

FINAL PROJECT

PKI, Proxy & DMZ

HAMZA RAMZA Hamza.ramzan@epita.fr

Contents

1.	Local domain Setup and SSL/TLS Certificate:.....	2
1.1	Local domain setup:.....	2
1.2	Setup SSL/TLS Certificate	5
2.	ELK Stack Implementation and Apache Log Ingestion:.....	7
2.1	Install the ELK Stack:	7
2.2	Configure Logstash to Process Apache Logs:.....	8
2.3	Start Logstash with the New Configuration:.....	8
2.4	Visualize web traffic in a Kibana dashboard:	9
3.	Integrating Squid Proxy with ELK:	10
3.1	Install and setup squid	10
3.2	Integration with ELK:.....	10
4.	DMZ Setup.....	11
4.1	Setup iptables	11
4.2	Flush existing rules and set default policy to drop	11
4.3	Allow Traffic on Loopback Interface	12
4.4	Allow Outgoing Traffic on the External Interface:	12
4.5	Allow Incoming HTTP and HTTPS Traffic on the External Interface.....	13
4.6	Save the iptables Configuration.....	13

1. Local domain Setup and SSL/TLS Certificate:

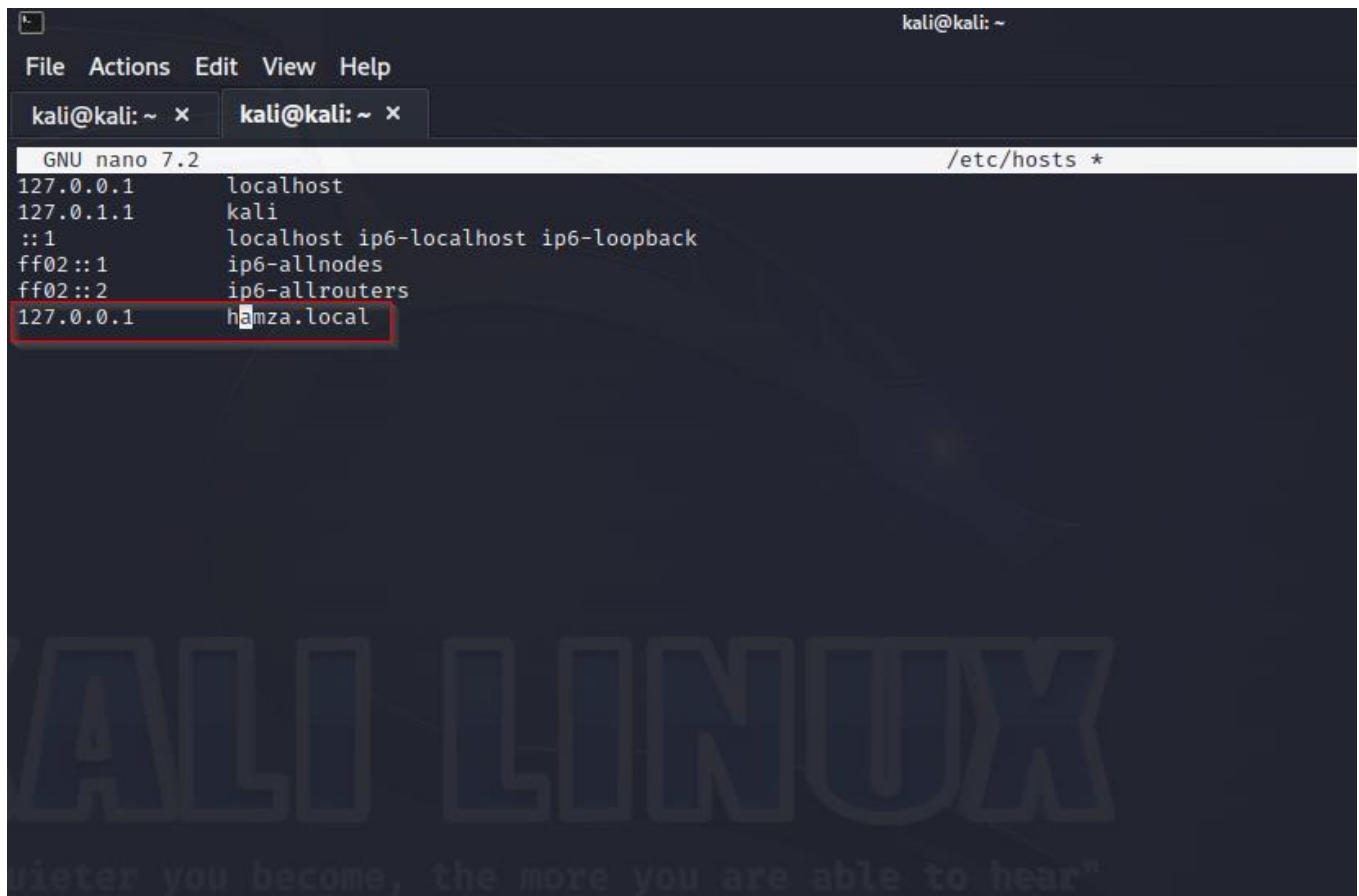
In this section we established an identity on the web and secured it using x509 certificate. Setup a local domain, apache server and the setup SSL/TLS (x509) certificate to secure it on web.

1.1 Local domain setup:

1.1.1 Local domain setup

Edit **/etc/hosts** file on our system and add hamza.local, is hostname mapping to 127.0.0.1 . This is loopback address.

Command: *Sudo nano /etc/hosts*



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x  
GNU nano 7.2 /etc/hosts *  
127.0.0.1 localhost  
127.0.1.1 kali  
::1 localhost ip6-localhost ip6-loopback  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
127.0.0.1 hamza.local  
KALI LINUX  
quieter you become, the more you are able to hear"
```

1.1.2 Installed webserver:

Sudo apt-get install apache2: Used this command to install web server apache2. This helps us to create a local server and able us to host a website. In our case that is hamza.local

```
kali@kali: ~ x kali@kali: ~ x
(kali@kali)~[~]
$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following packages will be upgraded:
  apache2 apache2-bin apache2-data apache2-utils
4 upgraded, 0 newly installed, 0 to remove and 1268 not upgraded.
Need to get 1,939 kB of archives.
After this operation, 1,024 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive-4.kali.org/kali kali-rolling/main amd64 apache2 amd64 2.4.57-3 [214 kB]
Get:2 http://archive-4.kali.org/kali kali-rolling/main amd64 apache2-bin amd64 2.4.57-3 [1,363 kB]
Get:3 http://archive-4.kali.org/kali kali-rolling/main amd64 apache2-data all 2.4.57-3 [160 kB]
Get:4 http://archive-4.kali.org/kali kali-rolling/main amd64 apache2-utils amd64 2.4.57-3 [202 kB]
Fetched 1,939 kB in 3s (744 kB/s)
(Reading database ... 408904 files and directories currently installed.)
Preparing to unpack .../apache2_2.4.57-3_amd64.deb ...
Unpacking apache2 (2.4.57-3) over (2.4.57-2) ...
Preparing to unpack .../apache2-bin_2.4.57-3_amd64.deb ...
Unpacking apache2-bin (2.4.57-3) over (2.4.57-2) ...
Preparing to unpack .../apache2-data_2.4.57-3_all.deb ...
Unpacking apache2-data (2.4.57-3) over (2.4.57-2) ...
```

1.1.3 Configuring Webserver for hamza.local

sudo nano /etc/apache2/sites-available/hamza.local.conf : Used this command to configure our server to host pur local domain. Add following instruction in hamza.local.conf

```
kali@kali: ~ x kali@kali: ~ x
GNU nano 7.2 /etc/apache2/sites-available/
<VirtualHost *:80>
  ServerName hamza.local
  DocumentRoot /var/www/hamza.local
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

1.1.4 Create Root Directory

Created a root directory in **/var/www/** this directory can be used to store web files for a website or web application. In our case it is hamza.local

```
(kali㉿kali)-[~]
$ sudo mkdir /var/www/hamza.local

(kali㉿kali)-[~]
$ ls /var/www/
hamza.local  html

(kali㉿kali)-[~]
$
```

1.1.5 Enable site

sudo a2ensite hamza.local.conf: this command used to enable our local site. And restart apache2.

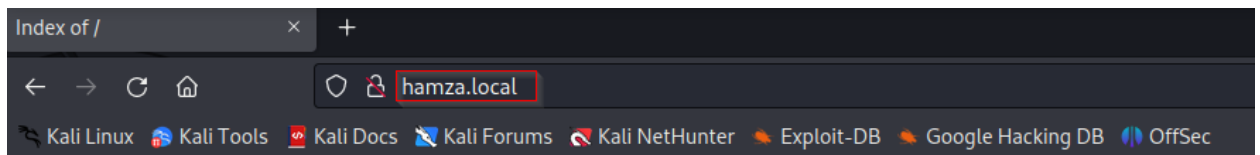
```
(kali㉿kali)-[~]
$ sudo a2ensite hamza.local.conf
Enabling site hamza.local.
To activate the new configuration, you need to run:
systemctl reload apache2

(kali㉿kali)-[~]
$ sudo systemctl restart apache2

(kali㉿kali)-[~]
$
```

1.1.6 Local domain setup Testing

Setup our server and add our local domain. Now we can access hamza.local on browser.



Index of /

Name	Last modified	Size	Description
------	---------------	------	-------------

Apache/2.4.57 (Debian) Server at hamza.local Port 80

1.2 Setup SSL/TLS Certificate

In this section generated SSL certificate and private key to secure our local domain. It is visible in previous testing phase, we can access but it's not secure. Before SSL domain address is **http://hamza.local**

1.2.1 Generate SSL certificate and private key

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
etc/ssl/private/hamza.local.key -out /etc/ssl/certs/hamza.local.crt
```

Generated a private key and also x509 certificate.

```
(kali㉿kali)-[~]
└─$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/hamza.local.key -out /etc/ssl/certs/hamza.local.crt
```

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]:FR
State or Province Name (full name) [Some-State]:ILD
Locality Name (eg, city) []:Argenteuil
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Epita
Organizational Unit Name (eg, section) []:EPIAT1
Common Name (e.g. server FQDN or YOUR name) []:hamza.local
Email Address []:hamzaramzan081@gmail.com

1.2.2 Set Permissions

sudo chmod 600 /etc/ssl/private/hamza.local.key: Used this command to set permission to read and write for hamza.local.key only for root.

```
(kali㉿kali)-[~]  
$ sudo chmod 600 /etc/ssl/private/hamza.local.key
```

1.2.3 Create Apache SSL Virtual Host Configuration

`sudo vim /etc/apache2/sites-available/hamza.local-ssl.conf`

Configure Virtual host configuration and add following code. This code configure HTTPS on our local domain

```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
GNU nano 7.2 /etc/apache2/sites-available/hamza.local-ssl.conf *
<!--
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
ServerAdmin webmaster@hamza.local
ServerName salman.local
DocumentRoot /var/www/hamza.local
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined
SSLEngine on
SSLCertificateFile /etc/ssl/certs/hamza.local.crt
SSLCertificateKeyFile /etc/ssl/private/hamza.local.key
<FilesMatch "\.(cgi|shtml|phtml|php)$">
SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
SSLOptions +StdEnvVars
</Directory>
</VirtualHost>
</IfModule>
```

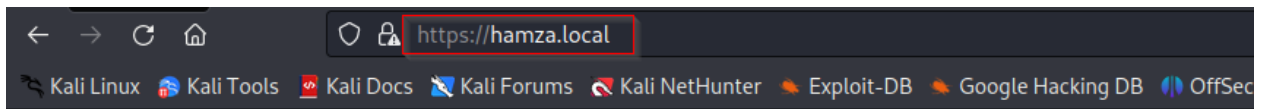
1.2.4 Enable the SSL site

```
(kali@kali)-[~]
$ sudo a2ensite hamza.local-ssl.conf
Enabling site hamza.local-ssl.
To activate the new configuration, you need to run:
systemctl reload apache2

(kali@kali)-[~]
$ sudo systemctl restart apache2
```

1.2.5 Test SSL/TLS

SSL configured.



Index of /

Name	Last modified	Size	Description
----------------------	-------------------------------	----------------------	-----------------------------

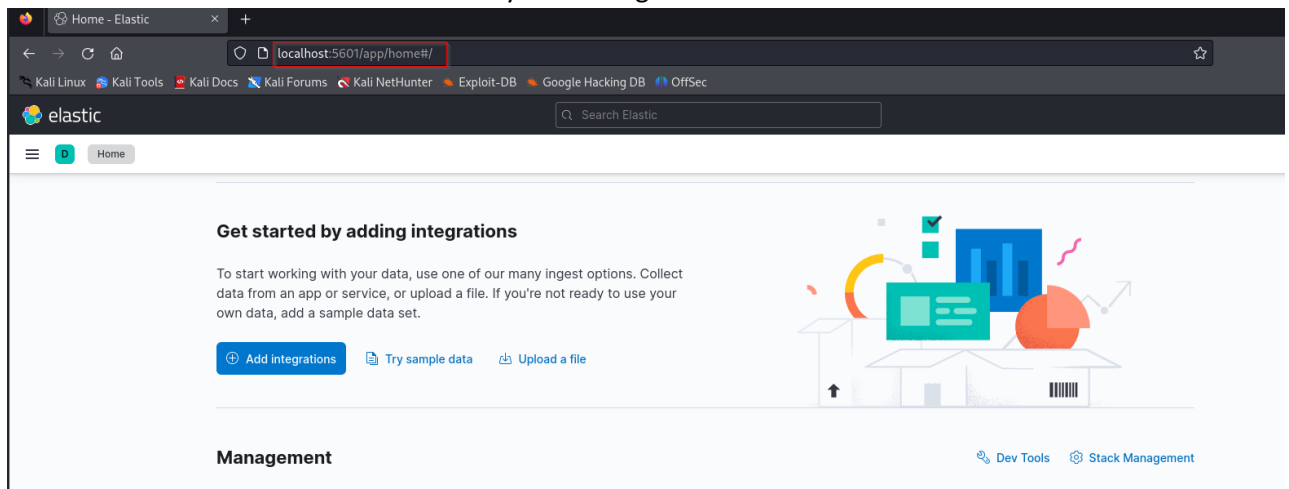
<hr/>			
<i>Apache/2.4.57 (Debian) Server at hamza.local Port 443</i>			

2. ELK Stack Implementation and Apache Log Ingestion:

In this section we configure ELK (Elastic search, Logstash and Kibana).

2.1 Install the ELK Stack:

In this section we installed ELK successfully and configured kibana dashboard on **localhost:5601**



2.2 Configure Logstash to Process Apache Logs:

`sudo vim /etc/logstash/conf.d/apache.conf`

The command is used to open the Apache configuration file for Logstash.

```
File Actions Edit View Help
GNU nano 7.2 /etc/logstash/conf.d/apache.conf
input {
  file {
    path => "/var/log/apache2/access.log"
    start_position => "beginning"
    sincedb_path => "/dev/null"
  }
}
filter {
  if [path] =~ "access" {
    mutate { replace => { "type" => "apache_access" } }
    grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
    date {
      match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]
    }
  }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
  }
}
```

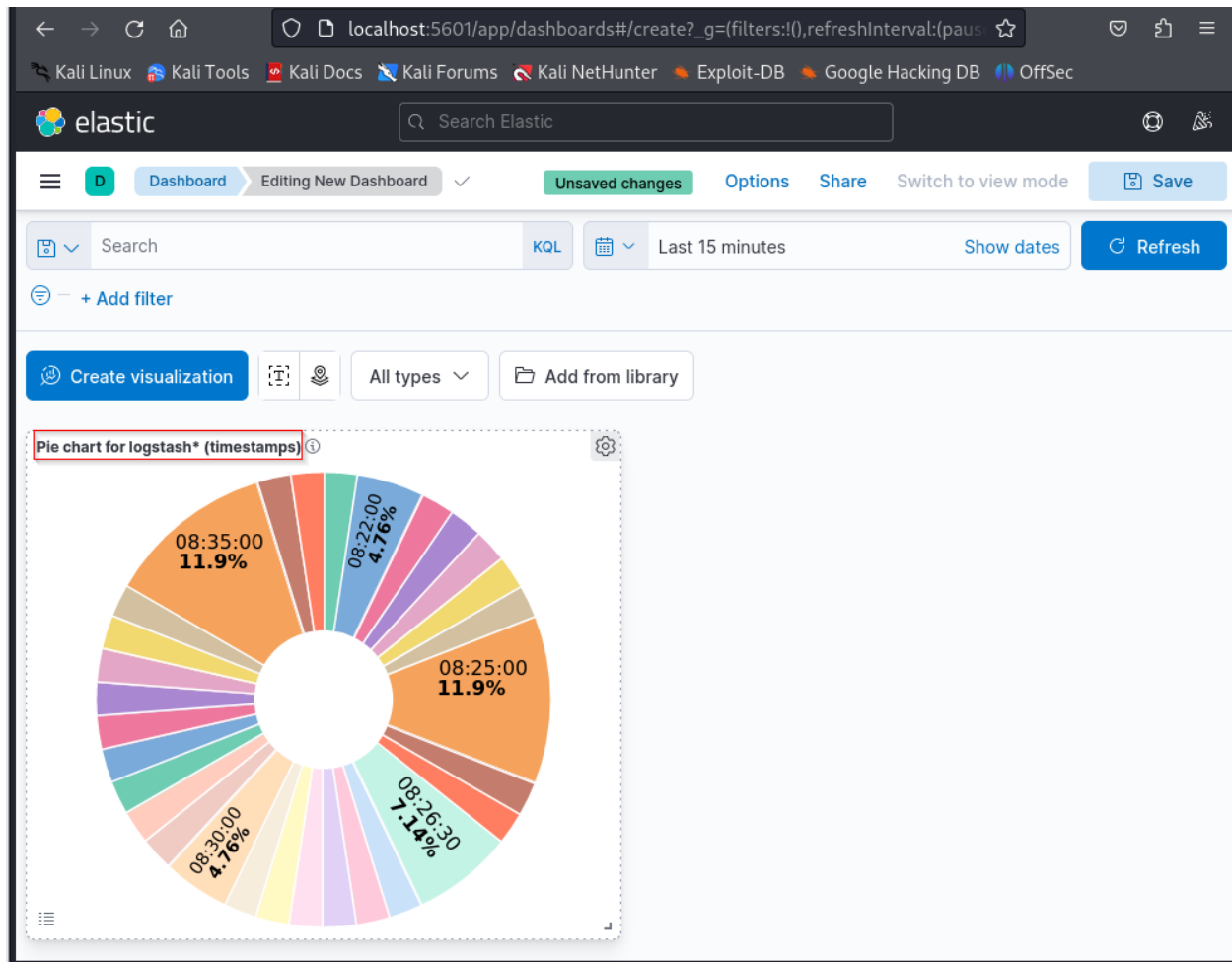
2.3 Start Logstash with the New Configuration:

`Sudo service logstash start`

```
(kali㉿kali)-[~]
$ sudo service logstash start
(kali㉿kali)-[~]
$
```

2.4 Visualize web traffic in a Kibana dashboard:

Setup kibana dashboard on <http://localhost:5601> and create some visualization for logs.



3. Integrating Squid Proxy with ELK:

In this section we will configure proxy server squid.

3.1 Install and setup squid

Installed squid by using ***sudo apt-get install squid***

Configure squid using ***sudo vim /etc/squid/squid.conf*** and add following code.

```
http_access allow localnet
```

```
http_access allow localhost
```

```
http_port 3128
```

Mostly this setting comes with installation of squid. After installation and configuration start squid.

```
(kali@kali)-[~]
$ sudo systemctl start squid

(kali@kali)-[~]
$ sudo systemctl restart squid

(kali@kali)-[~]
```

3.2 Integration with ELK:

Created a new logstash pipeline to process Squid logs.

sudo vim /etc/logstash/conf.d/squid.conf used this command to edit conf file.

```
kali@kali: ~
File Actions Edit View Help
GNU nano 7.2 /etc/logstash/conf.d/squid.conf *
input {
  file {
    path => "/var/log/squid/access.log"
    start_position => "beginning"
    sincedb_path => "/dev/null"
    type => "squid_log"
  }
}
filter {
  if [type] == "squid_log" {
    grok {
      Page 5 of 6
      match => { "message" => "%{NUMBER:timestamp} %{NUMBER:
%{INT:response_time} %{IP:src_ip}
%{WORD:squid_request_status}/%{NUMBER:http_status_code}
%{NUMBER:reply_size} %{WORD:http_method} %{URI:requested_url}
%{USERNAME:user} %{WORD:squid_hierarchy_status}/%{IP:dst_ip}" }
    }
    date {
      match => [ "timestamp", "UNIX" ]
    }
  }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
  }
}
```

4. DMZ Setup

4.1 Setup iptables

sudo apt-get install iptables

```
(kali㉿kali)-[~]
$ sudo apt-get update
Hit:1 http://archive-4.kali.org/kali kali-rolling InRelease
Hit:2 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Reading package lists... Done
W: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: Key is stored in legacy t
d.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.

(kali㉿kali)-[~]
$ sudo apt-get install iptables
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables is already the newest version (1.8.9-2).
iptables set to manually installed.
The following packages were automatically installed and are no longer required:
gir1.2-gtksource-3.0 gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-webkit2-4.0
gobject-introspection king-phisher libarmadillo11 libblockdev-crypto2 libblockdev-fs2
libblockdev-loop2 libblockdev-part-err2 libblockdev-part2 libblockdev-swap2 libblockdev-utils2
libblockdev2 libgdal32 libgeos3.11.1 libgumbo1 libgupnp-igd-1.0-4 libjim0.81 liblc3-0
libmongocrypt0 libmuj2 libncurses5 libnfs13 libobjc-12-dev libsoup-gnome2.4-1 libspatialite7
libsuperlu5 libtexluaajit2 libtinfo5 libwebsockets17 libyara9 pipewire-alsa pwgen
python3-advancedhttpserver python3-boltons python3-cairo-dev python3-cryptography37
python3-flask-security python3-geoip2 python3-geojson python3-graphene python3-graphene-sqlalche
python3-graphql-core python3-graphql-relay python3-icalendar python3-jaraco.classes python3-jdca
python3-maxminddb python3-promise python3-py python3-pytz-deprecation-shim python3-rule-engine
python3-rx python3-smoke-zephyr python3-texttable tftp
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
```

4.2 Flush existing rules and set default policy to drop

```
(kali㉿kali)-[~]
$ sudo iptables -F

(kali㉿kali)-[~]
$ sudo iptables -P INPUT DROP

(kali㉿kali)-[~]
$ sudo iptables -P FORWARD DROP

(kali㉿kali)-[~]
$ sudo iptables -P OUTPUT DROP
```

4.3 Allow Traffic on Loopback Interface

```
(kali㉿kali)-[~]  
$ sudo iptables -A INPUT -i lo -j ACCEPT  
  
(kali㉿kali)-[~]  
$ sudo iptables -A OUTPUT -o lo -j ACCEPT
```

4.4 Allow Outgoing Traffic on the External Interface:

```
(kali㉿kali)-[~]  
$ sudo iptables -A OUTPUT -o EXTERNAL_INTERFACE_NAME -j ACCEPT  
  
iptables v1.8.9 (nf_tables): interface name 'EXTERNAL_INTERFACE_NAME' must be shorter than IFNAMSIZ (15  
)  
Try 'iptables -h' or 'iptables --help' for more information.  
  
(kali㉿kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.105 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::8680:3faf:58cb:f22d prefixlen 64 scopeid 0x20<link>  
    inet6 2a01:e0a:415:1420:9f8d:6899:cf79:681f prefixlen 64 scopeid 0x0<global>  
    ether 00:0c:29:a5:40:ed txqueuelen 1000 (Ethernet)  
    RX packets 17820 bytes 18006623 (17.1 MiB)  
    RX errors 0 dropped 23 overruns 0 frame 0  
    TX packets 2413 bytes 354295 (345.9 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 891147 bytes 147973688 (141.1 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 891147 bytes 147973688 (141.1 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
(kali㉿kali)-[~]  
$ sudo iptables -A OUTPUT -o eth0 -j ACCEPT
```

4.5 Allow Incoming HTTP and HTTPS Traffic on the External Interface

```
(kali㉿kali)-[~]  
$ sudo iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT  
  
(kali㉿kali)-[~]  
$ sudo iptables -A INPUT -i eth0 -p tcp --dport 443 -j ACCEPT  
  
(kali㉿kali)-[~]  
$
```

4.6 Save the iptables Configuration

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali㉿kali)-[~]  
$ sudo iptables-save |sudo tee /etc/iptables/rules.v4  
tee: /etc/iptables/rules.v4: No such file or directory  
# Generated by iptables-save v1.8.9 (nf_tables) on Sun Oct  1 10:05:05 2023  
*filter  
:INPUT DROP [805:57102]  
:FORWARD DROP [0:0]  
:OUTPUT DROP [218:17010]  
-A INPUT -i lo -j ACCEPT  
-A INPUT -i eth0 -p tcp -m tcp --dport 80 -j ACCEPT  
-A INPUT -i eth0 -p tcp -m tcp --dport 443 -j ACCEPT  
-A OUTPUT -o lo -j ACCEPT  
-A OUTPUT -o eth0 -j ACCEPT  
COMMIT  
# Completed on Sun Oct  1 10:05:05 2023  
  
(kali㉿kali)-[~]  
$
```

