Paweł Kankowski 165764
Wojciech Borostowski 165375
Alexander Stiegler 165534

Wykorzystanie narzędzia docker do zobrazowania ataku typu man in the middle - Instrukcja

1. Wprowadzenie

Celem ćwiczenia jest zaprezentowanie ataku Man-in-the-Middle (MiTM) przy użyciu narzędzia Docker - otwartego oprogramowania do realizacji wirtualizacji na poziomie systemu operacyjnego, tworzącego kontenery ze skonfigurowanymi aplikacjami w środku.

Atak MiTM jest atakiem polegającym na wniknięciu atakującego lub złośliwych narzędzi pomiędzy komputer ofiary a docelowy zasób, do którego ofiara chce mieć dostęp. Jest kilka rodzajów tych ataków, ale wszystkie cechuje przechwytywanie wysyłanych komunikatów między dwiema stronami i możliwość podszywania się pod jedną ze stron. Ten atak kojarzy się przede wszystkim negatywnie przez wykorzystywanie go przez intruzów w sieci, lecz może być również zastosowany legalnie do monitorowania i analizy ruchu w sieci. W szczególności tutaj na myśl przychodzi zastosowanie przy testowaniu i analizie aplikacji webowych lub w środowisku pracy, gdzie administrator sieci, chciałby mieć kontrolę i wgląd w ruch odbywający się w jego sieci. Jednym z popularniejszych narzędzi jest **Mitmproxy** (oprócz tego, możemy jeszcze wyróżnić podobne programy: Burp Suite, Sslsplit, Squid).

Mitmproxy

Mitmproxy - zestaw narzędzi, które zapewniają przechwytywanie zaszyfrowanego ruchu przez SSL/TLS przez proxy dla protokołów HTTP/1, HTTP/2 i Websocketów.

Do głównych funkcji zaliczamy:

- Przechwytywanie i modyfikacja na bieżąco zapytań i odpowiedzi HTTP i HTTPS.
- Zapisywanie zapytań
- Odtwarzanie ze strony klienta i serwera wcześniej zapisanych zapytań i odpowiedzi
- Tryb reverse-proxy by przekierowywać ruch od określonych serwerów
- Tryb transparent-proxy na systemach macOS i Linux
- Tworzenie rozszerzeń za pomocą języka Python i mitmproxy API
- Generowanie na bieżąco certyfikatów SSL/TLS
- inne

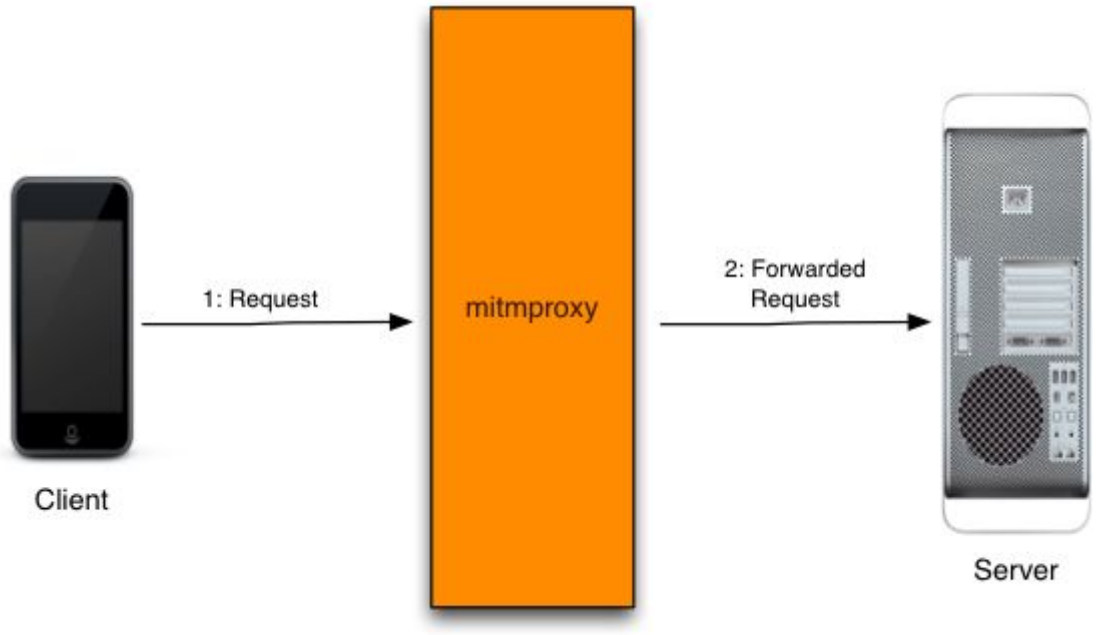
Mitmproxy zawiera 3 główne narzędzia: Po krótce zapoznamy się z każdym w poszczególnych zadaniach.

- mitmproxy
- mitmweb
- mitmdump

W tym laboratorium przedstawimy najważniejsze funkcje mitmproxy, lecz po całość możliwości odsyłamy na oficjalną stronę https://docs.mitmproxy.org.

Jak mitmproxy działa?

Skonfigurowanie klienta do użycia proxy przy HTTP nie jest skomplikowane i polega na prostym przekierowaniu zapytania przez proxy, tak jak przedstawiono to na poniższym rysunku.



Zupełnie inaczej wygląda sytuacja z protokołem HTTPS, zaszyfrowanym przez protokoły SSL/TLS.

Podczas połączenia do strony z protokołem https, klient wysyła następujące zapytanie:

```
CONNECT example.com:443 HTTP/1.1
```

Tradycyjny proxy nie może zobaczyć lub zmanipulować zaszyfrowanego zapytania, więc po prostu otwiera pipe pomiędzy klientem a serwerem. Proxy staje się jedynie pośrednikiem który może tylko przekierowywać dane zapytania bez możliwości zajrzenia do środka.

W celu ominięcia szyfrowania SSL/TLS, mitmproxy generuje na bieżąco, z każdą nową domeną do której wysyłą zapytanie klient, własne certyfikaty CA (Certificate Authority). Najpierw by zapewnić, że klient będzie ufał tym certyfikatom musimy ręcznie zarejestrować zaufany certyfikat CA. Będziemy to wykonywać w pierwszych zadaniach przy konfiguracji mitmproxy.

Aby generować na bieżąco certyfikaty potrzebna jest nazwa domenowa strony. Niestety nie w każdym przypadku jest to takie oczywiste. (np. łączymy się do strony po adresie ip, virtual hosting powoduje, że jedna domena posiada wiele ip). Mitmproxy radzi sobie z tym za pomocą takich rozwiązań jak upstream certificate sniffing, przekopiowaniem Subject Alternative Name z oryginalnego certyfikatu czy rozszerzeniem certyfikatu Server Name Indication. Łącząc to wszystko, cały ruch HTTPS możemy zobrazować poniższym rysunkiem.

Po szczegółowe informacje odsyłamy na oficjalną stronę - https://docs.mitmproxy.org/stable/concepts-howmitmproxyworks.

2. Instrukcja obsługi

Docker

W następujących ćwiczeniach, najczęściej będziemy korzystać z komendy:

```
$ docker run --rm -it -v ~/tmp/mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080 mitmproxy/mitmproxy
```

Docker jest to zaawansowane narzędzie, posiadającą szeroki wachlarz funkcji. Nie będziemy ich tu wszystkich wyjaśniać. Przedstawimy tylko te, które będą używane przy wykonywaniu zadań z tego laboratorium.

Komenda uruchamiająca kontener ma następującą składnię:

```
$ docker run  <flagi konfiguracyjne> <nazwa obrazu> <komendy przesyłane do kontenera>
```

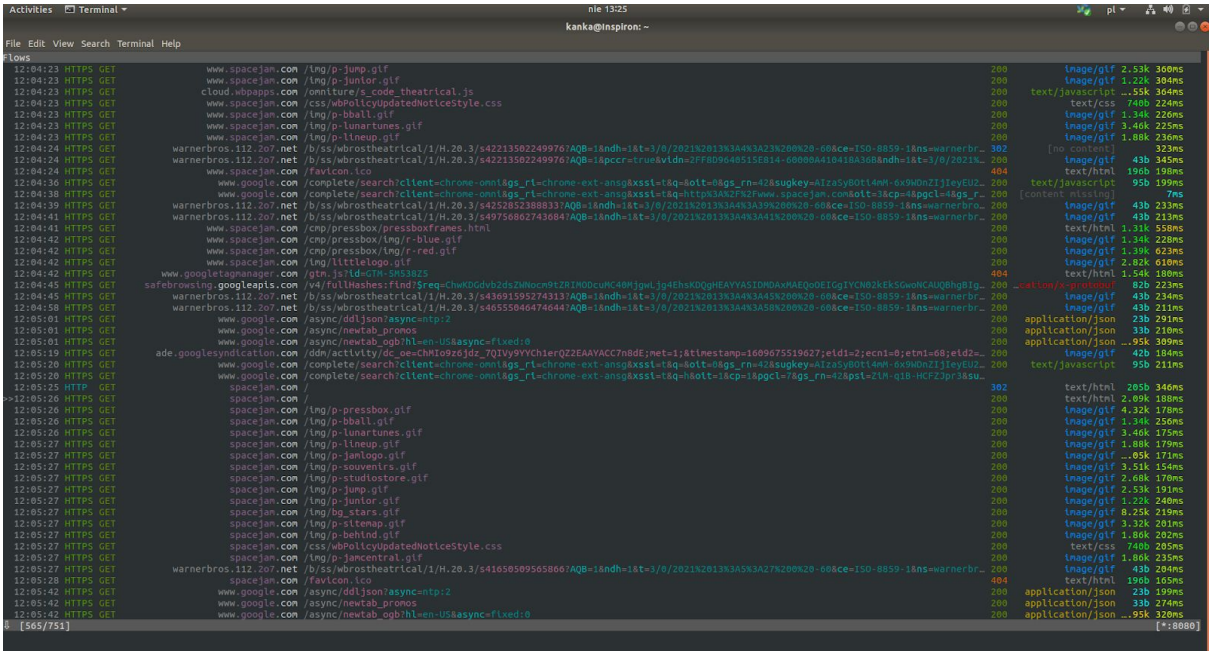
Flagi konfiguracyjne:
-p <nr portu hosta>:<nr portu kontenera> - mapowanie portów
-v <ścieżka do folderu na hoście>:<ścieżka do folderu na kontenerze> - Tworzenie volumenów, które pozwalają na współdzielenie danych przez kontener i hosta. Zapobiegają również utracie danych przy każdym wyłączaniu kontenera.
--rm - Automatyczne czyszczenie. Usuwa kontener po zamknięciu kontenera.
-it - Pozwala na interakcję z terminalem poprzez terminal.

Mitmproxy - Ustawienie certyfikatu CA.

Przy pierwszym uruchomieniu mitmproxy w folderze /home/.mitmproxy zostaną wygenerowane certyfikaty CA. Aby się do nich dostać tworzymy volumen, który będzie odpowiadał folderowi /home/.mitmproxy w środku kontenera. W poleceniach w instrukcji domyślnie jest to miejscy /tmp/mitmproxy. Stamtąd należy wziąć certyfikat do zaimportowania.

Mitmproxy - Przechwytywanie zapytań

Jak szybko można zauważyć mitmproxy przechwytuje wszystkie zapytania http i https jakie wykonujesz. Każdy wiersz odpowiada jednemu przechwytywanemu zapytaniu. Możesz użyć mitmproxy do dokładnej analizy ruchu webowego, a dzięki zaimportowaniu certyfikatu CA, dane są w odszyfrowanej postaci.



Rys 1. Panel mitmproxy.

Po kliknięciu w dowolny przechwycony komunikat, ukażą ci się do wyboru 3 panele:

Request - możesz zobaczyć tutaj zapytanie które zostało wysłane na daną stronę.
Response - odpowiedź serwera na zapytanie
Detail - szczegóły dotyczące serwera (adresy ip, certyfikaty CA...)

Rys 2. Przykładowe zdjęcie panelu po wybraniu przechwyconego zapytania.

Mitmproxy - Ustawienie filtrów

Po krótkim czasie ilość przechwyconych zapytań może być przytłaczająca, możemy wtedy ustawić filtry. Naciśnij ‘f’ lub wpisz “: set view_filter ‘google.com’ “’ by znaleźć zapytania tylko dotyczące domeny google. Jest to najprostszy sposób filtrowania pakietów, by sprawdzić bardziej zaawansowane funkcję sprawdź: https://docs.mitmproxy.org/stable/concepts-filters/.

Mitmproxy - Zatrzymywanie zapytań

Przydatną funkcją mitmproxy jest zatrzymywanie zapytań w trakcie ruchu. Przechwycone zapytanie jest wtedy wstrzymywane, użytkownik może wtedy odrzucić lub zmodyfikować dane zapytanie przed wysłaniem do serwera. Przechwytywanie wszystkich zapytań nie jest wygodne, ponieważ przerywa ci to co chwilę poprawną pracę przeglądarki. Dlatego mitmproxy oczekuje na podanie filtru, jako argument do komendy set intercept, który sprecyzuje nam informacje o zapytaniu, który chcemy przejąć. Naciśnij ‘i’ lub wpisz “:set intercept ‘’ ”, po czym w cudzysłowiu napisz filtr, by przechwycić stronę, która nas interesuje. Dla przykładu by przechwycić wszystkie strony, z domeny gov.pl, użyjemy komendy :set intercept ‘~u .*\.gov.pl’

Po złapaniu danej strony, zostanie ona podświetlona na czerwono w panelu mitmproxy i będzie czekała na dalsze operacje. Możemy wznowić ruch za pomocą zaznaczenia danej strony (strzałki >> muszą wskazywać na dane zapytanie) za pomocą przycisku ‘a’ lub odrzucić za pomocą ‘X’ (Uwaga: Duża litera X). Modyfikowanie pakietu opiszemy w kolejnym podpunkcie.
Aby usunąć filtr przechwytujący, powtórz komendę zostawiając cudzysłów pusty (wpisz -> “:set intercept ‘’ “).

Mitmproxy - Modyfikacja zapytań

Po przechwyceniu zapytania możemy modyfikować jego zawartość. Należy wybrać dane zapytanie (strzałki >> muszą wskazywać na zapytanie), a następnie nacisnąć przycisk ‘e’. Ukaże się menu, z którego można wybrać którą część chcemy edytować.

Mitmweb

Mitmweb - jest to web-based GUI narzędzia mitmproxy.

Rys 3. Panel mitmweb.

Możemy go uruchomić za pomocą komendy:

```
$ docker run --rm -it -v ~/tmp/mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080
-p 8081:8081 mitmproxy/mitmproxy mitmweb --web-host 0.0.0.0
```

Uruchamiamy tutaj nie sam program mitmproxy lecz wersję przeglądarkową mitmweb**.** Samo proxy jest uruchomione na porcie 8080 localhosta, lecz podprogram strona mitmweb znajduje się na porcie 8081. Wejdź na stronę localhost:8081 lub 0.0.0.0:8081 aby ujrzeć interfejs mitmweb.

Na chwilę obecną mitmweb jest w wersji beta, więc wciąż brakuje mu dużo funkcji mitmproxy. Na potrzeby tego laboratorium można użyc mitmproxy jak i mitmweb.

Gdy wybierzemy z panelu opcję Start, ujrzymy tam miejsca na filtrowanie zapytań, wstrzymwanie ich lub podświetlanie. Mitmweb obsługuje tą samą składnie do tworzenia filtrów jaka znajduje się w mitmproxy. (Mitmweb jest nadal w wersji beta, nie wszystkie filtry będą działać).

Mitmdump

Mitmdump udostępnia funkcjonalność podobną do narzędzia tcpdump. Pozwala na analizę, zapisywanie i automatyzacje transformacji na ruchu HTTP.

Dzięki mitmdump możemy sami pisać potrzebne dla nas rozszerzenia w postaci skryptów w języku python wraz z użyciem mitmproxy API. Poprzez odpowiadanie na zdarzenia, które są generowane podczas pracy mitmproxy, możemy wchodzić w interakcje z zapytaniami i zmieniać sposób zachowania mitmproxy.

Uruchomić mitmdump możemy za pomocą komendy:

```
$ docker run --rm -it -v ~/tmp/mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080 mitmproxy/mitmproxy mitmdump <argumenty>
```

Aby użyć skryptów należy najpierw przesłać je do użycia przez kontener. Najlepiej użyć do tego volumen, który ustawiamy za pomocą flagi “-v”. Domyślnie w komendach volumen wskazujemy na ścieżkę ~/tmp/mitmproxy ze strony hosta, a /home/mitmproxy/.mitmproxy ze strony kontenera. To znaczy, że to co znajdzie się w folderze ~/tmp/mitmproxy na hoście, znajdziemy w kontenerze pod ścieżką /home/mitmproxy/.mitmproxy

Przykład

Ściągnijmy pliki potrzebne do realizacji laboratorium za pomocą narzędzia git:

```
$ git clone https://github.com/Kankarollo/MiTMInDocker.git
```

Przekopiujmy skrypt anatomy.py z folderu addons do ścieżki ~/.mitmproxy

```
$ cp addons/anatomy.py ~/tmp/mitmproxy/
```

Od strony konteneru skrypt anatomy.py znajdzie się pod ścieżką /home/mitmproxy/.mitmproxy/anatomy.py, w takim razie taką też ścieżkę musimy podać podczas uruchomienia kontenera. Aby powiedzieć programowi mitmdump, żeby uruchomił skrypt, używamy flagi “-s”:

```
$ docker run --rm -it -v ~/tmp/mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080 mitmproxy/mitmproxy mitmdump -s /home/mitmproxy/.mitmproxy/anatomy.py
```

Skrypt ten ma za zadanie zliczać zapytania. W trakcie działania mitmdump powinniśmy obserwować wiadomości o treści *“We’ve seen **X** flows”*.

Analogicznie będziemy postępować w zadaniu 3.

Więcej o mitmdump sprawdź na stronie https://docs.mitmproxy.org/stable/tools-mitmdump/ , a o rozszerzeniach - https://docs.mitmproxy.org/stable/addons-overview/.

3. Zadania laboratoryjne

Uwagi dla rozwiązujących

- Do rozwiązania wszystkich zadań wymagany jest program Docker i przeglądarka internetowa (zalecamy przegladarkę: Firefox ponieważ posiada opcje ustawienia osobistego proxy)
- System operacyjny - dowolny (Testowano na: Windows 10 i Ubuntu 18.04)

0. Instalacja dockera

Do realizacji laboratoriów należy zainstalować program docker. W celu zainstalowania odpowiedniej wersji na poszczególne systemy operacyjne odsyłamy do oficjalnej strony: docker - tutaj

Pamiętaj by sprawdzić poprawność instalacji za pomocą komendy:

```
$ docker run hello-world

  Unable to find image 'hello-world:latest' locally
  latest: Pulling from library/hello-world
  ca4f61b1923c: Pull complete
  Digest:
sha256:ca0eeb6fb05351dfc8759c20733c91def84cb8007aa89a5bf606bc8b315b9fc7
  Status: Downloaded newer image for hello-world:latest

  Hello from Docker!
  This message shows that your installation appears to be working correctly.
```

1. Konfiguracja środowiska

1.1 Pobierz wymagane obrazy dockera:

```
$ docker pull mitmproxy/mitmproxy
```

1.2 Pobierz repozytorium.

Za pomocą narzędzia git pobierz repozytorium z dodatkowymi plikami potrzebnymi do realizacji zadań.

```
$ git clone https://github.com/Kankarollo/MiTMInDocker.git
```

Wejdź do repozytorium

```
$ cd MiTMInDocker
```

1.3 Uruchom mitmproxy

Uruchom mitmproxy za pomocą poniższej komendy.

> **Linux:**
> $ docker run --rm -it -v ~/tmp/mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080 mitmproxy/mitmproxy
>
> **Windows:**
> $ docker run --rm -it -v <ścieżka do MiTMInDocker> /mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080 mitmproxy/mitmproxy

Notka: Używając tej komendy utworzy nam się folder mitmproxy, który będzie volumenem naszego kontenera, po pierwszym uruchomieniu znajdziemy tam wygenerowane certyfikaty CA, które będziemy musieli zaimportować do przeglądarki w celu poprawnego działania programu. Pamiętaj by w każdej komendzie upewnić się, że wskazujemy na ten sam folder.

Powinna nam się ukazać konsola mitmproxy. Sprawdź czy reaguje na wysyłane żądanie za pomocą komend:

> **Linux:**
> $ http_proxy=http://localhost:8080/ curl http://example.com/
> $ https_proxy=http://localhost:8080/ curl -k https://example.com/
> **Windows:**
> <pomiń>

Na panelu mitmproxy powinniśmy zobaczyć 2 nowe wiersze, które zawierają informacje o zaobserwowanym zapytaniu HTTP do http://example.com i zapytaniu HTTPS do https://example.com.

1.4 Ustaw proxy.

Instrukcja dla przeglądarki Firefox:

- **Wejdź w opcje -> Ogólne**
- **Na samym dole wejdź w zakładce Sieć w Ustawienia…**

- **Ustaw proxy tak jak na zdjęciu:**

W przypadku przeglądarki Chrome ustawienia proxy są pobierane z systemu. Dlatego trzeba ustawić proxy dla całego systemu operacyjnego. Instrukcja dla Ubuntu, Windows.

1: Spróbuj wejść na stronę https://wp.pl przez przeglądarkę internetową (chrome,firefox). Czy strona załaduje się bez żadnych kłopotów? Opisz co zauważyłeś? Przez jaki mechanizm jest to spowodowane?

1.4 Zaimportuj certyfikat CA.

Zaimportuj certyfikat CA w przeglądarce internetowej z której będziesz korzystał podczas wykonywania laboratorium.

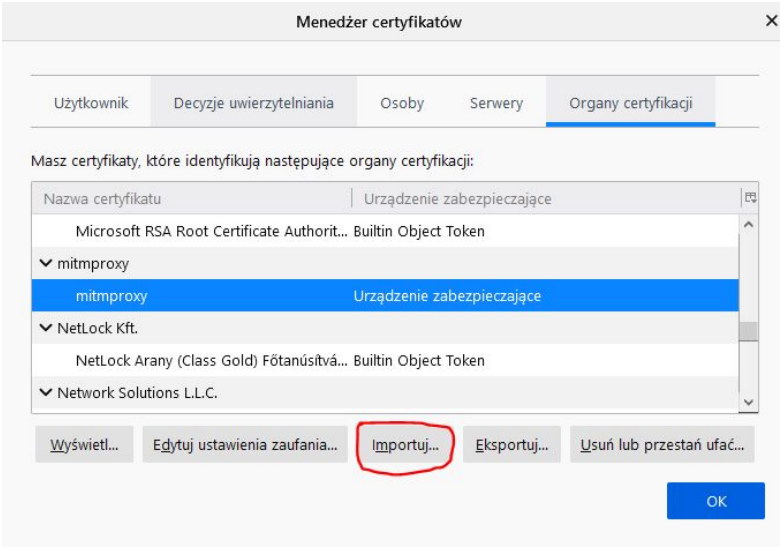
Instrukcja dla przeglądarki Firefox:

- **Wejdź w ustawienia w zakładkę Prywatność i bezpieczeństwo**
- **Znajdź podpunkt Certyfikaty i wybierz opcję “Wyświetl certyfikaty”**

- **Wybierz opcję Importuj i wybierz certyfikat. Powinieneś go znaleźć w folderze, który ustawiłeś jako twój volumen z kontenerem dockera. (Domyślnie**

~/tmp/mitmproxy) Dla przeglądarki firefox, plik ten powinien nazywać się mitmproxy-ca-cert.pem.

- **W przypadku poprawnego zaimportowania powinieneś móc znaleźć certyfikat na liście.**



Instrukcja dla Chrome:

- **Wejdź w ustawienia w zakładkę Prywatność i bezpieczeństwo**
- **Wejdź w pole Bezpieczeństwo**
- **Na samym dole wejdź w opcję “Zarządzaj certyfikatami”**
- **Wybierz opcję Importuj i wybierz certyfikat. Powinieneś go znaleźć w folderze, który ustawiłeś jako twój volumen z kontenerem dockera. (Domyślnie ~/tmp/mitmproxy) Dla systemu linux wybierz plik mitmproxy-ca-cert.pem. Dla systemu Windows wybierz plik mitmproxy-ca-cert.p12.**
- **W przypadku poprawnego zaimportowania powinieneś móc znaleźć certyfikat na liście.**

Wejdź na dowolną stronę przez protokół https (np. https://wp.pl). W przypadku poprawnego skonfigurowania, strona powinna załadować się bez żadnych problemów.

2: Wejdź na stronę https://wp.pl i sprawdź w przeglądarce certyfikat, który jest używany(naciśnij na kłódkę przy pasku url i wejdź w szczegóły). Podaj nazwę certyfikatu. Wstaw zrzut ekranu panelu mitmproxy.

2. Analiza zapytań HTTP i HTTPS.

2.1 Filtracja zapytań w mitmproxy.

Po prawidłowym skonfigurowaniu proxy należy zapoznać się z funkcjami filtrowania i przechwytywania zapytań opisanych w drugim rozdziale i odpowiedzieć na poniższe pytania.

3: Wejdź na stronę internetową - link, użyj funkcji filtrowania by na panelu były widoczne jedynie zapytania związane z podanym linkiem. Wstaw zrzut z ekranu panelu mitmproxy.

2.2 Uruchom mitmweb.

Mitmproxy zawiera również wersje GUI pod nazwą mitmweb. Więcej informacji o nim znajdziesz w drugim rozdziale.

Zakończ działanie mitmproxy i uruchom mitmweb poniższą komendą po czym otwórz jego interfejs wchodząc na adres http://127.0.0.1:8081. (Sprawdź czy flaga -v wskazuje na prawidłowy folder.)

Linux:

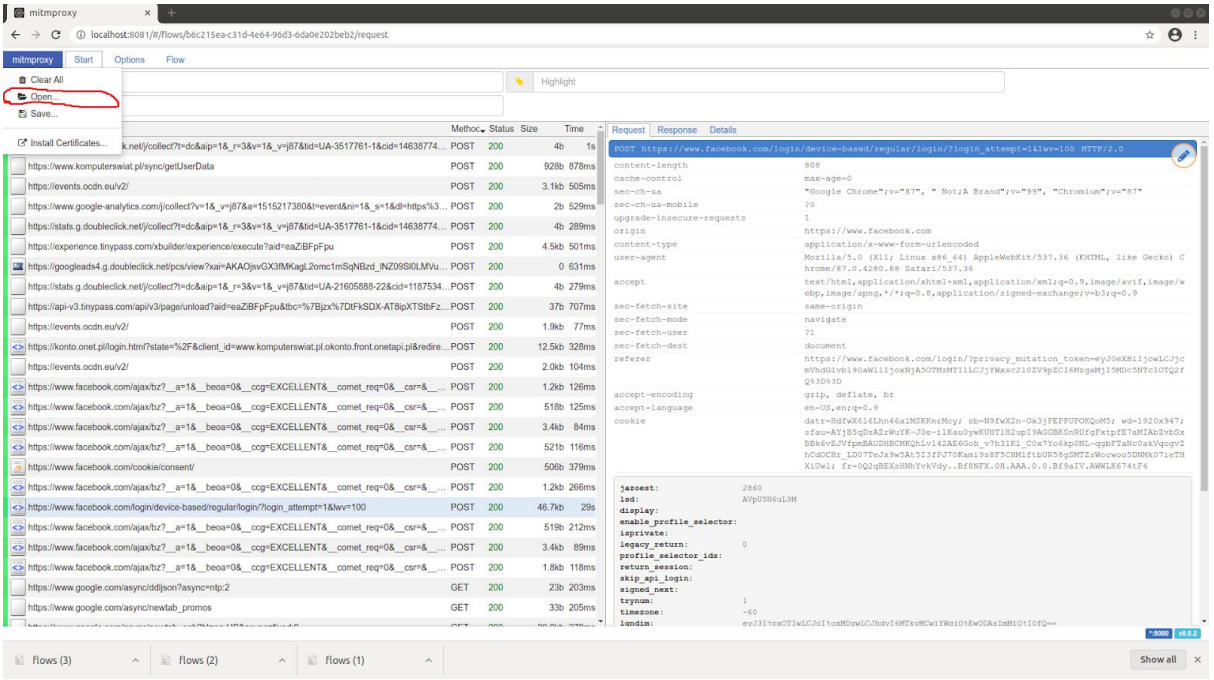
```
$ docker run --rm -it -v ~/tmp/mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080
-p 8081:8081 mitmproxy/mitmproxy mitmweb --web-host 0.0.0.0
```

Windows:

```
$ docker run --rm -it -v <ścieżka do MiTMInDocker>
/mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080 -p 8081:8081
mitmproxy/mitmproxy mitmweb --web-host 0.0.0.0
```

W celu zapoznania się z narzędziem sprawdzimy jak poszczególne strony internetowe przesyłają dane logowania.

Otwórz plik “mitmweb_exercise/zadanie_znajdz_haslo” z poziomu mitmweb. Rozwiń pasek ‘mitmproxy’ z panelu w lewym górnym rogu i wybierz opcję open.



Załadują ci się przygotowane wcześniej przechwycone zapytania HTTP i HTTPS. Znajdują się tam zapytania, które mitmweb przechwycił podczas logowania się na strony komputerświat i facebook. Znajdź je do rozwiązania kolejnego zadania.

4: Znajdź zapytania które dotyczą logowania na stronie komputerświat. Wyciągnij z nich login i hasło admina i napisz je w odpowiedzi. Zrób to samo dla logowania w przypadku platformy facebook.com. W przypadku facebook opisz w jakiej formie widzisz hasło logowania.
(Podpowiedź: Szukaj po słowie 'login' w nazwie i po metodzie POST).

Kolejne zadanie można wykonać za pomocą dowolnego narzędzia. Pamiętaj, żeby zakończyć działanie poprzedniego narzędzia przed uruchomieniem kolejnego.

5: Za pomocą poznanych narzędzi sprawdź nagłówki strony internetowej link. Czy możemy rozpoznać mechanizm HSTS po nagłówkach na tej stronie? Uzasadnij odpowiedź. Spróbuj wejść na stronę http://spacejam.com. Co się stało?

(EXTRA) Znajdź stronę (poza tą z poprzedniego pytania) na której nie występuje mechanizm HSTS. Podaj url poniżej.

3. Modyfikowanie zapytań HTTP i HTTPS

3.1 Modyfikowanie odpowiedzi strony

W tym zadaniu ukażemy jedno z potężniejszych narzędzi mitmproxy czyli tworzenie rozszerzeń za pomocą mitmproxy API i języka python by automatyzować operacje na zapytaniach HTTP i HTTPS. Do tego zadania użyjemy program mitmdump i przygotowane wcześniej skrypty w języku python. Mitmdump możemy uruchomić za pomocą poniższej komendy.(Sprawdź czy flaga -v wskazuje na prawidłowy folder.)

> **Linux:**
> $ docker run --rm -it -v ~/tmp/mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080 mitmproxy/mitmproxy mitmdump <argumenty>
>
> **Windows:**
> $ docker run --rm -it -v <ścieżka do MiTMInDocker> /mitmproxy:/home/mitmproxy/.mitmproxy -p 8080:8080 mitmproxy/mitmproxy mitmdump <argumenty>

Uruchom skrypt block_urls.py z folderu addons, opieraj się na przykładzie znajdującym się pod koniec rozdziału 2 w ramach omawiania narzędzia mitmdump. W celu poprawnego działania programu pamiętaj o przekopiowaniu plików block_urls.py oraz urls.txt z folderu addons do volumena, który będzie współdzielony z kontenerem mitmproxy (domyślnie ~/tmp/mitmproxy/).

Wejdź na stronę https://wp.pl by sprawdzić działanie skryptu. Powinno ukazać ci się zdjęcie z napisem "Access Denied" oznaczające blokadę strony.

By pokazać prostotę tworzonych skryptów, otwórz skrypt w dowolnym edytorze tekstu. Znajdź w kodzie pole, w którym jako string jest zapisany kod HTML, który będziesz otrzymywał w odpowiedzi w przypadku wejścia na stronę internetową, która jest obecna na liście urls.txt.

6. Zmodyfikuj skrypt block_urls.py tak by wyświetlał dowolne inne zdjęcie (lub po prostu inną treść) w przypadku wejścia na stronę znajdującą się na blackliście -urls.txt. Wstaw zrzut ekranu przeglądarki po uruchomienia twojej wersji skryptu.

3.2 Przekierowanie adresu

W tym przykładzie skorzystamy ze skryptu http_redirect.py do przekierowania adresów z naszej listy urls.txt na adres, który sami zechcemy.

Analogicznie do poprzedniego przykładu uruchom przykładowy skrypt http_redirect.py za pomocą mitmdump z poziomu dockera. I wejdź na stronę https://wp.pl. W przypadku gdy oryginalny cel podróży stosuje protokół HTTP/2, a strona na którą przekierowujemy nie wspiera tego protokołu, może dojść do błędu. W takich przypadkach należy dodać flagę --no-http2 na koniec komendy uruchamiającej mitmdump.

W przypadku poprawnego uruchomienia, po wejściu na stronę https://wp.pl powinniśmy zauważyć, że tak na prawdę znajdujemy się na stronie mitmproxy.org.

Gdy otworzymy skrypt w dowolnym edytorze tekstu, łatwo zobaczyć która linia kodu odpowiada za wybranie strony na którą przekierowujemy użytkownika.

7. Zmodyfikuj skrypt http_redirect.py tak by przekierowywał stronę z listy urls.txt na dowolną wybraną przez ciebie stronę (inną niż w przykładzie). Wstaw zrzut ekranu przeglądarki po uruchomienia twojej wersji skryptu.

Po więcej przykładów odsyłam na oficjalną stronę mitmproxy https://docs.mitmproxy.org/stable/addons-examples/#http-redirect-requests.

4. Czyszczenie (opcjonalne)

W celu usunięcia obrazu dockera użyj komend:

```
$ docker image ls
```

Znajdź wiersz który dotyczy obrazu mitmproxy/mitmproxy zapamiętaj jego TAG. A następnie usuń go komendą:

```
$ docker rmi <image_TAG>
```

4. Zakończenie

Jest to koniec zadań z tego laboratorium. Wyślij stworzony dokument tekstowy i nie zapomnij napisać też opinii co do zadań.

Dziękujemy za uwagę, mamy nadzieję, że się czegoś nowego nauczyliście! :)

P.S W przypadku wystąpienia pytań czy problemów zapraszamy do kontaktu jakąkolwiek drogą do któregoś z nas.