Wireshark Traffic Analysis Project

1. Isolating Web Application Traffic and Identifying the Network Interface

To capture traffic specifically related to the Django web application, I applied the following Wireshark display filter:

ip.addr == 127.0.0.1 && tcp.port == 8000

This filter helped isolate traffic between the local host and the Django server, which was running on port 8000. Since the application was hosted within a virtual machine, I monitored traffic using the `ens33` network interface—this corresponds to the VM's primary network adapter.

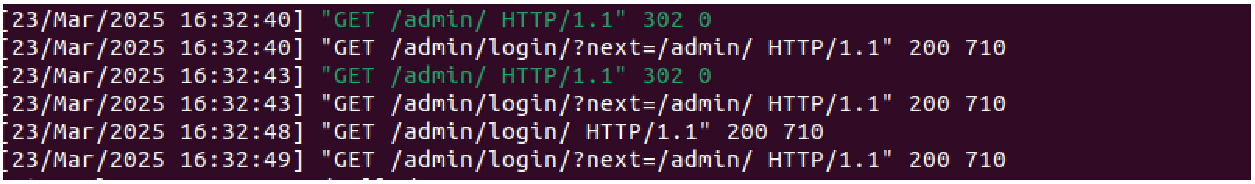
2. Finding Plain-Text Data in the Wireshark Trace

With the Django application running and accessed through a browser using HTTP, I captured the network traffic using Wireshark. The applied filter allowed me to focus on communication between the browser and server.

Upon inspecting the captured packets, I found clear-text data in the HTTP payloads. This included:

- **HTML page content**
- **URL paths and query strings**
- **Form data entered by the user**

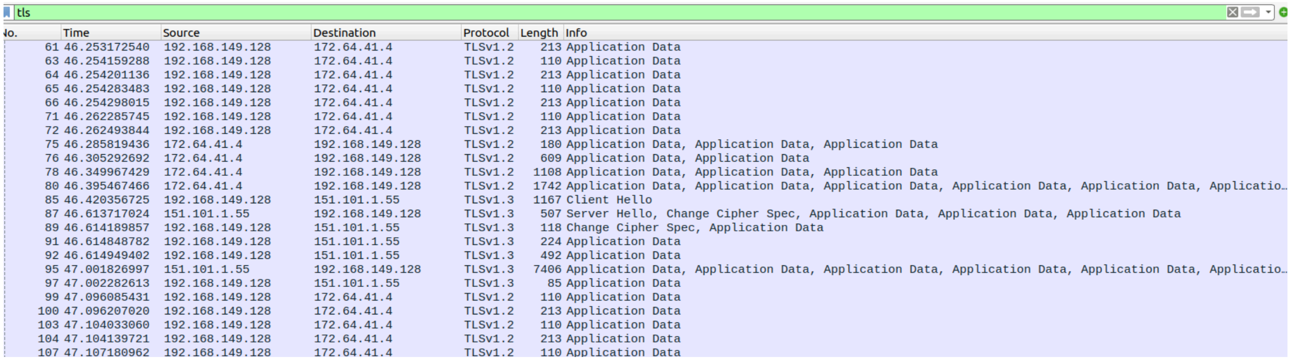
This observation demonstrated that without encryption, web traffic is fully visible to anyone monitoring the network. A screenshot was taken to illustrate the presence of this unprotected data.



```
[23/Mar/2025 16:32:40] "GET /admin/ HTTP/1.1" 302 0
[23/Mar/2025 16:32:40] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 710
[23/Mar/2025 16:32:43] "GET /admin/ HTTP/1.1" 302 0
[23/Mar/2025 16:32:43] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 710
[23/Mar/2025 16:32:48] "GET /admin/login/ HTTP/1.1" 200 710
[23/Mar/2025 16:32:49] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 710
```

3. Searching for Clear-Text Data After Enabling SSL/TLS

Next, I attempted to configure my Django application to run with SSL/TLS using the django-sslserver package. After setting up the secure server and accessing the application via HTTPS, I repeated the traffic capture process. This time, when I inspected the packets in Wireshark, I could no longer see any readable plain-text data. The previously visible HTML content and form data were now encrypted. The screenshot below shows this:



tls

No.	Time	Source	Destination	Protocol	Length	Info
61	46.253172540	192.168.149.128	172.64.41.4	TLSv1.2	213	Application Data
63	46.254159288	192.168.149.128	172.64.41.4	TLSv1.2	110	Application Data
64	46.254201136	192.168.149.128	172.64.41.4	TLSv1.2	213	Application Data
65	46.254283483	192.168.149.128	172.64.41.4	TLSv1.2	110	Application Data
66	46.254298015	192.168.149.128	172.64.41.4	TLSv1.2	213	Application Data
71	46.262285745	192.168.149.128	172.64.41.4	TLSv1.2	110	Application Data
72	46.262493844	192.168.149.128	172.64.41.4	TLSv1.2	213	Application Data
75	46.285819436	172.64.41.4	192.168.149.128	TLSv1.2	180	Application Data, Application Data, Application Data
76	46.305292692	172.64.41.4	192.168.149.128	TLSv1.2	609	Application Data, Application Data
78	46.349967429	172.64.41.4	192.168.149.128	TLSv1.2	1108	Application Data, Application Data, Application Data
80	46.395467466	172.64.41.4	192.168.149.128	TLSv1.2	1742	Application Data, Application Data, Application Data, Application Data, Application Data, Applicatio…
85	46.420356725	192.168.149.128	151.101.1.55	TLSv1.3	1167	Client Hello
87	46.613717024	151.101.1.55	192.168.149.128	TLSv1.3	507	Server Hello, Change Cipher Spec, Application Data, Application Data, Application Data
89	46.614189857	192.168.149.128	151.101.1.55	TLSv1.3	118	Change Cipher Spec, Application Data
91	46.614848782	192.168.149.128	151.101.1.55	TLSv1.3	224	Application Data
92	46.614949402	192.168.149.128	151.101.1.55	TLSv1.3	492	Application Data
95	47.001826997	151.101.1.55	192.168.149.128	TLSv1.3	7406	Application Data, Application Data, Application Data, Application Data, Application Data, Applicatio…
97	47.002282613	192.168.149.128	151.101.1.55	TLSv1.3	85	Application Data
99	47.096085431	192.168.149.128	172.64.41.4	TLSv1.2	110	Application Data
100	47.096207020	192.168.149.128	172.64.41.4	TLSv1.2	213	Application Data
103	47.104033060	192.168.149.128	172.64.41.4	TLSv1.2	110	Application Data
104	47.104139721	192.168.149.128	172.64.41.4	TLSv1.2	213	Application Data
107	47.107180962	192.168.149.128	172.64.41.4	TLSv1.2	110	Application Data

Conclusion

This project demonstrated the stark contrast between HTTP and HTTPS traffic. Using Wireshark, I was able to show how sensitive data, such as form submissions and page content, is exposed over HTTP but protected under HTTPS.

Enabling SSL/TLS not only encrypts the application's traffic but also prevents attackers from easily viewing user input and session details. This reinforces the importance of using secure protocols in any web-facing application.