Detailed Report: Local Domain Setup and SSL/TLS Certificate

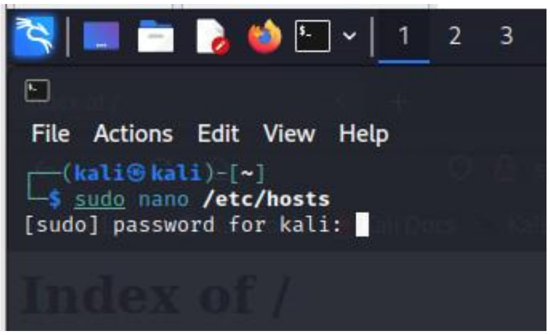
Purpose:

The purpose of this setup is to establish a local web domain (inaam.local) on an Apache web server and secure it with an SSL/TLS certificate using X.509 encryption. This report will detail the steps taken, configurations made, and their implications.

Local Domain Setup:

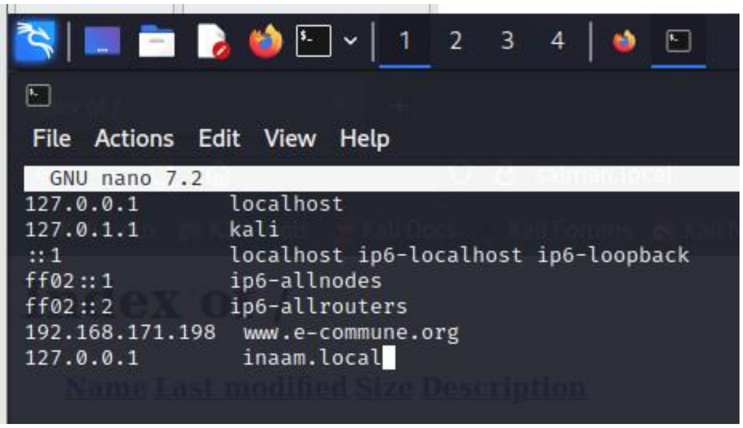
Step 1: Provision Local Domain Entry

- Command:



```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/hosts
[sudo] password for kali:
```

- Added the following entry: **127.0.0.1 inaam.local**

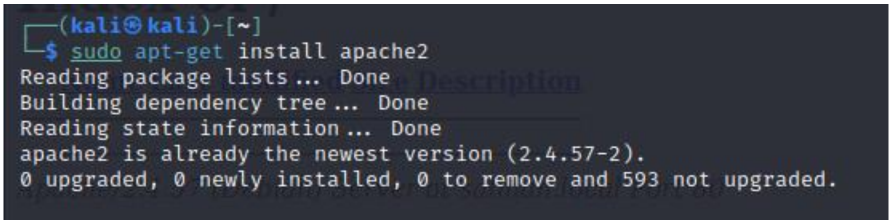


```
 GNU nano 7.2
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
192.168.171.198  www.e-commune.org
127.0.0.1        inaam.local
```

- Implication: This entry maps the domain name 'inaam.local' to the localhost IP address (127.0.0.1).

Step 2: Install Apache Web Server

- Command:

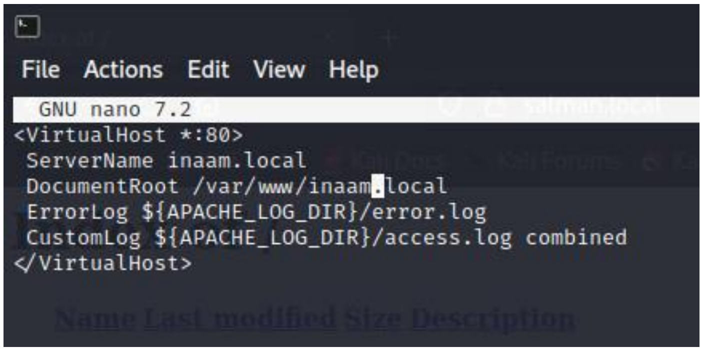


```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.57-2).
0 upgraded, 0 newly installed, 0 to remove and 593 not upgraded.
```

- Implication: Apache web server is installed, allowing us to host web content locally.

Step 3: Configure Web Server for inaam.local

- Command: **sudo vim /etc/apache2/sites-available/inaam.local.conf**
- Configuration:

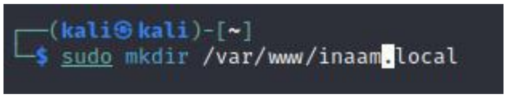


```
  GNU nano 7.2
<VirtualHost *:80>
 ServerName inaam.local
 DocumentRoot /var/www/inaam.local
 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- Implication: This configuration sets up a virtual host for 'inaam.local,' specifying the document root and log file locations.

Step 4: Create Root Directory

- Command:

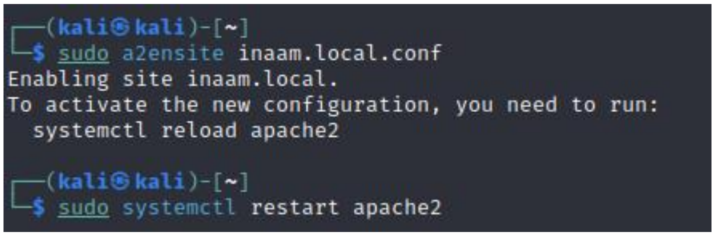


```
┌──(kali㉿kali)-[~]
└─$ sudo mkdir /var/www/inaam.local
```

- Implication: The root directory for the 'inaam.local' website is created.

Step 5: Enable the Site and Restart Apache

- Commands:

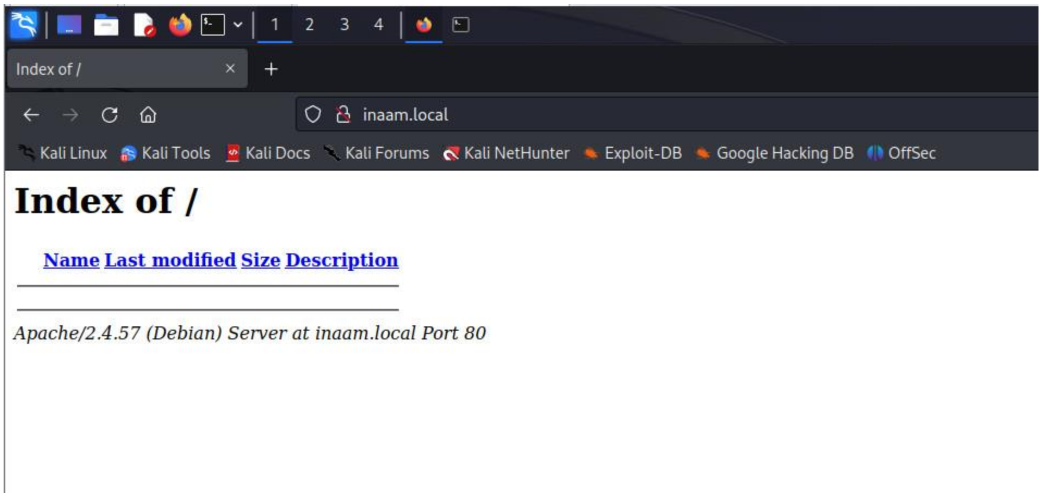


```
┌──(kali㉿kali)-[~]
└─$ sudo a2ensite inaam.local.conf
Enabling site inaam.local.
To activate the new configuration, you need to run:
  systemctl reload apache2

┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart apache2
```

- Implication: The 'inaam.local' site is enabled, and Apache is restarted to apply the changes.

Step 6: Test Local Domain Setup

- URL: http://inaam.local



Index of /

Name Last modified Size Description

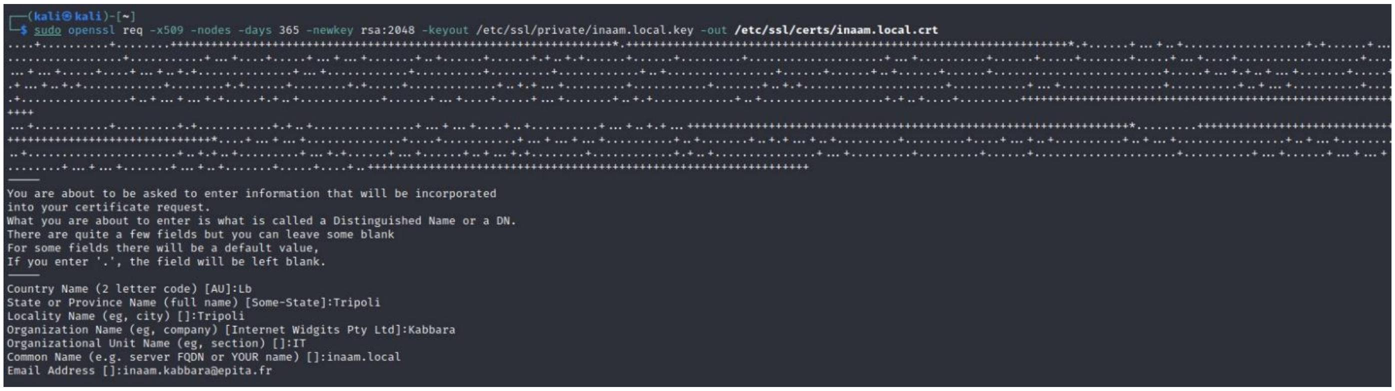
Apache/2.4.57 (Debian) Server at inaam.local Port 80

- Implication: Accessing this URL should display content from the '/var/www/inaam.local' directory.

SSL/TLS Certificate Setup:

Step 1: Generate the SSL Certificate and Private Key

- Command:

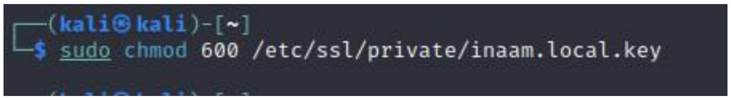


```
┌──(kali㉿kali)-[~]
└─$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/inaam.local.key -out /etc/ssl/certs/inaam.local.crt
....+.........+........+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+......+...+..+..................+.+......+...
.................+...........+...+....+.....+...+...+.......+..+......+......+.+..+.+......+......+..........+.......+....................+...+.......................+...+.........+......+.....+.......+.....+...+....+.........................+...
...+...+.....+....+...+..+.+..............+...+.............+..........+.........+...........+..+................+......+..............+......+......+..+......+......+..........................+.....+...+.+..+...+.......+.....+
.+...+..+.+............+........+.+......+........+.+.....+............+..+.+...+.........+..........+.......+..+.+..........+..+.+.......+.......................+............+...+................+..........+..+...+...........+...
.+...............+..+...+...+.+.....+.+..+............+......+...+....+.....+...+.......+..+.+.............+..+.........+..+.....................+.+..+....+..........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++
...+...........+.........+.+............+.+..+...............+...+...+....+..+..........+...+..+.+...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*........+++++++++++++++++++++++++++
+++++++++++++++++++++++++++++*....+...+...+...............+....+............+...+...+...+.........+..+........+..+.+...+..+.......+..+.+...+..+.........+.........+....+...+..+..........+..+...+...................+..+...+......
..+......................+..+.+..+.........+...+.+......+...+......+..+...+.+........+.................+.+..+...............+...+..+...............+...+..........+.........+......+......................+..........+...+.......+...+...+
........+...+...+.......+...+..+.......+.....+....+..+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
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]:Lb
State or Province Name (full name) [Some-State]:Tripoli
Locality Name (eg, city) []:Tripoli
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Kabbara
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:inaam.local
Email Address []:inaam.kabbara@epita.fr
```

- Implication: A self-signed SSL certificate and private key are generated for 'inaam.local,' with a validity period of 365 days.

Step 2: Set Permissions

- Command:

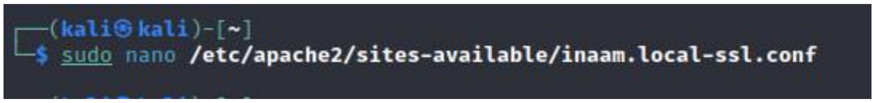


```
┌──(kali㉿kali)-[~]
└─$ sudo chmod 600 /etc/ssl/private/inaam.local.key
```

- Implication: Restricts access to the private key, ensuring its security.

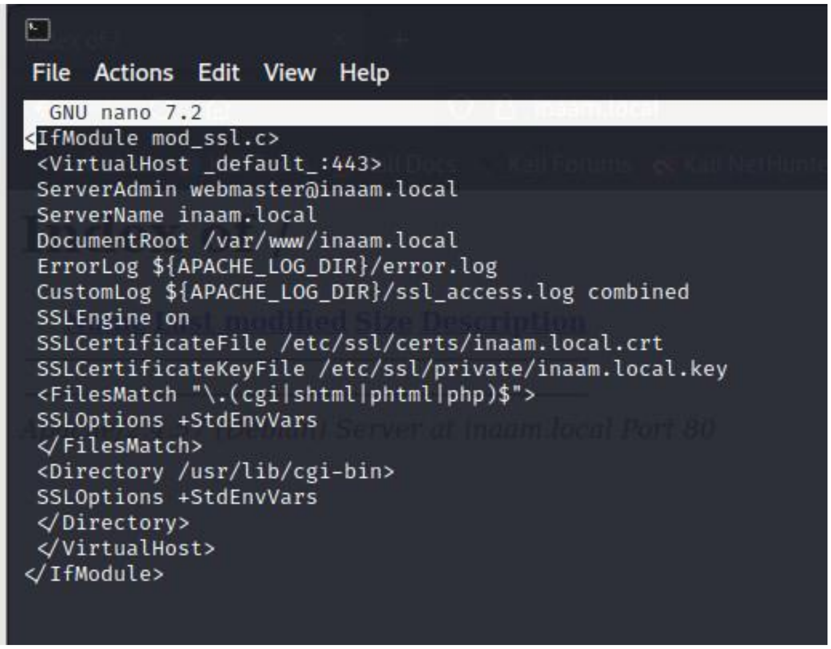
Step 3: Create Apache SSL Virtual Host Configuration

- Command:



```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/apache2/sites-available/inaam.local-ssl.conf
```

- Configuration:

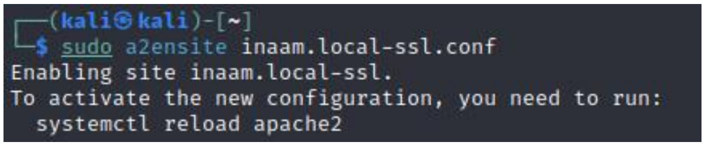


```
 GNU nano 7.2
<IfModule mod_ssl.c>
 <VirtualHost _default_:443>
 ServerAdmin webmaster@inaam.local
 ServerName inaam.local
 DocumentRoot /var/www/inaam.local
 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined
 SSLEngine on
 SSLCertificateFile /etc/ssl/certs/inaam.local.crt
 SSLCertificateKeyFile /etc/ssl/private/inaam.local.key
 <FilesMatch "\.(cgi|shtml|phtml|php)$">
 SSLOptions +StdEnvVars
 </FilesMatch>
 <Directory /usr/lib/cgi-bin>
 SSLOptions +StdEnvVars
 </Directory>
 </VirtualHost>
</IfModule>
```

- Implication: This configuration sets up an SSL-enabled virtual host for 'inaam.local,' specifying SSL certificate and key locations.

Step 4: Enable the SSL Site

- Command:

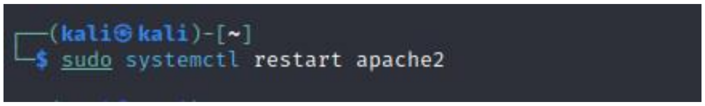


```
┌──(kali㉿kali)-[~]
└─$ sudo a2ensite inaam.local-ssl.conf
Enabling site inaam.local-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

- Implication: The SSL site for 'inaam.local' is enabled.

Step 5: Restart Apache

- Command:

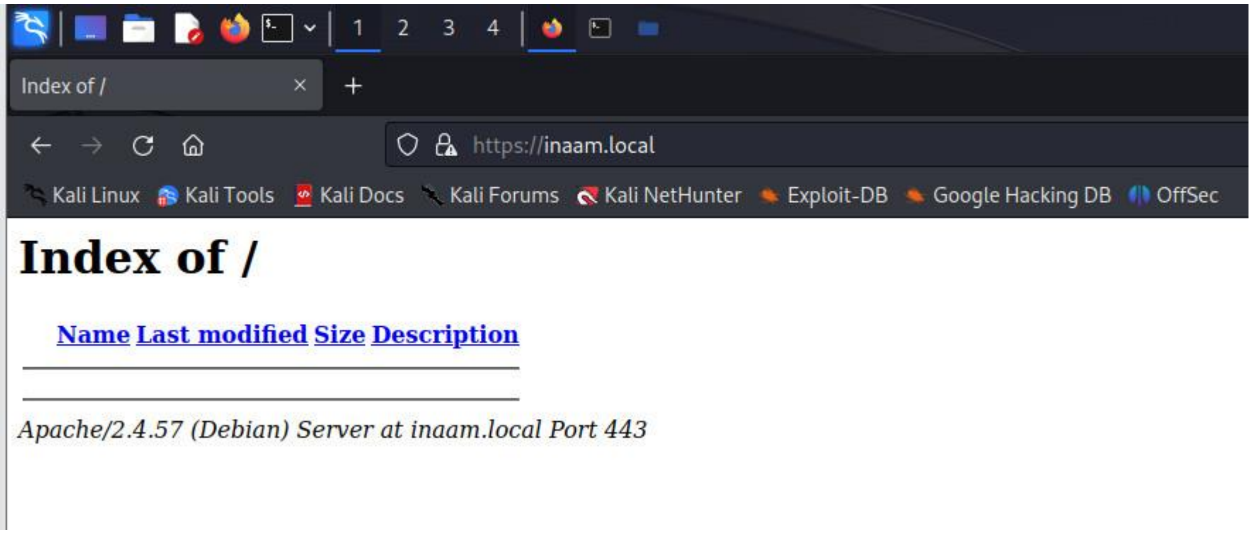


```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart apache2
```

- Implication: Apache is restarted to apply the SSL configuration.

Step 6: Test SSL/TLS Setup on Local Domain

- URL: https://inaam.local



Index of /

Name Last modified Size Description

Apache/2.4.57 (Debian) Server at inaam.local Port 443

- Implication: Accessing this URL should secure the connection with the self-signed SSL certificate.

ELK Stack Implementation and Apache Log Ingestion

Purpose:

The purpose of this implementation is to set up the ELK (Elasticsearch, Logstash, and Kibana) Stack to monitor web server activity, specifically Apache logs. The ELK Stack provides a powerful platform for log aggregation, analysis, and visualization.

Step 1: Install the ELK Stack (Assuming Elasticsearch, Logstash, and Kibana are installed):

- Implication: The ELK Stack provides the necessary components for log processing, storage, and visualization.

Step 2: Configure Logstash to Process Apache Logs:

Substep 2.1: Create a New Logstash Pipeline

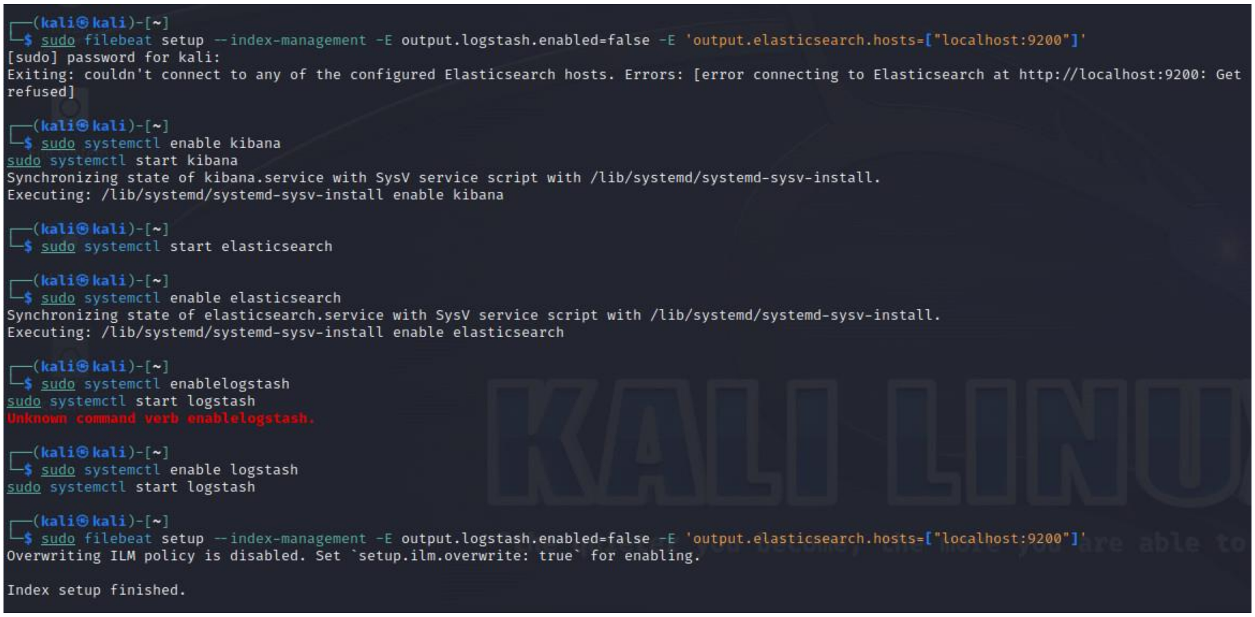
- Command: **sudo vim /etc/logstash/conf.d/apache.conf**

- Configuration:
- Implication: This Logstash configuration specifies the Apache access log file as a source, processes log entries using the grok plugin, and sends the structured logs to Elasticsearch.

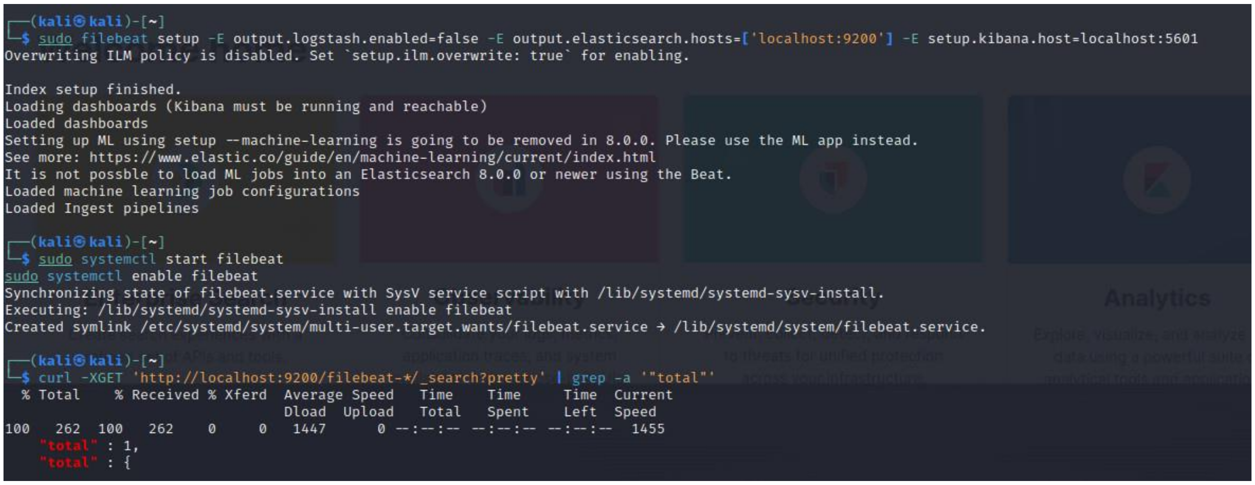
Step 3: Start Logstash with the New Configuration:

- Command: **sudo service logstash start**

- Implication: Logstash is started with the new configuration to begin processing Apache logs and sending them to Elasticsearch.



```
┌──(kali㉿kali)-[~]
└─$ sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["localhost:9200"]'
[sudo] password for kali:
Exiting: couldn't connect to any of the configured Elasticsearch hosts. Errors: [error connecting to Elasticsearch at http://localhost:9200: Get
refused]

┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable kibana
sudo systemctl start kibana
Synchronizing state of kibana.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana

┌──(kali㉿kali)-[~]
└─$ sudo systemctl start elasticsearch

┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch

┌──(kali㉿kali)-[~]
└─$ sudo systemctl enablelogstash
sudo systemctl start logstash
Unknown command verb enablelogstash.

┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable logstash
sudo systemctl start logstash

┌──(kali㉿kali)-[~]
└─$ sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["localhost:9200"]'
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
```

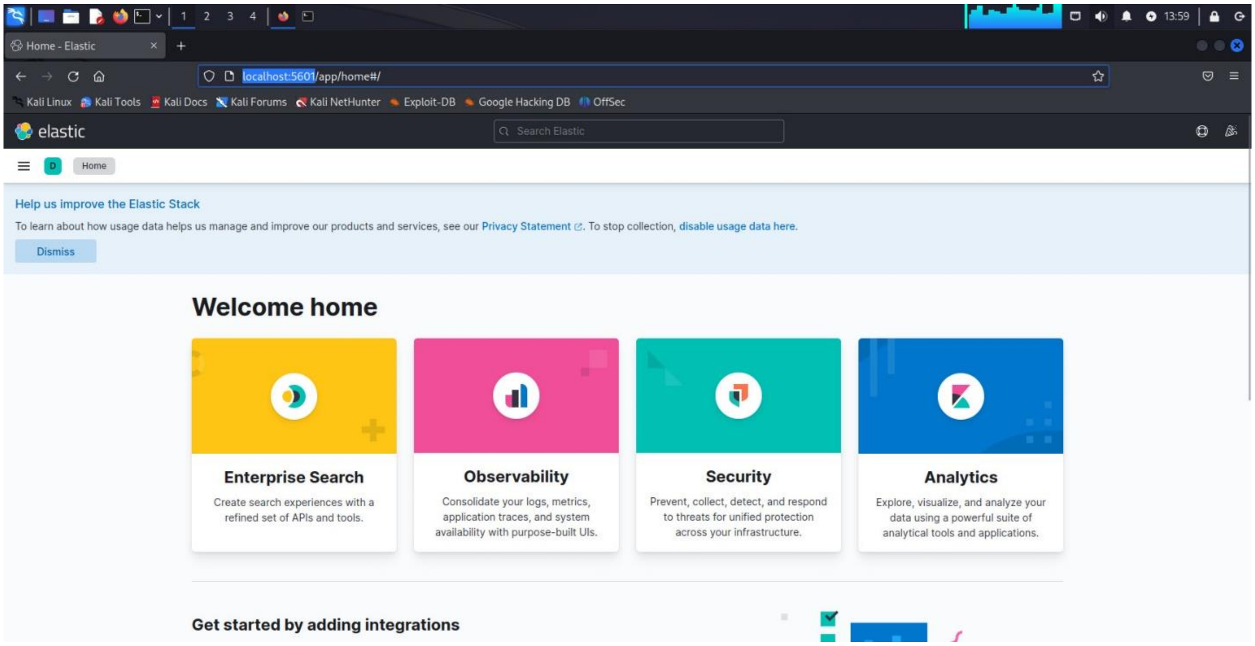


```
┌──(kali㉿kali)-[~]
└─$ sudo filebeat setup -E output.logstash.enabled=false -E output.elasticsearch.hosts=['localhost:9200'] -E setup.kibana.host=localhost:5601
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Setting up ML using setup --machine-learning is going to be removed in 8.0.0. Please use the ML app instead.
See more: https://www.elastic.co/guide/en/machine-learning/current/index.html
It is not possble to load ML jobs into an Elasticsearch 8.0.0 or newer using the Beat.
Loaded machine learning job configurations
Loaded Ingest pipelines

┌──(kali㉿kali)-[~]
└─$ sudo systemctl start filebeat
sudo systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.

┌──(kali㉿kali)-[~]
└─$ curl -XGET 'http://localhost:9200/filebeat-*/_search?pretty' | grep -a '"total"'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   262  100   262    0     0   1447      0 --:--:-- --:--:-- --:--:--  1455
    "total" : 1,
    "total" : {
```

Step 4: Visualize Web Traffic in a Kibana Dashboard:

Substep 4.1: Open Kibana in the Web Browser

- URL: http://localhost:5601



Help us improve the Elastic Stack

To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.

Welcome home

Enterprise Search
Create search experiences with a refined set of APIs and tools.

Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

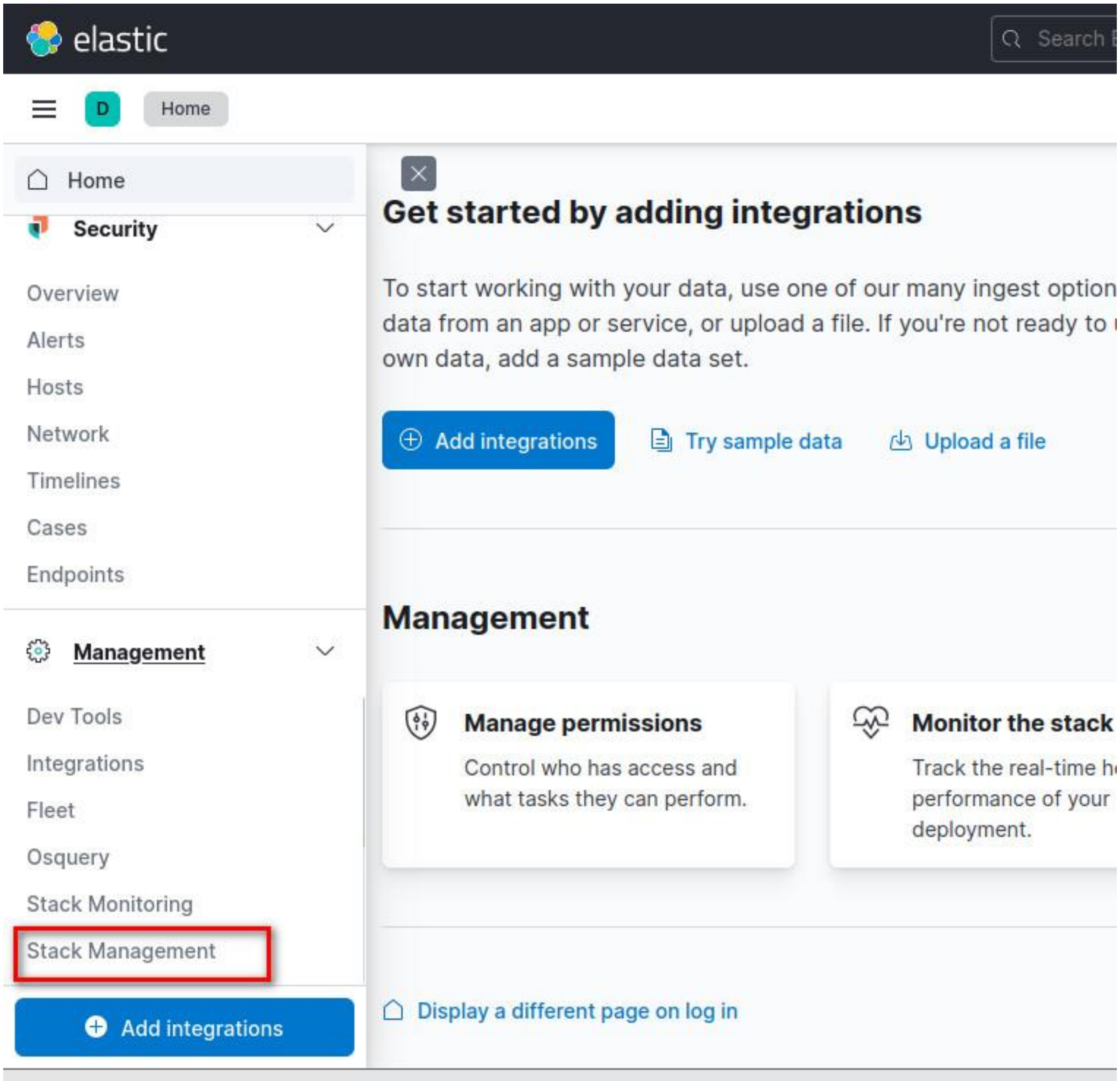
Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

Analytics
Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

Substep 4.2: Create a New Index Pattern

- Go to "Management" -> "Index Patterns."



Get started by adding integrations

To start working with your data, use one of our many ingest option data from an app or service, or upload a file. If you're not ready to own data, add a sample data set.

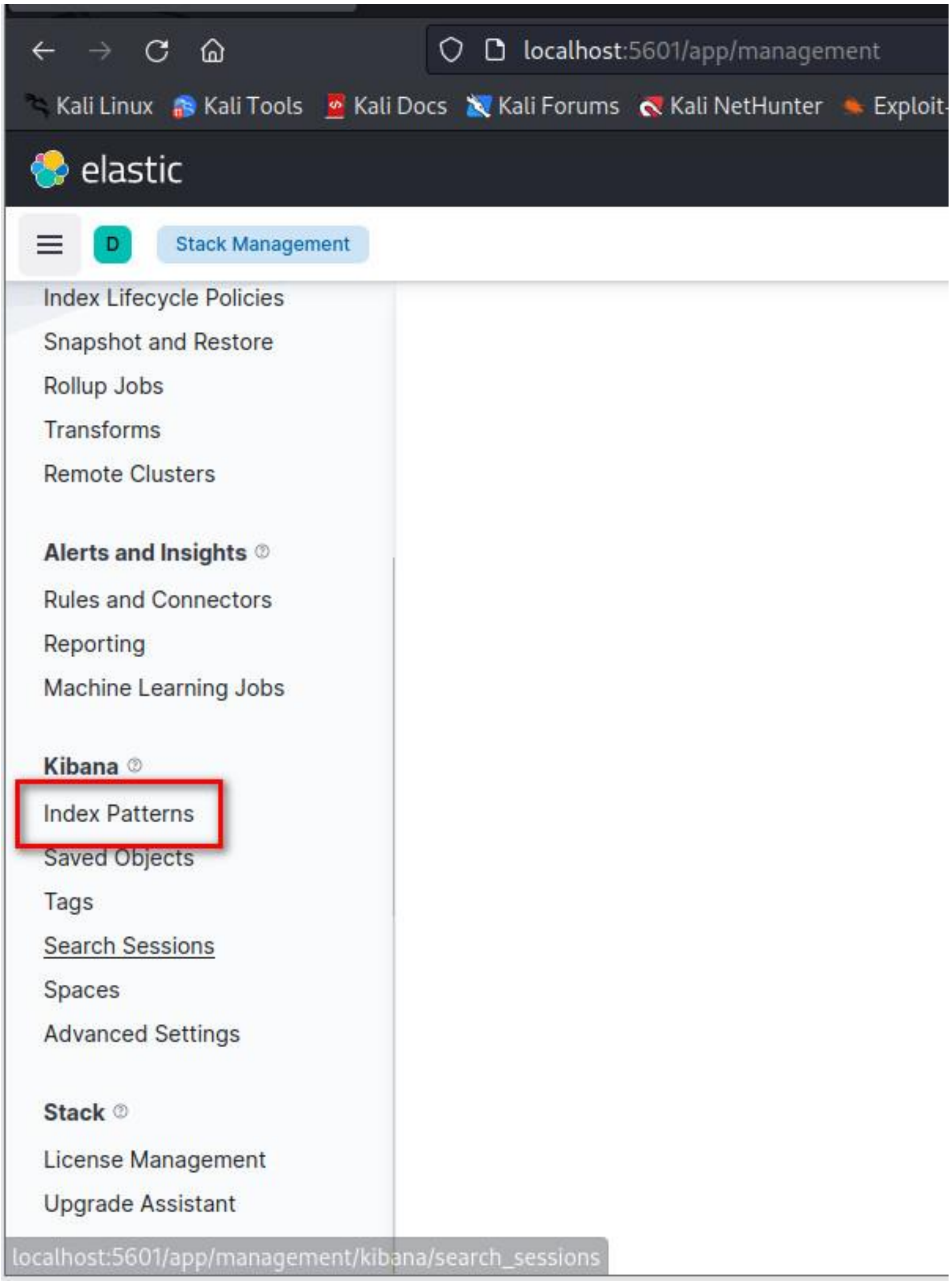
Management

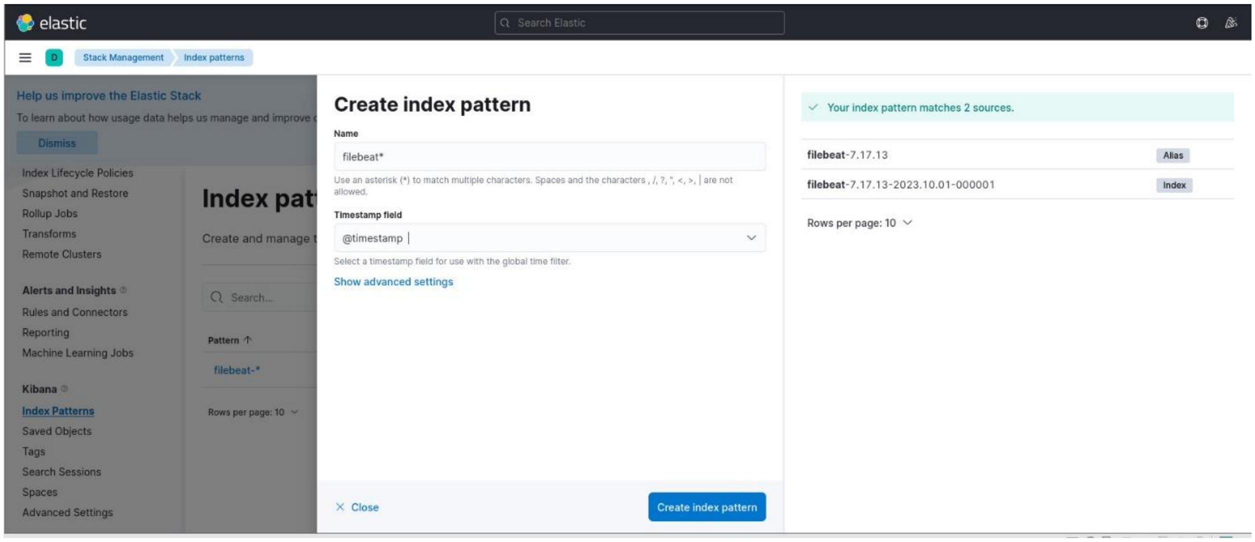
Manage permissions

Control who has access and what tasks they can perform.

Monitor the stack

Track the real-time h performance of your deployment.


localhost:5601/app/management
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-
elastic
D
Stack Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters
Alerts and Insights
Rules and Connectors
Reporting
Machine Learning Jobs
Kibana
Index Patterns
Saved Objects
Tags
Search Sessions
Spaces
Advanced Settings
Stack
License Management
Upgrade Assistant
localhost:5601/app/management/kibana/search_sessions



Create index pattern

Name: filebeat*

Use an asterisk (*) to match multiple characters. Spaces and the characters , /, ?, ", <, >, | are not allowed.

Timestamp field: @timestamp

Select a timestamp field for use with the global time filter.

✓ Your index pattern matches 2 sources.

filebeat-7.17.13	Alias
filebeat-7.17.13-2023.10.01-000001	Index

- Choose "logstash-*" as the index pattern and set '@timestamp' as the time filter.

Integrating Squid Proxy with ELK and DMZ Setup

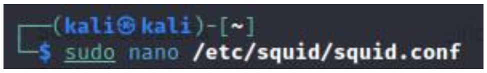
Purpose:

The purpose of this integration is to establish a Squid proxy server, monitor its activity through ELK, and set up a DMZ (Demilitarized Zone) to segment the ELK and Squid services for security.

Step 1: Install and Setup Squid (Assuming Squid Proxy is already installed):

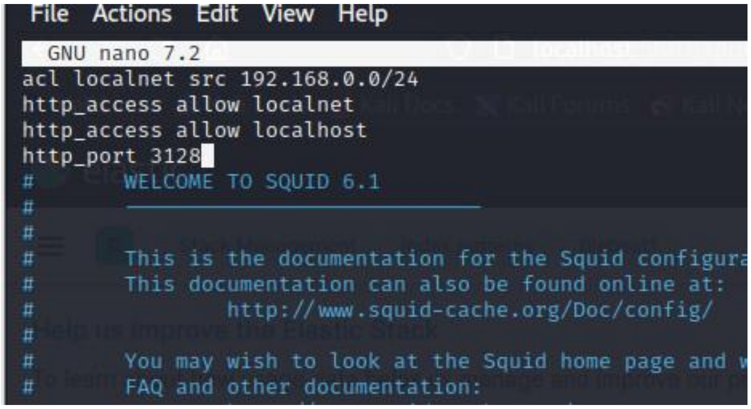
Substep 1.1: Configure Squid to Allow Traffic

- Command:



```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/squid/squid.conf
```

- Configuration:

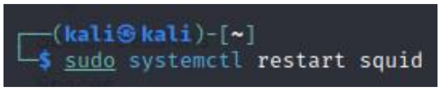


```
  GNU nano 7.2
acl localnet src 192.168.0.0/24
http_access allow localnet
http_access allow localhost
http_port 3128
#       WELCOME TO SQUID 6.1
#       ____________________
#
#       This is the documentation for the Squid configura
#       This documentation can also be found online at:
#               http://www.squid-cache.org/Doc/config/
#
#       You may wish to look at the Squid home page and w
#       FAQ and other documentation:
```

- Implication: Squid is configured to allow traffic from the local network and localhost on port 3128.

Substep 1.2: Start/Restart Squid

- Command:



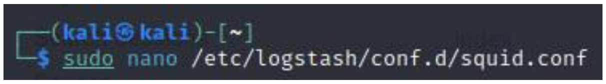
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart squid
```

- Implication: Changes to the Squid configuration are applied.

Step 2: Integration with ELK:

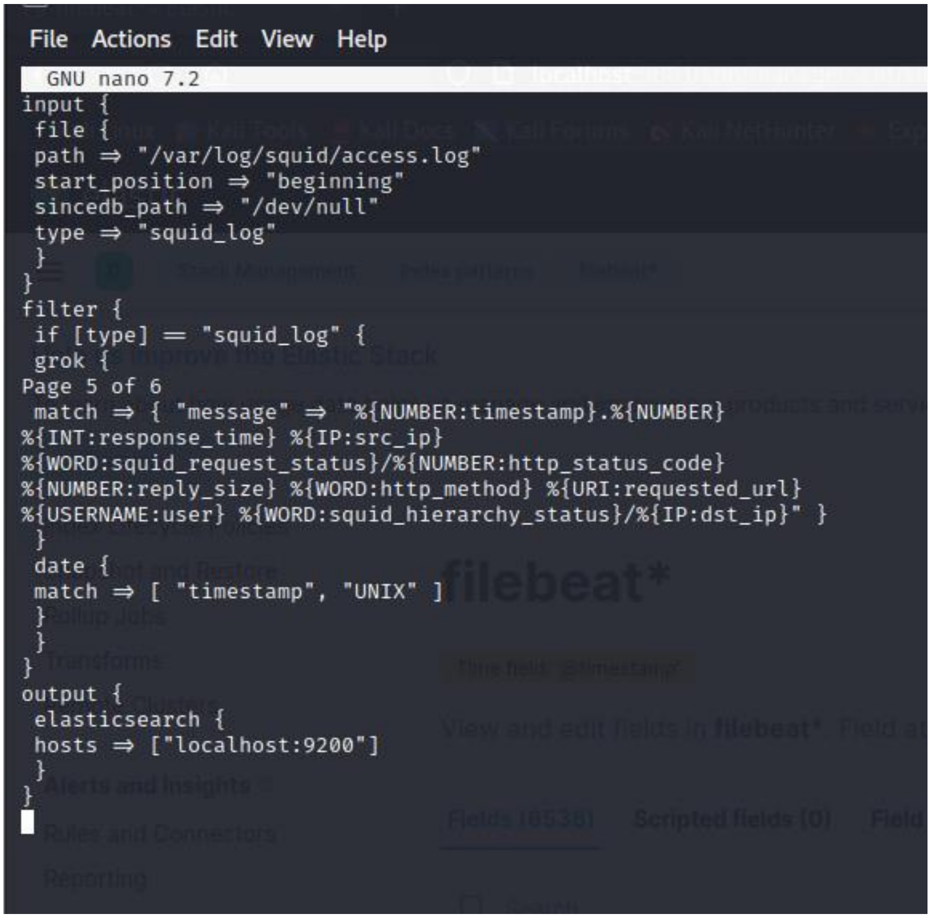
Substep 2.1: Create a New Logstash Pipeline to Process Squid Logs

- Command: **sudo vim /etc/logstash/conf.d/squid.conf**



```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/logstash/conf.d/squid.conf
```

- Configuration:

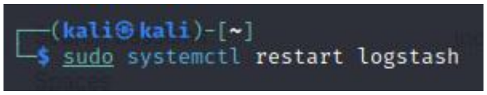


```
input {
 file {
 path => "/var/log/squid/access.log"
 start_position => "beginning"
 sincedb_path => "/dev/null"
 type => "squid_log"
 }
}
filter {
 if [type] == "squid_log" {
 grok {
Page 5 of 6
 match => { "message" => "%{NUMBER:timestamp}.%{NUMBER}
%{INT:response_time} %{IP:src_ip}
%{WORD:squid_request_status}/%{NUMBER:http_status_code}
%{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"]
 }
}
```

- Implication: Logstash is configured to process Squid access logs using the specified grok patterns and sends the processed logs to Elasticsearch.

Substep 2.2: Reload/Restart Logstash

- Command: **sudo systemctl restart logstash**

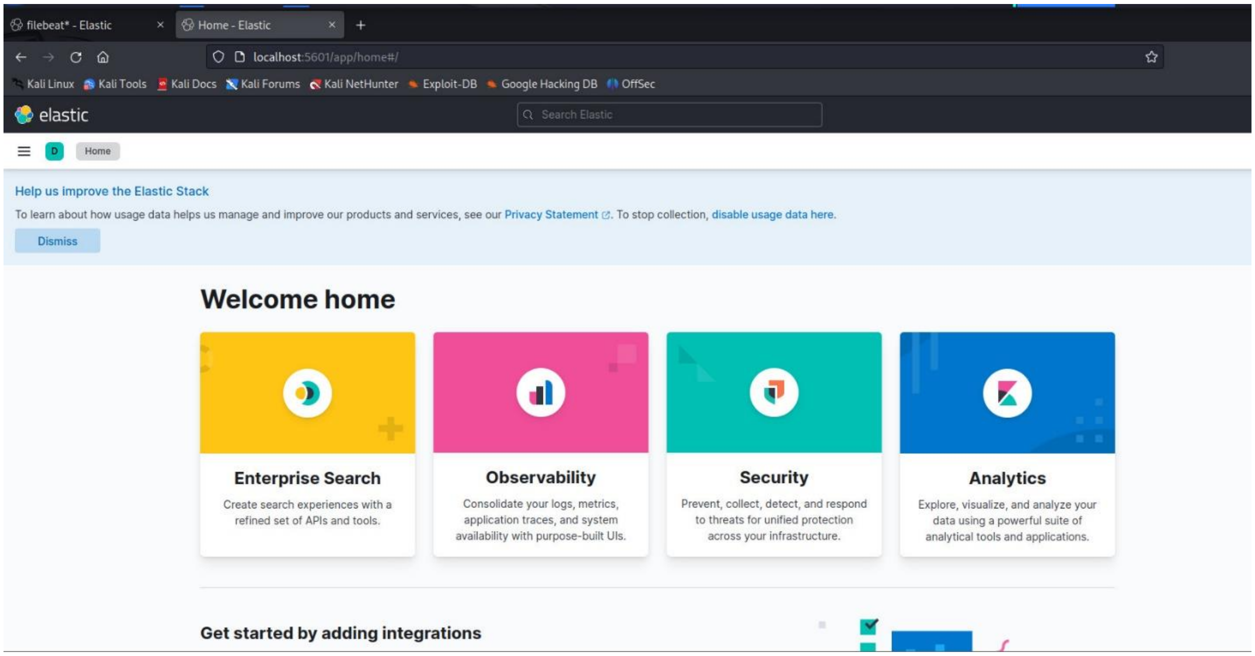


```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart logstash
```

- Implication: Logstash is restarted to apply the new configuration.

Substep 2.3: Visualize Squid Logs in Kibana

- URL: http://localhost:5601



Help us improve the Elastic Stack

To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.

Welcome home

Enterprise Search

Create search experiences with a refined set of APIs and tools.

Observability

Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

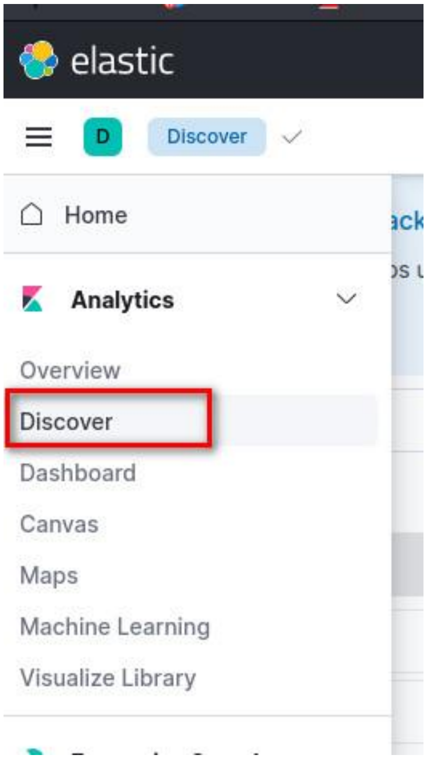
Security

Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

Analytics

Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

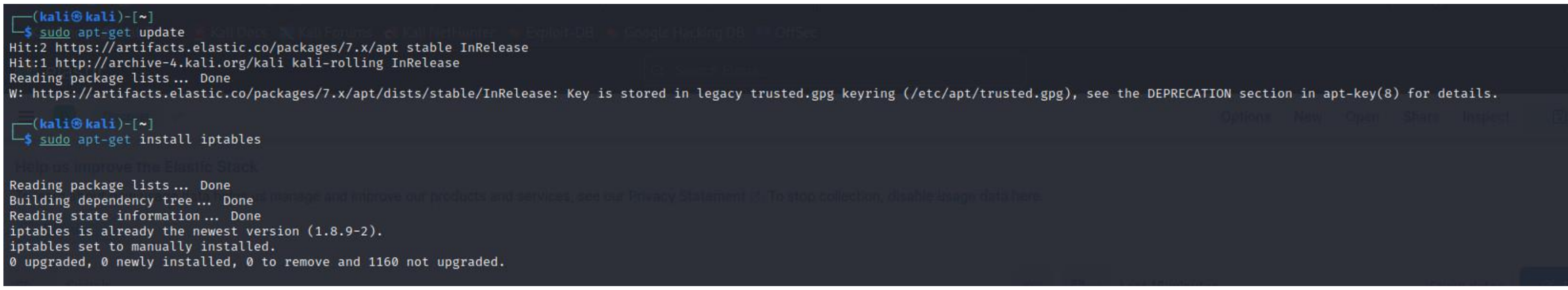


- Implication: Squid logs are visualized and analyzed in Kibana, similar to Apache logs.

Step 3: DMZ Setup:

Substep 3.1: Setup iptables

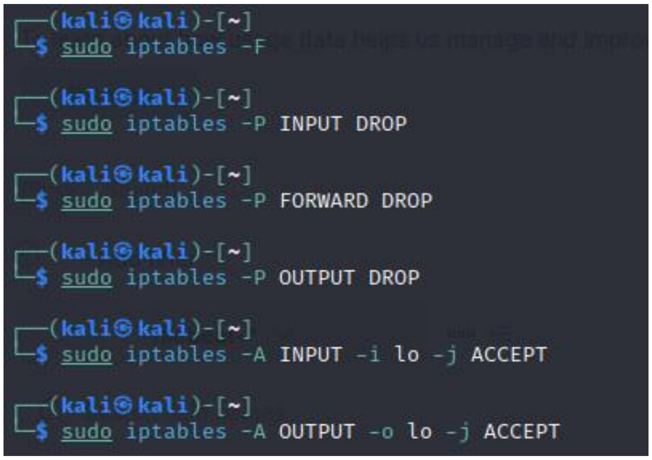
- Commands:



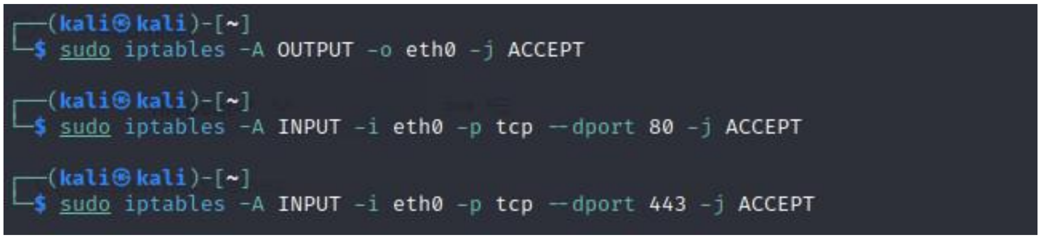
```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
Hit:2 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:1 http://archive-4.kali.org/kali kali-rolling InRelease
Reading package lists... Done
W: https://artifacts.elastic.co/packages/7.x/apt/dists/stable/InRelease: Key is stored in legacy trusted.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.
0 upgraded, 0 newly installed, 0 to remove and 1160 not upgraded.
```

Substep 3.2: Configure iptables Rules

- Commands:



```
┌──(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

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -i lo -j ACCEPT

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A OUTPUT -o lo -j ACCEPT
```

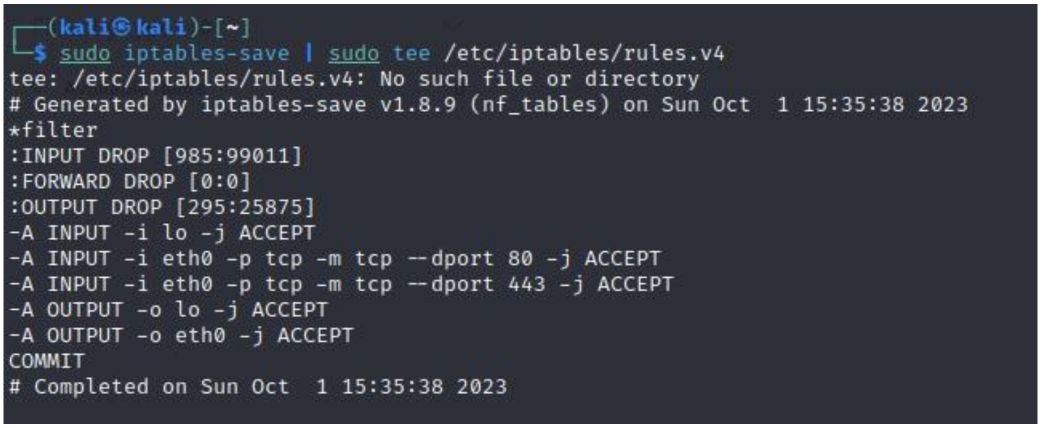


```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A OUTPUT -o eth0 -j ACCEPT

┌──(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
```

- Implication:
  - Default policy is set to drop for INPUT, FORWARD, and OUTPUT, providing a secure starting point.
  - Loopback traffic is allowed.
  - Outgoing traffic on the external interface is allowed.
  - Incoming HTTP and HTTPS traffic on the external interface is allowed.

Substep 3.3: Save iptables Configuration

- Command:



```
┌──(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 15:35:38 2023
*filter
:INPUT DROP [985:99011]
:FORWARD DROP [0:0]
:OUTPUT DROP [295:25875]
-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 15:35:38 2023
```

- Implication: Saves the iptables rules to persist after a reboot.