX,og-rwx /var/lib/tor/onionsite
```

```
git clone https://github.com/cathugger/mkp224o.git
```

```
apt-get install openssl
```

```
apt-get install libsodium-dev
```

```
apt-get install libboost-all-dev
```

```
apt-get install autoconf
```

```
apt-get install automake
```

```
./autogen.sh
```

```
./configure
```

```
make
```


LAB 15 Crawler - Seed Sites, Hunting for Sites.

Unlike 'Torrent' seeding. We are specifically focused on the Dark Web and Crawling, and we will seed (place a few hosts) in the config that can be crawler and branched from there.

We will use several Docker containers, Torproxy, Ache, and Elastic. This exercise will take over one hour to install, test and perform appropriately.



<https://github.com/ViDA-NYU/ache>

<https://ache.readthedocs.io/en/latest/tutorial-crawling-tor.html>

Running using Docker

Prerequisite: You will need to install a recent version of Docker. See <https://docs.docker.com/engine/installation/> for details on how to install Docker for your platform.

<https://hub.docker.com/r/dperson/torproxy/>

```
sudo docker run -it -p 8118:8118 -p 9050:9050 -d dperson/torproxy
```

```
git clone https://github.com/ViDA-NYU/ache.git
cd ache
./gradlew installDist
```

1. mkdir config_docker_tor/
cd config_docker_tor/
curl -O https://raw.githubusercontent.com/ViDA-NYU/ache/master/config/config_docker_tor/ache.yml
curl -O https://raw.githubusercontent.com/ViDA-NYU/ache/master/config/config_docker_tor/docker-compose.yml
curl -O https://raw.githubusercontent.com/ViDA-NYU/ache/master/config/config_docker_tor/tor.seeds
2. Cd config_docker_tor
3. docker-compose up -d

How to operate ACHE <https://ache.readthedocs.io/en/latest/tutorial-crawling-tor.html>

Search Scope: ache.yml:

LAB 16 Passwords

PASSWORDS:

Finding password Dumps

- Combining TheHarvester and Crosslinked tools, you can see if an organizations email address have been compromised. <https://github.com/khast3x/h8mail>
- 1. apt install python3-pip
- 2. Pip3 install h8mail
- 3. cd /home/kali/.local/bin

















Acquire and install API keys, this provides you the option to use premium services.

Python3 h8mail -g (like: haveibeenpwnd)

\$> view /home/kali/.local/bin/h8mail_config.ini

(enter your api here after hibp)

Features

-  Email pattern matching (reg exp), useful for reading from other tool outputs
-  Pass URLs to directly find and target emails in pages
-  Loosey patterns for local searches ("john.smith", "evilcorp")
-  Painless install. Available through pip, only requires requests
-  Bulk file-reading for targeting
-  Output to CSV file or JSON
-  Compatible with the "Breach Compilation" torrent scripts
-  Search cleartext and compressed .gz files locally using multiprocessing
 -  Compatible with "Collection#1"
-  Get related emails
-  Chase related emails by adding them to the ongoing search
-  Supports premium lookup services for advanced users
-  Custom query premium APIs. Supports username, hash, ip, domain and password and more
-  Regroup breach results for all targets and methods
-  Includes option to hide passwords for demonstrations
-  Delicious colors

Python3 h8mail -h

Python3 h8mail -t <target email address>

Python3 h8mail -t victim@email.com -c

Premium pay-for services such as HaveIbeenpwned.com

<https://haveibeenpwned.com/Passwords>

Pastebin

<https://cybernews.com/personal-data-leak-check>

<https://snusbase.com/>

LAB 16b Passwords

PwnedOrNot

PwnedOrNot is a OSINT tool to find passwords for compromised email addresses. pwnedOrNot uses haveibeenpwned v3 api to test email accounts and tries to find the password in Pastebin Dumps.

- Name of Breach
- Domain Name
- Date of Breach
- Fabrication status
- Verification Status
- Retirement status
- Spam Status

And with all this information **pwnedOrNot** can easily find passwords for compromised emails if the dump is accessible and it contains the password.

<https://github.com/thewhiteh4t/pwnedOrNot>

```
git clone https://github.com/thewhiteh4t/pwnedOrNot.git
cd pwnedOrNot
chmod +x install.sh
./install.sh
```

Docker:

```
git clone https://github.com/thewhiteh4t/pwnedOrNot.git
docker build -t pon .
docker run -it pon
```

```
python3 pwnedornot.py -h
usage: pwnedornot.py [-h] [-e EMAIL] [-f FILE] [-d DOMAIN] [-n] [-l]
                    [-c CHECK]
```

Complete command line examples: <https://github.com/thewhiteh4t/pwnedOrNot>

```
# Get only Breach Info, Skip Password Dumps
python3 pwnedornot.py -e <email> -n
```



LAB 17 OnionScan

Tools: Lab: OnionScan

1. Remove already installed Go **From a terminal:**

```
sudo apt-get remove golang-go
```

```
sudo apt-get remove --auto-remove golang-go
```

2. Browse to the Go website and download <https://golang.org/> version 1.16



Linux

Linux 2.6.23 or later, Intel 64-bit processor

[go1.16.linux-amd64.tar.gz](#) (123MB)

OnionScan in a Docker container Based on <https://github.com/s-rah/onionscan>.

The purpose of this tool is to make you a better onion service provider. You owe it to yourself and your users to ensure that attackers cannot easily exploit and deanonymize.

Run the container like so:

```
docker run -d --cap-drop=all --name onionscan mpatton/onionscan
```

This will startup the Tor connection and allow you to do the following:

```
docker exec -it onionscan onionscan <options> <onion address>
```

See <https://github.com/s-rah/onionscan> for further information.

Note: Run the following to make sure Tor has fully started, prior to running your scan:

```
docker logs -f onionscan
```

1. Suggest you reboot after you remove the old version of Go.
2. Go to your downloads directory under your user and open a terminal window

```
tar -C /usr/local/ -xzf go1.16.linux-amd64.tar.gz
```

```
export GOROOT=/usr/local/go export
```

```
GOPATH=/root/go-workspace
```

```
PATH=$PATH:$GOROOT/bin/:$GOPATH/bin go version
```

3. Next we download Onionscan

```
go get github.com/s-rah/onionscan
```

a. Next do a “go get” for each of the below packages

```
go get golang.org/x/crypto/openpgp/packet
```

```
go get golang.org/x/net/proxy - For the Tor SOCKS Proxy connection
```

go get golang.org/x/net/html - For HTML parsing go get github.com/rwcarlsen/goexif - For EXIF data extraction
go get github.com/HouzuoGuo/tiedot/db - For crawl database

4. To test you can find an .onion address and scan similar to below:

```
onionscan --verbose cardsa2u7pvmdamw.onion onionscan --jsonReport cardsa2u7pvmdamw
```

5. There is a OnionScan Correlation Lab you can try out as well depending on where your onionscandb is located you may need to change the path.

```
onionscan --mode analysis --dbdir ~/go/bin/onionscandb
```

LAB 17b OnionScan Docker Container

1. Make sure Docker is installed
2. Execute Docker Commands as follows:

```
docker run -ti --rm --name onionscan -v ${HOME}:/home/onionscan alexandreoda/onionscan``
```

Installing

Install OS dependencies

> On Debian based operating systems:

```
root@root-KitPloit: ~  
$ sudo apt-get install libexif-dev
```

> On Fedora based operating systems:

```
root@root-KitPloit: ~  
$ sudo dnf install libexif-devel
```

Grab with go get

```
root@root-KitPloit: ~  
$ go get github.com/s-rah/onionscan
```

Compile/Run from git cloned source

```
root@root-KitPloit: ~  
$ go install github.com/s-rah/onionscan
```

and then run the program in

```
root@root-KitPloit: ~  
$ ./bin/onionscan
```

Or, you can just do

```
root@root-KitPloit: ~  
$ go run github.com/s-rah/onionscan.go
```

to execute without compiling.

Running

For a simple report detailing the high, medium and low risk areas found:

```
root@root-KitPloit: ~  
$ ./bin/onionscan blahblahblah.onion
```

The most interesting output comes from the verbose option:

```
root@root-KitPloit: ~  
$ ./bin/onionscan --verbose blahblahblah.onion
```

There is also a JSON output, if you want to integrate with something else:

```
root@root-KitPloit: ~  
$ ./bin/onionscan --jsonReport blahblahblah.onion
```

If you would like to use a proxy server listening on something other than

```
root@root-KitPloit: ~  
$ 127.0.0.1:9050
```

, then you can use the --torProxyAddress flag:

```
root@root-KitPloit: ~  
$ ./bin/onionscan --torProxyAddress=127.0.0.1:9150 blahblahblah.onion
```

LAB 18 Katana-ds Docker Container

Katana-ds



1. Git clone <https://github.com/TebbaaX/Katana.git>
2. cd Katana
python3 -m pip install -r requirements.txt
python3 k cd Katana
3. python3 kds.py -h (for help)
Options :
-g :for google mode
-s :for scada mode
-t :for tor mode
-p :for proxy mode ds.py

Option 2

1. <https://github.com/K-Phoen/docker-sabre-katana.git>
2. docker run -d -p 8080:80 --name kphoen/sabre-katana
3. Store data elsewhere: docker run -d -p 8080:80 -v \$(pwd)/data:/var/www/html/data --name kphoen/sabre-katana
4. python3 kds.py -h (for help)
Options :
-g :for google mode
-s :for scada mode
-t :for tor mode
-p :for proxy mode ds.py

LAB 19 Hunchly Investigative tool



Optional: You can start with installing a 30 day trial of Hunch.ly , go to the website select free 30 day trial that will take you to the registration page, you must register to get a key.

1. <https://www.hunch.ly//try-it-now>
2. Select Download for Linux, Download the .deb file
3. After download select "Show in Folder", it show up as hunchly.deb
4. Next download your key file from email to the same download folder
5. Next open a terminal and type: **sudo apt install gnome-software**

6. Lets install google chrome next, a requirement for Hunchly

a. Using a **terminal** window type in the below wget to obtain the latest .deb file

wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb

7. From the same terminal window, Now lets install Google Chrome **apt install ./googlechrome-stable_current_amd64.deb**

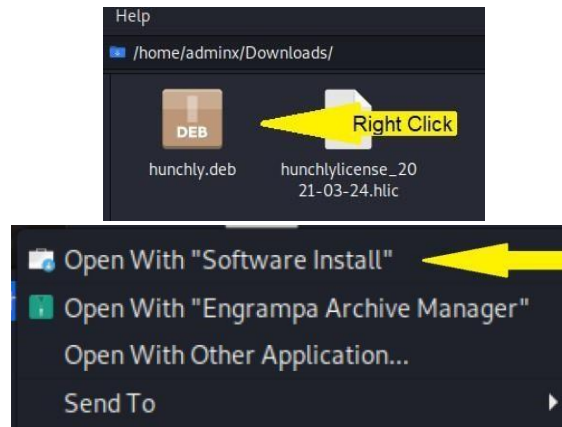
8. From the same terminal window you can launch chrome by type in **google-chrome --nosandbox**

9. With Chrome browse to the chrome webstore and add the Hunchly 2.0 extension

[https://chrome.google.com/webstore/detail/hunchly-](https://chrome.google.com/webstore/detail/hunchly-20/amfnegileeghgikpggcebehdepknalbf?hl=en)

[20/amfnegileeghgikpggcebehdepknalbf?hl=en](https://chrome.google.com/webstore/detail/hunchly-20/amfnegileeghgikpggcebehdepknalbf?hl=en)

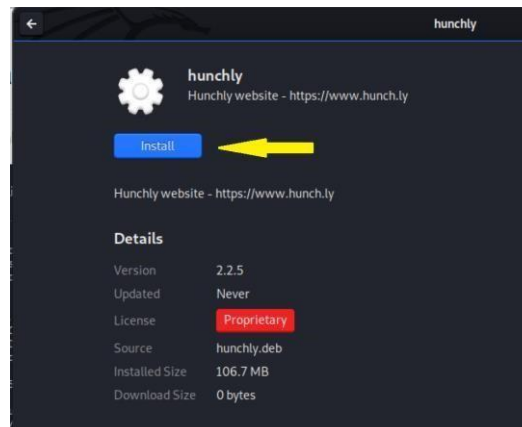
10. Next locate the Hunchly .deb file you downloaded using your file manager, and right click




Select “Open with “Software Install” if this doesn't appear you skipped step 5.

Next screen is the install screen, click the blue “Install” button

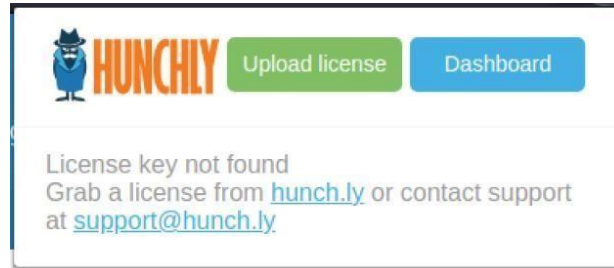
Once it says installed you can close the window by the right x



11. Once you close out the installer, you will need to install (if you already haven't) the Chrome extension for Hunchly from step 9.

12. Next "Open Google Chrome" and select the "Extensions"  icon,

13. You can click the Red Hunchly  2.0 icon and a window with licensing will appear -
License Key not found



14. Select "Upload License" and another popup with "license status" and an option to upload a license.
Select the green upload license and browse to locate the license file under downloads - Select that file and "okay"



15. Once your product has been licensed, you will see the license status change to green



16. Now lets pin the hunchly icon to your toolbar so you don't need to search for it under extensions again, go to extensions, and notice the thumb-tac icon next to Hunchly 2.0



Select that icon and it should now appear on your main browser window

17. Next let's fix the Kali linux issue with launching Hunchly since Kali main OS is a bit different then Ubuntu we have to edit the **“/usr/lib/hunchly”** parameter file
- a. In a terminal window, sudo nano (or view) /usr/lib/hunchly

```
GNU nano 5.3 hunchly
#!/bin/sh
# Looks like you're up to something. We want to help! Contact us at support>

#launch that 'lil detective!
exec /usr/lib/hunchly/Hunchly --no-sandbox
```

*We will add a space and then **--no-sandbox** (as shown) and then save the file.*

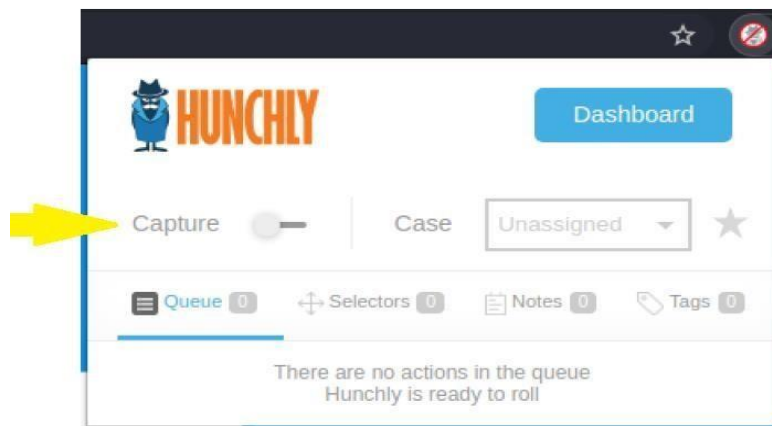
LAB2: Start an Investigation

18. First launch Tor Browser (Required to proxy the Chrome Browser to Tor)

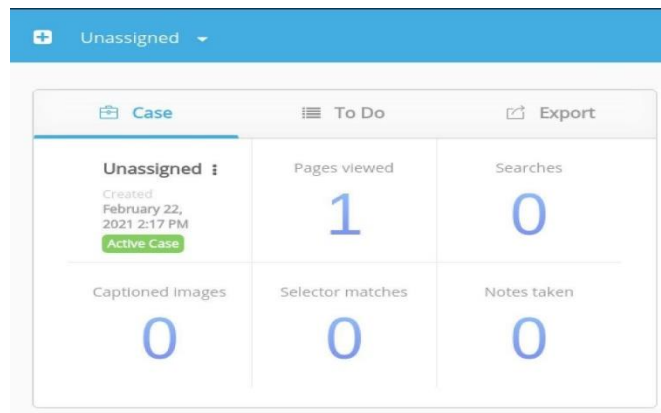
19. Next enter the line below in a terminal window to launch Chrome

```
google-chrome --proxy-server="socks5://localhost:9150" --host-  
resolVERRULES="MAP * ~NOTFOUND , EXCLUDE localhost"/Applications/Google\  
Chrome.app/ Contents/MacOS/Google\ Chrome --proxysERVER="socks5://localhost:9150" -  
-hostresolVERRULES="MAP * ~NOTFOUND , EXCLUDE localhost"
```

20. Next Lets start a new case - Select the Hunchly icon from your Chrome Browser 21. Select Capture to to on (blue) then



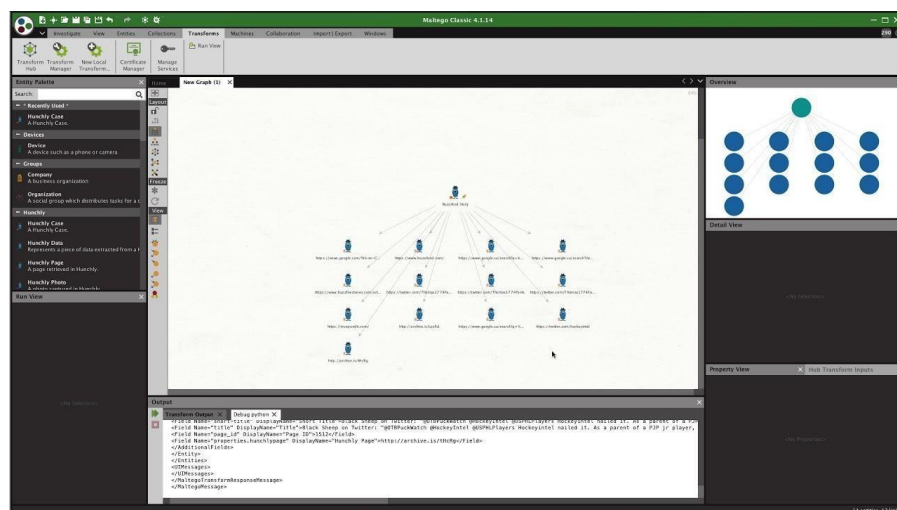
22. Now you will browse to your questionable site, and review the items you are investigating while Hunchly captures all in the background.
23. To launch the Hunchly (another Quark of Kali) from a terminal window type in hunchly - enter. You should see Hunchly dashboard come up, as shown below.



You should see one or more page views, and one unassigned active case, you can review all the details of the case, browse at the bottom section, or add details of what was found in notes.

Maltego Integration:

24. There is an integration for **Maltego** transforms with Hunchly and data forwarding.
25. You can sign up for the Hunchly mailing list where they send a daily “hidden services” spreadsheet each day, it includes a ton of information and can help you with your investigation.
26. If you want to continue with learning the in’s and outs of hunchly which is way beyond what this class teaches - checkout [Hunchly support](#)



LAB 20 I2P and Garlic

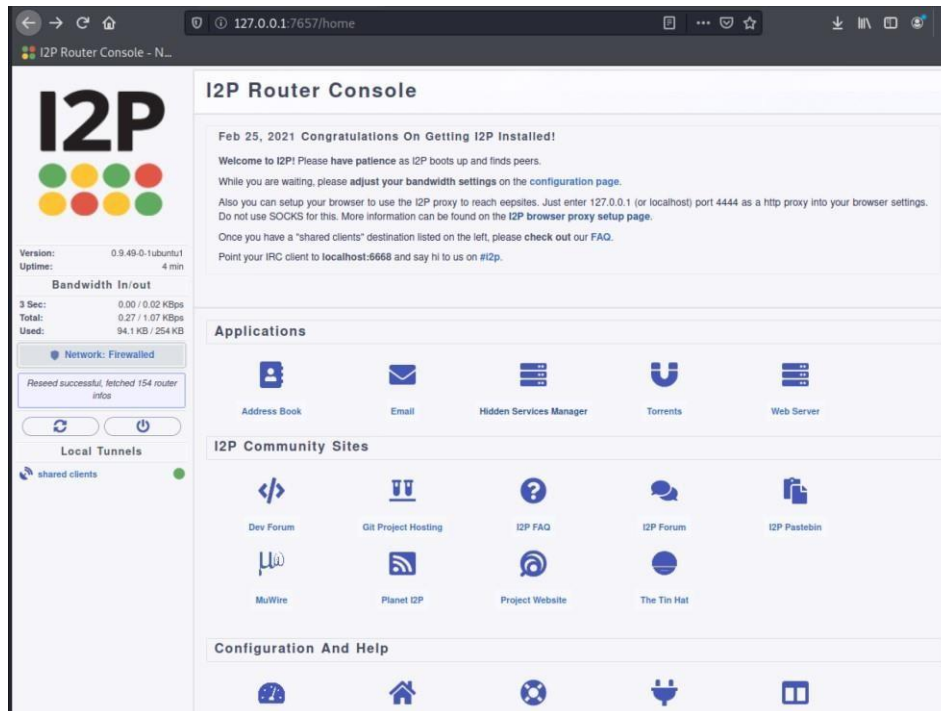
This is an optional exercise to get i2P running on your system, there are quite a few debian apps and commands required to get it up and running, and of course do not run as root after you finished installing.

1. From your Debian VM Host open firefox and Download <https://geti2p.net/en/download#deb>
2. Make sure your Tor demon is stopped - > service tor stop
3. > sudo apt-get install apt-transport-https curl

4. Create /etc/apt/sources.list.d/i2p.list (use nano or view) and add the following repos deb https://deb.i2p2.de/ buster main
deb-src https://deb.i2p2.de/ buster main
5. curl -o i2p-debian-repo.key.asc https://geti2p.net/_static/i2p-debian-repo.key.asc
6. gpg -n --import --import-options import-show i2p-debian-repo.key.asc
7. sudo apt-key add i2p-debian-repo.key.asc
8. sudo apt-get update
9. sudo apt-get install i2p i2p-keyring

Now from a web browser: <http://127.0.0.1:7657/welcome>

It will test your bandwidth, give it a minute, you wont be able to click next until its finished.



Make sure you setup your browser for the proxy, settings can be found here

Go to Preferences, Network proxy : In the Connection Settings pop-up, select Manual proxy configuration. Set both the HTTP and SSL Proxy to address 127.0.0.1 with port 4444 as shown in the following screenshot.