HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN

Học phần: **An toàn mạng**

Bài báo cáo:

Tìm hiểu công cụ mitmproxy

Giảng viên hướng dẫn: TS. Đặng Minh Tuấn
Sinh viên thực hiện: Ngô Văn Thắng
Mã sinh viên : B18DCAT240
Lớp: D18CQAT04_B
Nhóm môn học: 01
Tổ thực hành: 02

Hà Nội 2021

Mục lục

Danh mục từ viết tắt

Từ viết tắt	Thuật ngữ tiếng Anh/Giải thích	Thuật ngữ tiếng Việt/Giải thích
API	Application Programming Interface	Giao diện lập trình ứng dụng
CA	Certificate Authority	Tổ chức phát hành chứng chỉ
HTML	Hypertext Markup Language	Ngôn ngữ đánh dấu
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
HTTPS	Hypertext Transfer Protocol Secure	Giao thức truyền tải siêu văn bản an toàn
MITM	Man in the middle	Một kiểu tấn công trung gian Tấn công người đứng giữa
TCP	Transmission Control Protocol	Giao thức điều khiển vận chuyển
URL	Uniform Resource Locator	Định vị tài nguyên/ đường dẫn
SMTP	Simple Mail Transfer Protocol	Giao thức chuyển thư đơn giản
SAN	Subject Alternative Name	Tên thay thế trong chứng chỉ
SNI	Server Name Indication	Tên máy chủ trong chứng chỉ
SSL/TLS	Secure Socket Layer / Transport Layer Security	Bộ giao thức bảo mật SSL / TLS
YAML	Yet Another Markup Language	Một kiểu ngôn ngữ đánh dấu khác

Danh mục bảng biểu

Danh mục hình vẽ

Lời mở đầu

Kali Linux là một bản phân phối Linux được phát triển và duy trì bởi Offensive Security khi được tổ chức này phát hành vào tháng 3 năm 2013. Với mục đích là tập hợp nhiều nhất các công cụ kiểm tra bảo mật và thâm nhập trong một môi trường hệ điều hành, Kali Linux được sử dụng bởi rất nhiều hacker và các chuyên gia bảo mật. Với trên 600 công cụ được tích hợp, việc học tập sử dụng các công cụ này đối với sinh viên học tập An toàn thông tin là rất cần thiết.

Nằm trong chủ đề Sniffing và Spoofing, mitmproxy là một công cụ mạnh mẽ cho phép chúng ta thiết lập một proxy, theo dõi, chặn bắt, sửa đổi các tương tác HTTP, SSL/TLS, Websockets, …

Bài báo cáo "Công cụ mimtproxy" sẽ trình bày giới thiệu về mitmproy, các tính năng, cách hoạt động, cách cài đặt, cấu hình cùng một số bài thử nghiệm cài đặt kiểm chứng các tính năng của công cụ này.

Chương 1: Giới thiệu về mitmproxy, lịch sử hình thành

Những phiên bản nền móng đầu tiên của mitmproxy được phát triển từ tháng 2 năm 2010 bởi Aldo Cortesi.

Ngày 16/12/2010 Aldo Cortesi chính thức giới thiệu phiên bản đầu tiên của mitmproxy. Phiên bản đầu tiên này hướng tới các nhà phát triển, những người kiểm thử xâm nhập, người cần can thiệp và giám sát sâu các lưu lượng HTTP.[1]

Dự án của Aldo Cortesi nhanh chóng thu hút được các nhà phát triển khác. Đến năm 2014, Maximilian Hils bắt đầu tham gia phát triển mitmproxy. [2]

Sau 6 năm phát triển, ngày 26/12/2016 phiên bản v1.0.0 của mitmproxy được chính thức giới thiệu với nhiều tính năng và thay đổi tùy chỉnh. Các công cụ của mitmproxy ở phiên bản này chỉ hỗ trợ python 3 trở lên. Mitmweb được giới thiệu với giao diện web trực quan hơn nhưng chưa đầy đủ như mitmproxy. Khả năng tương thích với windows được cải thiện với mitmweb, mitmproxy có thể được cài đặt trên windows một cách dễ dàng hơn .Định dạng tệp cấu hình bây giờ là một tệp YAML. Cải tiến đáng kể về giao diện người dùng trên mitmproxy - bao gồm sắp xếp các luồng theo kích thước, loại và url, cải tiến thanh trạng thái, thụt lề nhanh hơn nhiều cho các chế độ xem HTTP. Bước đầu mitmproxy đã hỗ trợ HTTP/2 và Websocket mặc dù chưa đầy đủ. [3]

Với sự tham gia của nhiều nhà phát triển khác, phiên bản v2.0.0 của mitmproxy đã nhanh chóng được phát hành vào ngày 22/2/2017. HTTP / 2 hiện được bật theo mặc định hỗ trợ ở phiên bản này. Hỗ trợ proxy transparent cho OpenBSD. Tính năng kiểm tra mitmproxy CA xem có hết hạn hay không và cảnh báo người dùng tạo lại nếu cần được bổ sung. Chế độ xem nội dung hình ảnh: Hình ảnh hiện được phân tích cú pháp bằng Kaitai Struct (kaitai.io) thay vì Pillow. Điều này giúp đơn giản hóa việc cài đặt, giảm kích thước nhị phân và cho phép phân tích cú pháp bằng Python thuần túy. Cải tiến đáng kể, thực thi phạm vi phủ sóng riêng lẻ 100% cho các phần lớn của cơ sở mã, tăng phạm vi bao phủ tổng thể. [2]

Sau đó một năm, 23/2/2018 mitmproxy 3.0 được giới thiệu trên một trang web mới với các tài liệu hướng dẫn rõ ràng hơn. Mitmproxy phiên bản này có một cơ sở hạ tầng tiện ích mới mạnh mẽ. Các tiện ích mở rộng này có thể tham gia vào các sự kiện nội bộ của mitmproxy, có thể hiển thị các tùy chọn đã nhập cho cấu hình và có thể tạo các lệnh đã nhập để tương tác với người dùng. Các nhà phát triển đã chuyển

nhiều chức năng riêng của mitmproxy vào một tập hợp các phần bổ trợ bên trong, dẫn đến cấu trúc mã sạch hơn và dễ bảo trì hơn nhiều. [4]

Ngày 15/5/2018 mitmproxy v4.0 được phát hành với sự cải tiến về tốc độ. Tốc độ tăng gần gấp 4 lần trong mitmdump và tăng tốc hơn 10 lần đối với mitmproxy console. Một số thay đổi liên quan đến việc ghi nhật kí và hỗ trợ python 3.6 trở lên cũng được giới thiệu.[5]

Ngày 16/12/2019 các nhà phát triển tiếp tục phát hành mitmproxy v5.0 . Phiên bản mới này có nhiều sự thay đổi về giao diện đến từ các nhà phát triển mới tham gia, hỗ trợ tạo chứng chỉ trên các thiết bị chạy iOS 13, các cải tiến về bảo mật và sửa lỗi.[6]

Ngày 13/12/2020 phiên bản mitmproxy v6.0 tiếp tục được phát hành. Phiên bản này thực sự là một bản phát hành sửa lỗi trong khi các nhà phát triển chuẩn bị cho những thay đổi lớn hơn nhiều trong phiên bản tiếp theo. Mitmproxy v6.0 đã hỗ trợ python 3.8 trở lên. [7]

Mitmproxy v7.0 là một sự thay đổi và cải tiến đáng kể trên diện rộng của dự án với lõi proxy mới được phát hành ngày 16/7/2021.

Mitmproxy hiện hỗ trợ ủy quyền các kết nối TCP thô bên ngoài hộp, bao gồm các kết nối bắt đầu bằng lời chào phía máy chủ - ví dụ như SMTP. Opportunistic TLS (STARTTLS) chưa được hỗ trợ, nhưng TCP-over-TLS thông thường vẫn hoạt động. Chuyển đổi linh hoạt giữa HTTP/2 và HTTP/1.

Mitmproxy hiện có thể chấp nhận các yêu cầu HTTP / 2 từ máy khách và chuyển tiếp chúng đến máy chủ HTTP / 1. Bản dịch giao thức trực tuyến này hoạt động theo hai hướng: Tất cả các yêu cầu và phản hồi HTTP được tạo ra như nhau. Thay đổi này cũng giúp bạn có thể thay đổi đích yêu cầu cho các luồng HTTP / 2, điều mà trước đây hoàn toàn không thể thực hiện được.

Giờ đây, Mitmproxy hiển thị thông báo WebSocket không chỉ trong nhật ký sự kiện mà còn trong giao diện người dùng chuyên dụng. Điều này mới chỉ dành cho giao diện người dùng console thông qua mitmproxy.

Máy khách thường giao tiếp bằng bản rõ với proxy trước khi thiết lập kết nối TLS an toàn thông qua proxy với máy chủ đích. Với mitmproxy 7, các máy khách hiện có thể thiết lập TLS với proxy ngay từ đầu (trước khi đưa ra yêu cầu HTTP CONNECT). Điều này giúp có thể thêm một lớp bảo vệ quan trọng trong các mạng công cộng. Vì vậy, thay vì chỉ định đơn giản http://127.0.0.1:8080, giờ có thể sử dụng

HTTPS qua https://127.0.0.1:8080 (hoặc bất kỳ máy chủ và cổng lắng nghe nào khác).

Giao diện người dùng giao diện điều khiển của mitmproxy hiện đã có sẵn trên Windows. Các nhà phát triển đã xây dựng một tài liệu tham khảo API hoàn toàn mới cho API addon của mitmproxy. Kết hợp với các ví dụ hiện có trên GitHub, điều này làm cho việc viết các addon mới mitmproxy trở nên đơn giản hơn nhiều.[8]

Chương 2: Cách mitmproxy hoạt động

2.1 Giới thiệu các tính năng chính

Các tính năng:

- Chặn các yêu cầu và phản hồi của HTTP/HTTPS và sửa đổi chúng một cách nhanh chóng.
- Lưu các giao tiếp HTTP để phân tích và phát lại sau.
- Phát lại một giao tiếp HTTP của máy kh
- Phát lại phản hồi của một máy chủ đã ghi trước đó.
- Chuyển tiếp lưu lượng tới máy chủ được chỉ định.
- Chế độ proxy trong suốt trên MacOS và Linux.
- Thực hiện các thay đổi theo tập lệnh đối với lưu lượng HTTP bằng Python.
- Tạo chứng chỉ SSL / TLS để chặn các yêu cầu HTTPs nhanh chóng.

3 công cụ chính:

- mitmproxy là một proxy chặn tương tác, có khả năng SSL / TLS với giao diện bảng điều khiển cho HTTP / 1, HTTP / 2 và WebSockets.
- mitmweb là một giao diện dựa trên web cho mitmproxy.
- mitmdump là phiên bản dòng lệnh của mitmproxy. Hãy nghĩ đến tcpdump cho HTTP.

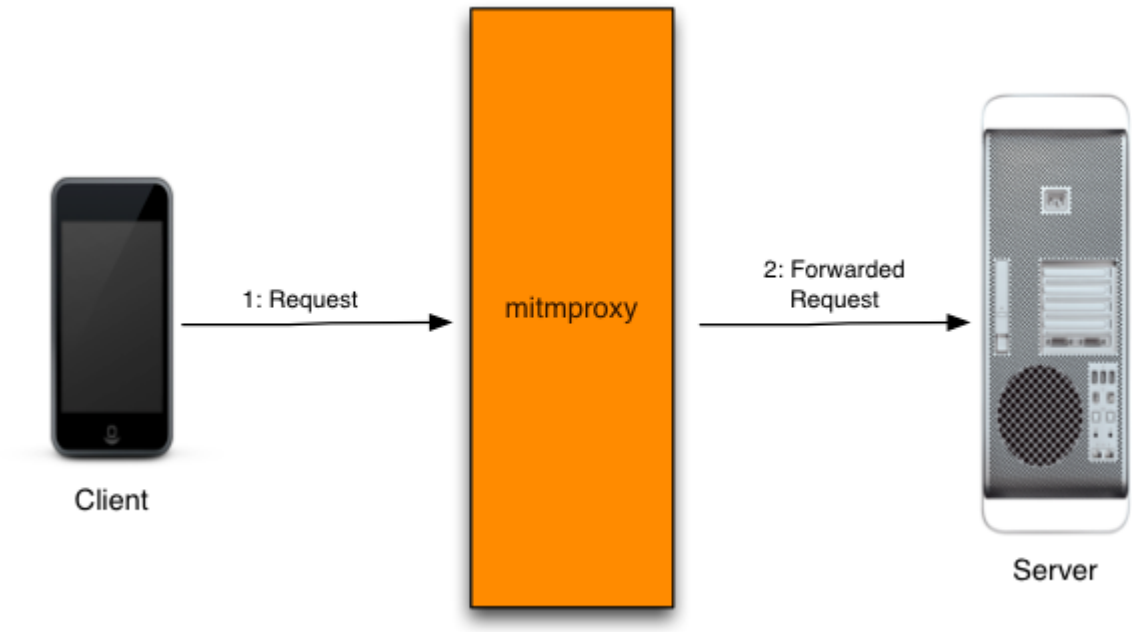
2.2 Cách hoạt động của mitmproxy

2.2.1 HTTP

Cấu hình máy khách sử dụng mitmproxy là một cách đơn giản để chặn các lưu lượng. Giao thức proxy được mã hóa trong HTTP RFC vì vậy hành vi của cả máy khách và máy chủ đều được xác định và tin cậy. Trong tương tác đơn giản nhất với mitmproxy, máy khách kết nối trực tiếp với proxy và đưa ra yêu cầu như sau:

GET http://example.com/index.html HTTP/1.1

Đây là một yêu cầu GET proxy - một dạng mở rộng của yêu cầu GET HTTP bao gồm một lược đồ và thông số kỹ thuật máy chủ, và nó bao gồm tất cả thông tin mà mitmproxy cần để tiến hành kết nối.



Hình 2.1 Kết nối HTTP proxy đơn giản.[9]

1. Máy khách kết nối tới mitmproxy và tạo yêu cầu
2. Mitmproxy kết nối thẳng đến máy chủ và chuyển tiếp yêu cầu.

2.2.2 HTTPS

Quá trình kết nối cho một proxy HTTPS thì khác một chút. Máy khách kết nối tới proxy và tạo một yêu cầu như sau:

CONNECT example.com:443 HTTP/1.1

Một proxy thông thường không thể xem cũng như thao tác luồng dữ liệu được mã hóa TLS, vì vậy yêu cầu CONNECT chỉ đơn giản là mở một đường dẫn giữa máy chủ và máy khách. Proxy ở đây chỉ đóng vai trò của một người hỗ trợ- nó chuyển tiếp dữ liệu theo cả 2 hướng mà không biết gì về nội dung. Việc thiết lập kết nối TLS diễn ra qua đường ống này và luồng yêu cầu và phản hồi tiếp theo proxy hoàn toàn không rõ.

2.2.3 MITM trong mitmproxy

MITM là viết tắt của Man-In-The-Middle một cách tiếp cận với các luồng dữ liệu truyền. Ý tưởng đơn giản là giả vờ làm máy chủ đối với máy khách và giả làm máy khách đối với máy chủ, trong khi người đứng giữa giải mã lưu lượng truy cập đến từ cả hai phía.

Phần khó khăn là hệ thống phát hành các chứng chỉ được thiết kế để ngăn chặn chính xác cuộc tấn công dạng này bằng cách cho phép bên thứ ba đáng tin cậy kí mã hóa các chứng chỉ của máy chủ để xác mình rằng chúng hợp pháp. Nếu chữ kí này không khớp hoặc đến từ một bên không đáng tin cậy, một ứng dụng máy khách an toàn sẽ đơn giản là ngắt kết nối và từ chối tiếp tục. Bất chấp nhiều thiếu sót của các hệ thống CA hiện nay, điều này gây khó khăn cho các nỗ lực MITM để kết nối TLS và phân tích. Để giải quyết vấn đề này, mitmproxy tự mình trở thành tổ chức phát hành chứng chỉ đáng tin cậy. Mitmproxy bao gồm một bộ triển khai CA đầy đủ để tạo chứng chỉ một cách nhanh chóng. Để các máy khách tin tưởng các chứng chỉ, mitmproxy được đăng kí làm CA đáng tin cậy với thiết bị theo cách thủ công.

2.2.3.1 Tên máy chủ từ xa là gì?

Chúng ta cần biết tên miền được sử dụng trong chứng chỉ bị chặn- ứng dụng máy khách sẽ xác mình rằng chứng chỉ miền mà nó đang kết nối và hủy bỏ nếu nó không đúng như vậy. Lúc đầu, có vẻ như yêu cầu CONNECT ở trên cung cấp cho chúng ta tất cả những gì chúng ta cần, trong ví dụ thì cả 2 giá trị đều là "example.com". Nhưng điều gì sẽ xảy ra nếu máy khách đã bắt đầu kết nối như sau:

CONNECT 10.1.1.1:443 HTTP/1.1

Sử dụng địa chỉ IP là hoàn toàn hợp pháp vì nó cung cấp đủ thông tin để thiết lập kết nối, mặc dù không tiết lộ tên máy chủ từ xa.

Mitmproxy có một cơ chế tinh vi làm trơn tru việc đánh hơi chứng chỉ này. Ngay sau khi thấy yêu cầu CONNECT, mitmproxy tạm dừng phần máy khách của cuộc trò chuyện và bắt đầu kết nối đồng thời với máy chủ. Mitmproxy hoàn thành quá trình bắt tay TLS với máy chủ và kiểm tra các chứng chỉ máy chủ đã sử dụng. Bây giờ sử dụng Common Name ở chứng chỉ gốc và tạo chứng chỉ giả cho máy khách. Vì vậy chúng ta có tên máy chủ chính xác để hiển thị cho máy khách, ngay cả khi nó chưa bao giờ được chỉ định.

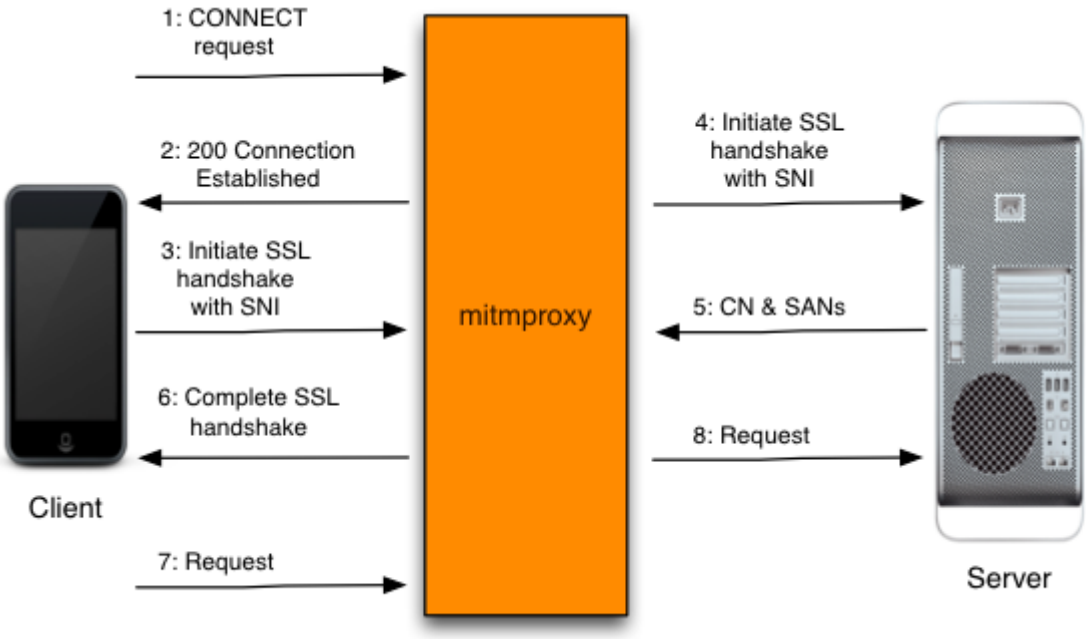
2.2.3.2 Subject Alternative Name?

Đôi khi trên thực tế, Common Name của chứng chỉ không phải là tên máy chủ mà máy khách đang kết nối. Điều này là do tùy chọn Subject Alternative Name trong chứng chỉ cho phép chỉ định một số miền thay thế tùy ý. Nếu miền dự kiến khớp với bất kỳ miền nào trong số này, máy khách sẽ tiếp tục, ngay cả khi miền không khớp với chứng chỉ CN. Cách giải quyết rất đơn giản: Khi trích xuất CN từ chứng chỉ chính, cũng trích xuất luôn SANs và thêm vào chứng chỉ giả đã tạo.

2.2.3.3 Server Name Indication

Một trong những hạn chế lớn của TLS là mỗi chứng chỉ yêu cầu địa chỉ IP riêng. Tức là ta không thể thực hiện lưu trữ ảo trong đó nhiều miền có chứng chỉ độc lập chia sẻ cùng một địa chỉ IP. Khi mà nhóm địa chỉ IPv4 đang bị thu hẹp nhanh chóng, mitmproxy có một giải pháp ở dạng Server Name Indication mở rộng cho các giao thức TLS. Điều này cho phép máy khách chỉ định tên máy chủ từ xa khi bắt đầu bắt tay TLS, sau đó cho phép máy chủ chọn chứng chỉ phù hợp để hoàn tất quá trình.

SNI phá vỡ quy trình dò tìm ngược chứng chỉ, bởi vì khi khi mitmproxy kết nối mà không sử dụng SNI, chúng ta sẽ nhận được chứng chỉ mặc định có thể không liên quan gì đến chứng chỉ mà khách mong đợi. Sau khi máy khách kết nối, mitmproxy cho phép bắt tay TLS tiếp tục cho đến khi giá trị SNI được chuyển. Bây giờ chúng ta đã có thể tạm dừng giao tiếp và bắt đầu kết nối với chứng chỉ chính bằng giá trị SNI chính xác, giá trị này sau đó cung cấp cho chúng ta chứng chỉ chính xác , từ đó trích xuất CN và SAN dự kiến.



Hình 2.2 Kết nối HTTPS Proxy.[9]

1. Máy khách kết nối tới mitmproxy, và đưa ra yêu cầu HTTP CONNECT.
2. Mitmproxy trả lời 200 Connection Established, như thể nó đã thiết lập một đường ống CONNECT.
3. Máy khách tin tưởng rằng nó đang giao tiếp với máy chủ từ xa và bắt đầu kết nối TLS. Nó đã sử dụng SNI để chỉ ra tên máy chủ mà nó đang kết nối.

4. Mitmproxy kết nối với máy chủ và thiết lập kết nối TLS bằng cách sử dụng tên máy chủ SNI do máy khách chỉ định.
5. Máy chủ phản hồi bằng chứng chỉ phù hợp, có chứa các giá trị CN và SAN cần thiết để tạo chứng chỉ nhận.
6. Mitmproxy tạo ra chứng chỉ chặn và tiếp tục bắt tay TLS với máy khách ở bước 3.
7. Máy khách gửi yêu cầu kết nối qua kết nối TLS đã thiết lập.
8. Mitmproxy chuyển yêu cầu đến máy chủ qua kết nối TLS được khởi tạo ở bước 4.

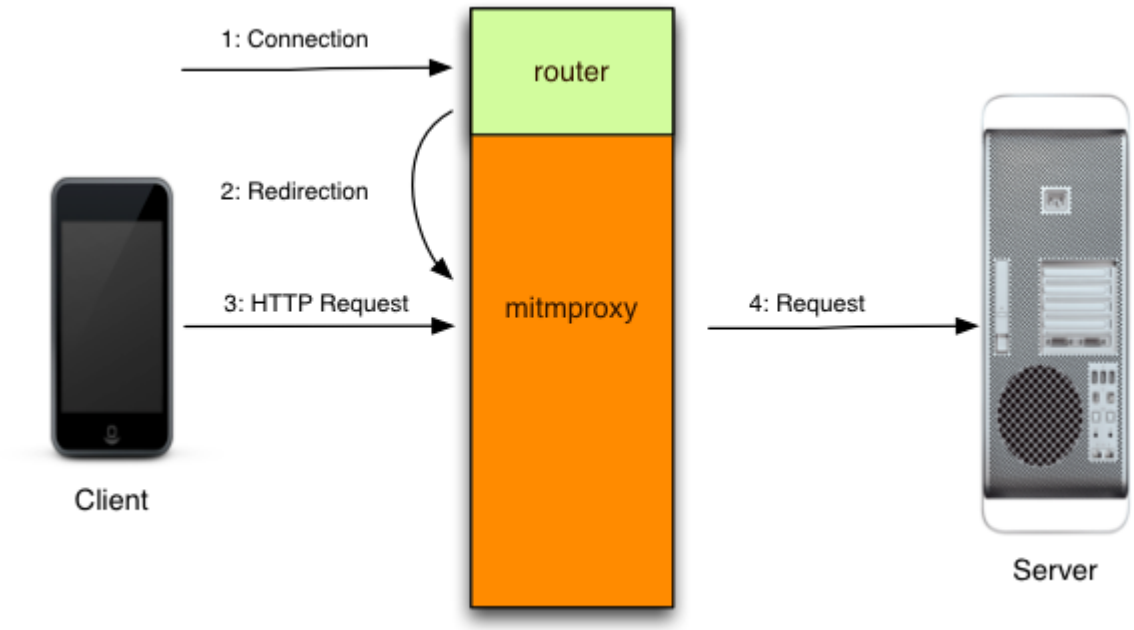
2.2.4 Transparent HTTP

Khi sử dụng proxy trong suốt với người dùng, kết nối được chuyển hướng thành proxy ở lớp mạng mà không cần bất kì cấu hình máy khách nào. Điều này làm cho proxy trở lên lí tưởng cho những trường hợp không thể thay đổi hành vi của máy khách.

Để đạt được điều này. Cơ chế chuyển hướng định tuyến lại các kết nối TCP dành cho một máy chủ trên internet đến một máy chủ proxy. Điều này thường có dạng tường lửa trên cùng một máy chủ như máy chủ proxy – iptables trên linux hoặc pf trên OSX. Khi máy khách đã bắt đầu kết nối, nó đưa ra một yêu cầu HTTP, có thể như sau:

GET /index.html HTTP/1.1

Lưu ý rằng yêu cầu này khác với biến thể proxy rõ ràng, nó bỏ qua lược đồ và tên máy chủ. Một module máy chủ lưu trữ biết cách truy xuất địa chỉ đích ban đầu từ bộ định tuyến. Trong mitmproxy là một tập hợp module được tích hợp sẵn để biết cách giao tiếp với cơ chế chuyển hướng của mỗi nền tảng.



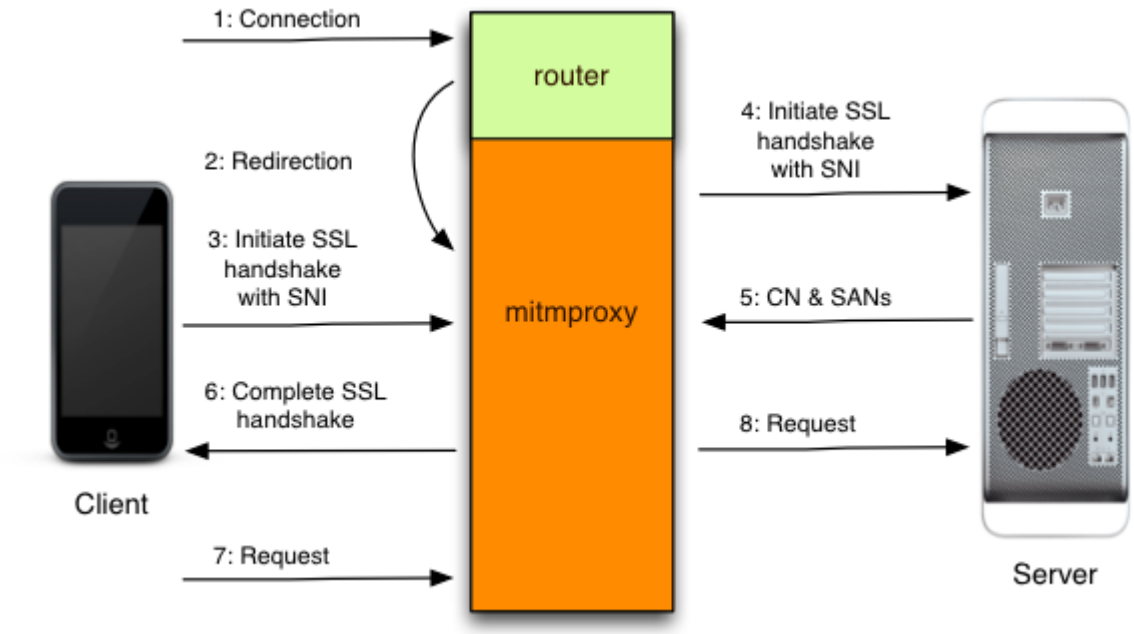
Hình 2.3 Transparent HTTP.[9]

1. Máy khách tạo kết nối tới máy chủ
2. Bộ định tuyến chuyển hướng kết nối đến mitmproxy, thường lắng nghe trên cùng một cổng của cùng máy chủ. Sau đó mitmproxy tham vấn cơ chế định tuyến để thiết lập điểm đến ban đầu.
3. Đọc yêu cầu từ máy khách
4. Chuyển tiếp đến Server chính.

2.2.5 Transparent HTTPS

Bước đầu tiên là xác định có phải kết nối HTTPS hay không. Sử dụng cơ chế định tuyến để tìm ra cổng đích ban đầu là gì. Tất các các kết nối đến đều đi qua các lớp khác nhau có thể xác định giao thức thực tế sẽ sử dụng.

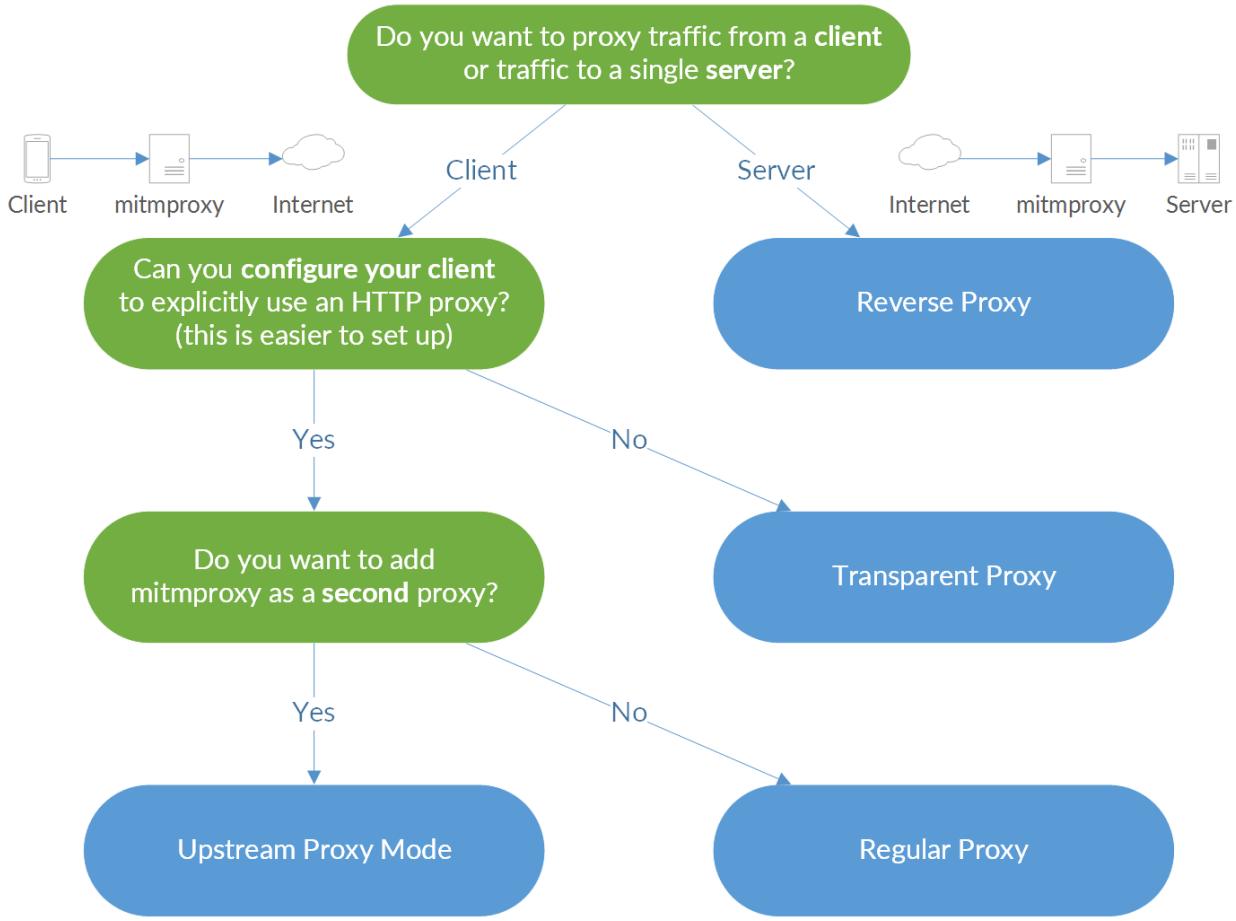
Từ đây quy trình là sự hợp nhất của các phương pháp đã mô tả để ủy quyền HTTP một cách minh bạch và ủy quyền HTTPS một cách rõ ràng. Sử dụng cơ chế định tuyến để thiết lập địa chỉ máy chủ chính, sau đó tiến hành như đối với kết nối HTTPS rõ ràng để thiết lập CN và SAN cũng như đối với SNI.



Hình 2.4 Transparent HTTPS.[9]

1. Máy khách tạo kết nối tới máy chủ
2. Bộ định tuyến chuyển hướng kết nối đến mitmproxy. Sau đó mitmproxy tham vấn cơ chế định tuyến để thiết lập điểm đến ban đầu.
3. Máy khách tin rằng đang giao tiếp với máy chủ từ xa và bắt đầu kết nối TLS. Nó sử dụng SNI để chỉ ra tên máy chủ mà nó đang kết nối.
4. Mitmproxy kết nối với máy chủ và thiết lập kết nối TLS bằng cách sử dụng tên máy chủ SNI do máy khách chỉ định .
5. Máy chủ phản hồi chứng chỉ phù hợp có chứa các giá trị CN và SAN cần thiết để tạo chứng chỉ chặn.
6. Mitmproxy tạo ra chứng chỉ và tiếp tục bắt tay TLS với máy khách ở bước 3.
7. Máy khách gửi yêu cầu qua kết nối TLS đã thiết lập.
8. Mitmproxy chuyển yêu cầu đến máy chủ qua kết nối TLS đã được khởi tạo ở bước 4.

2.3 Các chế độ hoạt động



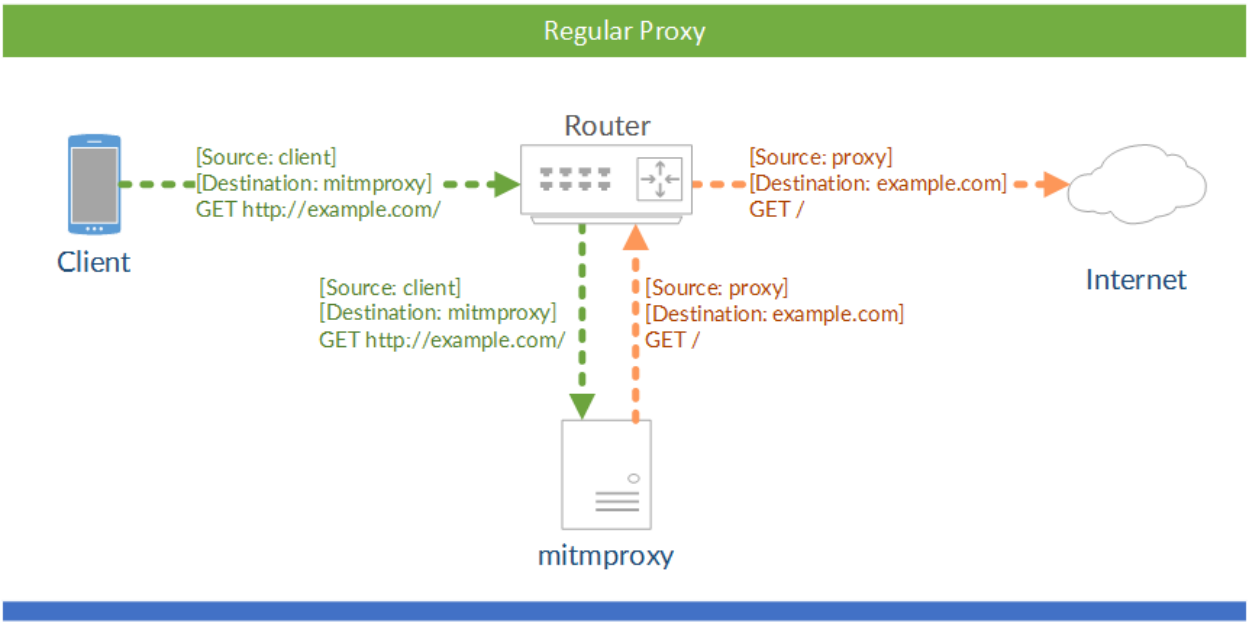
Hình 2.5 Khái quát các chế độ hoạt động của mitmproxy.[9]

2.3.1 Regular Proxy

1. Khởi động mitmproxy.
2. Thiết lập cấu hình máy khách để sử dụng mitmproxy. Mặc định, mitmproxy lắng nghe trên cổng 8080.
3. Kiểm tra nhanh: Truy cập một trang không mã hóa HTTP thông qua proxy.
4. Mở mitm.it và cài đặt chứng chỉ cho thiết bị

Một số ứng dụng bỏ qua cài đặt proxy HTTP của hệ thống - các ứng dụng Android là một ví dụ phổ biến. Trong những trường hợp này, ta cần sử dụng chế độ transparent của mitmproxy.

Nếu đang sử dụng proxy của một thiết bị bên ngoài, sơ đồ mạng có thể sẽ trông như thế này:



Hình 2.6 Ví dụ về mô hình mạng sử dụng Regular Proxy.[9]

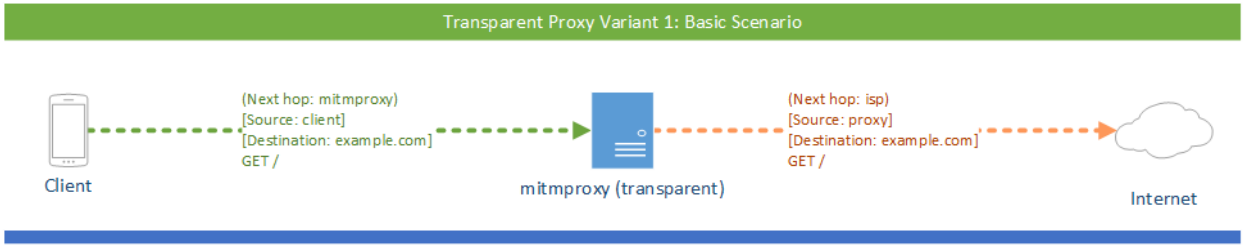
Dấu ngoặc vuông biểu thị địa chỉ IP nguồn và đích. Máy khách đang kết nối với mitmproxy và mitmproxy kết nối với máy chủ đích.

2.3.2 Transparent Proxy

Cách kích hoạt chế độ transparent:

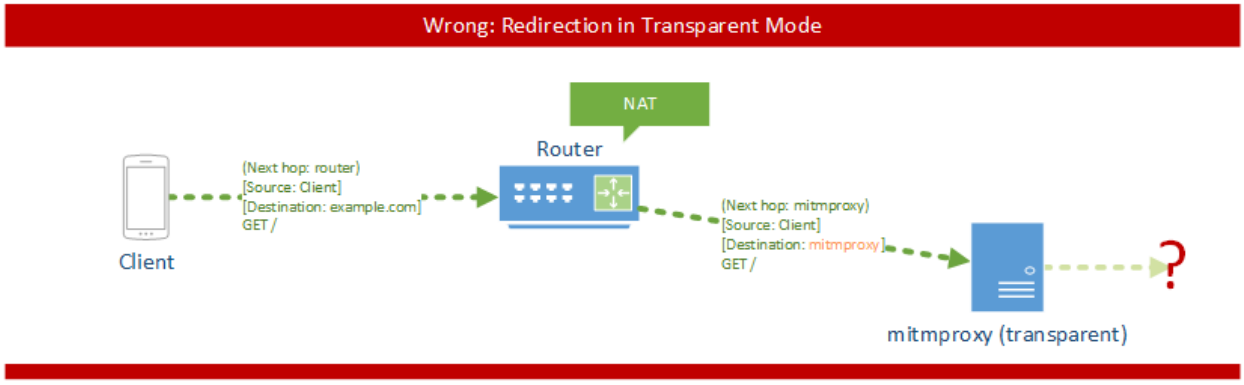
mitmdump –mode transparent

Trong chế độ này, lưu lượng truy cập được chuyển hướng vào một proxy ở lớp mạng mà không cần bất kỳ cấu hình máy khách nào. Điều này làm cho ủy quyền minh bạch trở nên lý tưởng cho những tình huống mà bạn không thể thay đổi hành vi của máy khách. Trong hình bên dưới, một máy chạy mitmproxy đã được chèn giữa bộ định tuyến và Internet:



Hình 2.7 Transparent proxy.[9]

Dấu ngoặc vuông biểu thị địa chỉ IP nguồn và đích. Dấu ngoặc tròn đánh dấu bước tiếp theo trên lớp liên kết dữ liệu / Ethernet. Sự phân biệt này rất quan trọng: khi gói tin đến máy mitmproxy, nó vẫn phải được gửi tới máy chủ đích. Điều này có nghĩa là không nên áp dụng NAT trước khi lưu lượng đến mitmproxy, vì điều này sẽ xóa thông tin đích, khiến mitmproxy không thể xác định đích thực.

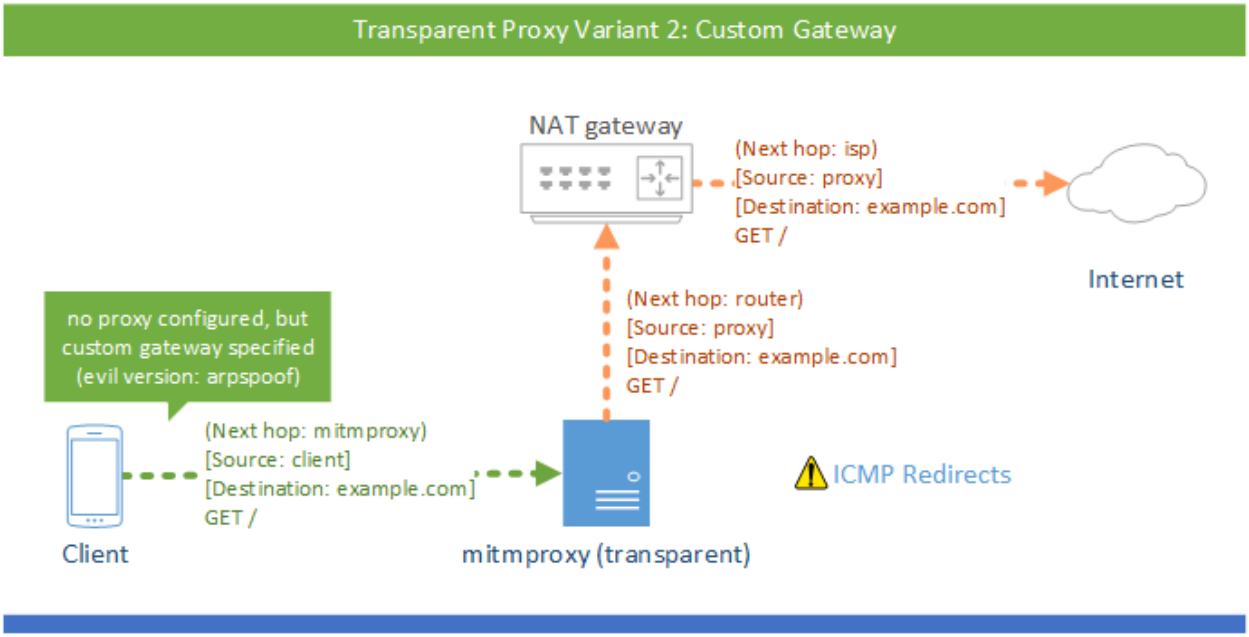


Hình 2.8 NAT trước mitmproxy trong chế độ transparent .[9]

2.3.2.1 Định cấu hình máy khách để sử dụng cổng / bộ định tuyến tùy chỉnh / "next hop"

Một cách đơn giản để có được lưu lượng truy cập vào máy mitmproxy với IP đích còn nguyên vẹn, là chỉ cần cấu hình máy khách với hộp mitmproxy làm cổng mặc định.

Việc đặt cổng tùy chỉnh trên máy khách có thể được tự động hóa bằng cách cung cấp các cài đặt cho máy khách qua DHCP. Điều này cho phép thiết lập một mạng đánh chặn nơi tất cả các máy khách được ủy quyền tự động, có thể tiết kiệm thời gian và công sức.

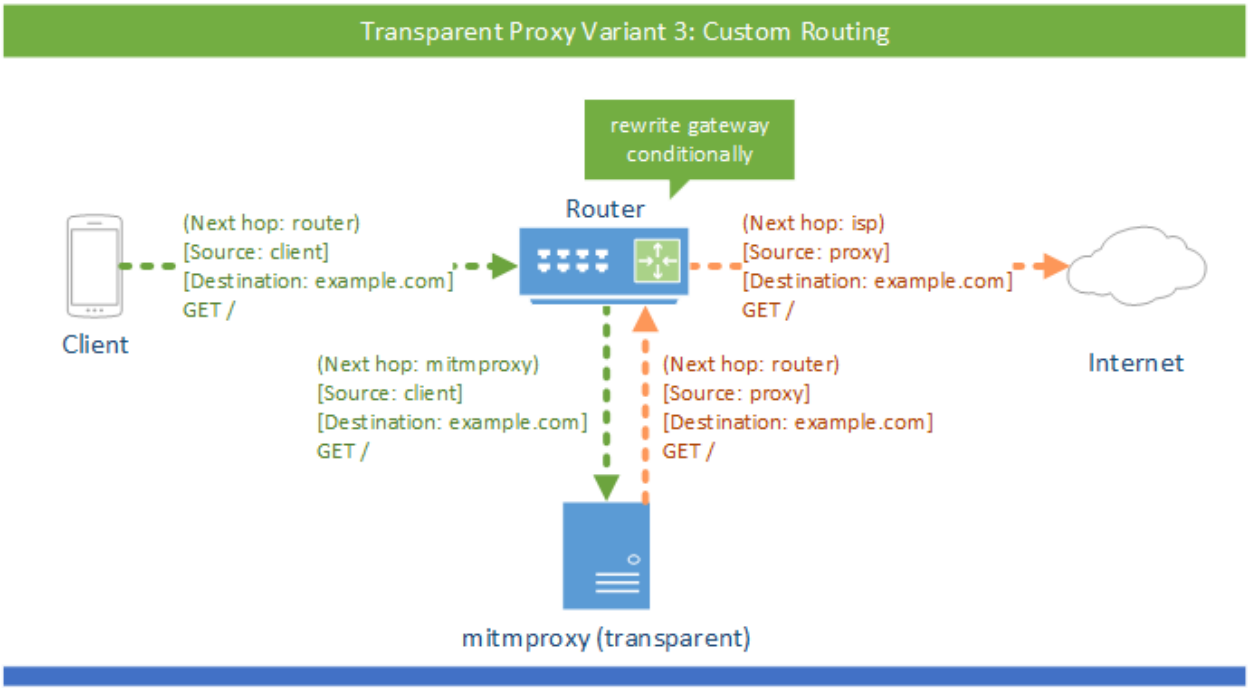


Hình 2.9 Cấu hình máy khách để sử dụng transparent.[9]

1. Định cấu hình máy proxy cho chế độ transparent.
2. Định cấu hình máy khách để sử dụng IP của máy proxy làm cổng mặc định.
3. Kiểm tra : Có thể truy cập trang web HTTP không được mã hóa qua proxy.
4. Truy cập mitm.it và cài đặt chứng chỉ cho thiết bị.

2.3.2.2 Thực hiện định tuyến tùy chỉnh trên bộ định tuyến

Trong một số trường hợp, ta cần kiểm soát chi tiết hơn đối với lưu lượng truy cập đến phiên bản mitmproxy và lưu lượng nào không. Ví dụ: có thể chọn chỉ chuyển hướng lưu lượng truy cập đến một số máy chủ sang transparent proxy. Có rất nhiều cách để thực hiện điều này và phần lớn sẽ phụ thuộc vào bộ định tuyến hoặc bộ lọc gói mà đang sử dụng. Trong hầu hết các trường hợp, cấu hình sẽ như thế này:



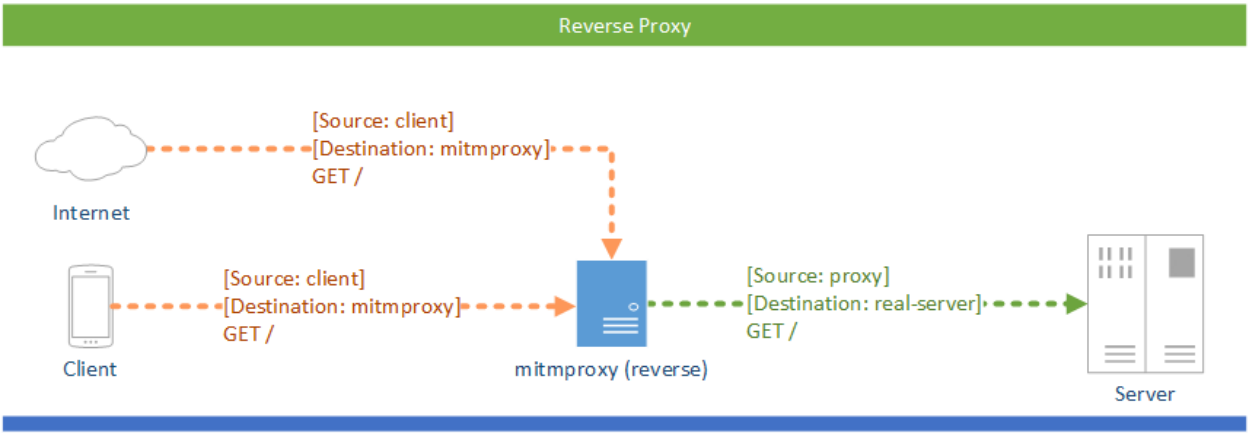
Hình 2.10 Cấu hình Router để sử dụng mitmproxy transparent.[9]

2.3.3 Reverse Proxy

Thiết lập:

mitmdump --mode reverse:https://example.com

mitmproxy thường được sử dụng với một máy khách sử dụng proxy để truy cập Internet. Sử dụng chế độ Reverse Proxy chúng ta có thể sử dụng mitmproxy để hoạt động như một máy chủ HTTP bình thường:



Hình 2.11 Reverse Proxy.[9]

Một vài trường hợp sử dụng:

- Giả sử chúng ta có một API nội bộ đang chạy tại http: //example.local/. Có thể thiết lập mitmproxy ở chế độ reverse proxy tại http://debug.example.local/ và tự động trỏ máy khách đến điểm cuối API mới này, cung cấp cho họ cùng một dữ liệu và chúng ta có thông tin để gỡ lỗi. Tương tự, chúng ta có thể di chuyển máy chủ thực của mình sang một IP / cổng khác và thiết lập mitmproxy ở vị trí ban đầu để gỡ lỗi và hoặc chuyển hướng tất cả các phiên.
- Giả sử nhà phát triển web đang làm việc trên http://example.com/ (với phiên bản phát triển chạy trên http://localhost: 8000 /). Nhà phát triển có thể sửa đổi tệp máy chủ của mình để example.com trỏ đến 127.0.0.1 và sau đó chạy mitmproxy ở chế độ reverse proxy trên cổng 80. Nhờ vậy có thể kiểm tra ứng dụng trên miền example.com và nhận tất cả các yêu cầu được ghi lại trong mitmproxy.
- Giả sử chúng ta có một số dự án trò chơi cần được hỗ trợ SSL. Chỉ cần thiết lập mitmproxy làm reverproxy trên cổng 443 và bạn đã hoàn tất (mitmdump -p 443 –mode reverseproxy: http://localhost:80/). Mitmproxy tự động phát hiện lưu lượng truy cập TLS và tự động chặn lưu lượng đó. Có nhiều công cụ tốt hơn cho nhiệm vụ cụ thể này, nhưng mitmproxy là cách rất nhanh chóng và đơn giản để thiết lập một máy chủ sủ dụng SSL.
- Khi muốn thêm proxy nén không hỗ trợ SSL vào phía trước máy chủ của mình? có thể tạo ra một phiên bản mitmproxy không SSL (--mode reverse: http: // ...), trỏ nó tới proxy nén và để proxy nén trỏ đến một mitmproxy khởi tạo SSL (--mode reverse: https : // ...), sau đó trỏ đến máy chủ thực.

2.3.3.1 Host Header

Trong chế độ reverse proxy, mitmproxy tự động ghi lại host header để khớp với upstream server. Điều này cho phép mitmproxy dễ dàng kết nối với các điểm cuối hiện đang mở trên web(ví dụ: mitmproxy --mode reverse: https: //example.com). Vô hiệu hóa hành vi này bằng tùy chọn keep_host_header.

Tuy nhiên, hãy nhớ rằng các URL tuyệt đối trong dữ liệu trả về hoặc chuyển hướng HTTP sẽ không được viết lại bằng mitmproxy. Điều này có nghĩa là nếu nhấp vào liên kết "http://example.com" trong trang web trả về, chúng ra sẽ được đưa trực tiếp đến URL đó, bỏ qua mitmproxy.

Một cách có thể để giải quyết vấn đề này là sửa đổi tệp máy chủ của hệ điều hành để "example.com" phân giải thành IP của proxy, sau đó truy cập proxy bằng cách truy cập trực tiếp vào example.com. Đảm bảo rằng proxy của vẫn có thể phân giải IP gốc hoặc chỉ định IP trong mitmproxy.

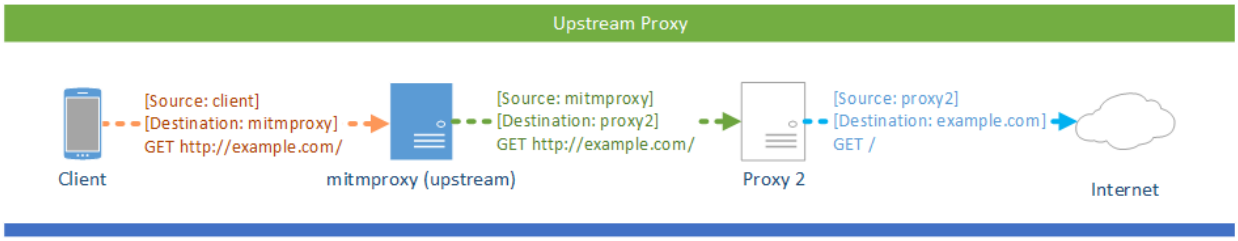
Chế độ Reverse Proxy thường không đủ để tạo bản sao của một trang web tương tác tại URL khác. HTML được cung cấp cho khách hàng vẫn không thay đổi - ngay sau khi người dùng nhấp vào một URL không tương đối (hoặc tải xuống tài nguyên hình ảnh không tương đối), lưu lượng truy cập không còn đi qua mitmproxy nữa.

2.3.4 Upstream Proxy

Cách thiết lập:

mitmdump --mode upstream:http://example.com:8081

Nếu chúng ta muốn sử dụng chuỗi proxy bằng cách thêm mitmproxy vào trước một thiết bị proxy khác, ta có thể sử dụng chế độ upstream của mitmproxy. Ở chế độ upstream , tất cả các yêu cầu được chuyển hướng vô điều kiện tới một proxy upstream được chỉ định.



Hình 2.12 Upstream Proxy[9]

Mitmproxy hỗ trợ cả HTTP và HTTPS trong chế độ upstream proxy.

2.3.5 SOCKS Proxy

Cách thiết lập :

mitmdump --mode socks5

Trong chế độ này, mitmproxy hoạt động như một proxy SOCKS5. Điều này tương tự như chế độ proxy thông thường, nhưng sử dụng SOCKS5 thay vì HTTP để thiết lập kết nối với proxy.

Chương 3: Chứng chỉ, đặc trưng, tiện ích mở rộng trên mitmproxy

3.1 Chứng chỉ

Mitmproxy có thể giải mã lưu lượng được mã hóa một cách nhanh chóng, miễn là máy khách tin tưởng tổ chức phát hành chứng chỉ tích hợp của mitmproxy. Thông thường, điều này có nghĩa là chứng chỉ mitmproxy CA phải được cài đặt trên thiết bị máy khách.

Cách đơn giản nhất để cài đặt chứng chỉ mitmproxy CA là sử dụng ứng dụng cài đặt chứng chỉ tích hợp sẵn. Để thực hiện việc này, bắt đầu mitmproxy và định cấu hình thiết bị mục tiêu với cài đặt proxy chính xác. Khởi động trình duyệt trên thiết bị và truy cập mitm.it để cài đặt.

3.1.1 Cơ quan cấp chứng chỉ mitmproxy

Lần đầu mitmproxy được chạy, nó tạo các khóa cho tổ chức phát hành chứng chỉ (CA) trong thư mục cấu hình (~ / .mitmproxy theo mặc định). CA này được sử dụng để tạo nhanh các chứng chỉ giả cho mỗi trang web đã truy cập. Vì trình duyệt của sẽ không tin tưởng vào mitmproxy CA nên bạn sẽ cần phải nhấp qua cảnh báo chứng chỉ TLS trên mọi miền hoặc cài đặt chứng chỉ CA một lần để nó được tin cậy.

Bảng 3.1 Các tệp chứng chỉ được tạo.

Tên file	Nội dung
Mitmproxy-ca.pem	Chứng chỉ và khóa riêng ở định dạng PEM
Mitmproxy-ca-cert.pem	Chứng chỉ ở dạng PEM sử dụng phân phối trên các nền tảng không phải windows
Mitmproxy-ca-cert.p12	Chứng chỉ dạng PKCS12 sử dụng cho Windows.
Mitmproxy-ca-cert.cer	Giống file pem nhưng với phần mở rộng cho các thiết bị Android.

Vì lý do bảo mật, mitmproxy CA được tạo duy nhất trong lần khởi động đầu tiên và không được chia sẻ giữa các cài đặt mitmproxy trên các thiết bị khác nhau. Điều này đảm bảo rằng những người dùng mitmproxy khác không thể chặn lưu lượng truy cập của nhau.

3.1.2 Upstream Certificate Sniffing

Khi mitmproxy nhận được yêu cầu thiết lập TLS (dưới dạng thông báo ClientHello), nó sẽ đặt máy khách ở trạng thái chờ và đầu tiên tạo kết nối với máy chủ chính để đánh hơi nội dung chứng chỉ TLS của nó. Thông tin thu được có thể bao gồm: Common Name, Organization, Subject Alternative Names- sau đó được sử dụng để tạo chứng chỉ đánh chặn mới một cách nhanh chóng, được ký bởi mitmproxy CA. Sau đó, Mitmproxy trở lại máy khách và tiếp tục bắt tay với chứng chỉ mới được giả mạo. Tính năng dò tìm chứng chỉ chính được bật theo mặc định và có thể tắt tùy chọn bằng cách tắt tùy chọn upstream_cert.

3.1.3 Certificate Pinning

Một số ứng dụng sử dụng tính năng ghim chứng chỉ để ngăn chặn các cuộc tấn công man-in-the-middle. Các chứng chỉ của mitmproxy sẽ không được các ứng dụng này chấp nhận nếu không sửa đổi chúng. Nếu nội dung của các kết nối không quan trọng, nên sử dụng ignore_hosts để ngăn mitmproxy chặn các lưu lượng truy cập đến các miền cụ thể này. Nếu muốn chặn các kết nối đã bị ghim cần phải sửa ứng dụng theo cách thủ công.

3.1.4 Sử dụng chứng chỉ từ server tùy chỉnh

Có thể sử dụng chứng chỉ của riêng mình bằng cách chuyển tùy chọn –certs [domain=] path_to_certificate tới mitmproxy. Sau đó, mitmproxy sử dụng chứng chỉ được cung cấp để chặn miền chỉ định thay vì tạo chứng chỉ do CA của chính nó kí.

Tệp chứng chỉ được mong đợi ở dạng PEM. Có thể bao gồm các chứng chỉ trung gian ngay bên dưới chứng chỉ của mình, tệp PEM có thể gần giống như sau:

-----BEGIN PRIVATE KEY-----

<private key>

-----END PRIVATE KEY-----

-----BEGIN CERTIFICATE-----

<cert>

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

<intermediary cert (optional)>

-----END CERTIFICATE-----

Ví dụ: có thể tạo chứng chỉ ở định dạng này bằng cách sử dụng các hướng dẫn sau:

openssl genrsa -out cert.key 2048

*# (Specify the mitm domain as Common Name, e.g. *.google.com)*

openssl req -new -x509 -key cert.key -out cert.crt

cat cert.key cert.crt > cert.pem

Bây giờ, ta có thể chạy mitmproxy với chứng chỉ đã tạo

Cho tất cả các tên miền :

mitmproxy --certs *=cert.pem

Cho một số tên miền cụ thể:

mitmproxy --certs *.example.com=cert.pem

3.1.5 Sử dụng CA tùy chỉnh

Theo mặc định, mitmproxy sẽ sử dụng ~ / .mitmproxy / mitmproxy-ca.pem làm tổ chức phát hành chứng chỉ để tạo chứng chỉ cho tất cả các miền mà không có chứng chỉ tùy chỉnh nào được cung cấp (xem ở trên). Bạn có thể sử dụng tổ chức phát hành chứng chỉ của riêng mình bằng cách chuyển tùy chọn --set confdir = DIRECTORY tới mitmproxy. Sau đó, Mitmproxy sẽ tìm kiếm mitmproxy-ca.pem trong thư mục được chỉ định. Nếu không có tệp nào như vậy tồn tại, nó sẽ được tạo tự động.

Tệp chứng chỉ mitmproxy-ca.pem giống như sau:

-----BEGIN PRIVATE KEY-----

<private key>

-----END PRIVATE KEY-----

-----BEGIN CERTIFICATE-----

<cert>

-----END CERTIFICATE-----

Khi xem chứng chỉ với openssl x509 -noout -text -in ~ / .mitmproxy / mitmproxy-ca.pem, nó phải có ít nhất các phần mở rộng X509v3 sau để mitmproxy có thể sử dụng nó để tạo chứng chỉ:

X509v3 extensions:

X509v3 Key Usage: critical

Certificate Sign

X509v3 Basic Constraints: critical

CA:TRUE

3.1.7 Sử dụng chứng chỉ bên phía máy khách

Có thể sử dụng chứng chỉ ứng dụng máy khách bằng cách chuyển tùy chọn --set client_certs = DIRECTORY | FILE vào mitmproxy. Sử dụng thư mục cho phép chọn chứng chỉ dựa trên tên máy chủ, trong khi sử dụng tên tệp cho phép một chứng chỉ cụ thể duy nhất được sử dụng cho tất cả các kết nối TLS. Tệp chứng chỉ phải ở định dạng PEM và phải chứa cả khóa cá nhân không được mã hóa và chứng chỉ.

3.1.8 Nhiều chứng chỉ máy khách

Có thể chỉ định một thư mục cho --set client_certs = DIRECTORY, trong trường hợp này, chứng chỉ phù hợp được tra cứu theo tên tệp. Vì vậy, nếu truy cập example.org, mitmproxy sẽ tìm kiếm tệp có tên example.org.pem trong thư mục được chỉ định và sử dụng tệp này làm chứng chỉ ứng dụng máy khách.

3.2 Các đặc trưng

3.2.1 Anticache

Khi tùy chọn anticache được chọn, nó sẽ loại bỏ tiêu đề (if-none-match và if-modified-since) điều đó có thể dẫn đến phản hồi 304 Not Modified từ máy chủ. Điều này hữu ích khi muốn đảm bảo rằng người sử dụng nắm bắt toàn bộ một trao đổi HTTP. Nó cũng thường được sử dụng trong quá trình phát lại phía máy khách, khi muốn đảm bảo máy chủ phản hồi với dữ liệu đầy đủ.

3.2.2 Blocklist

Sử dụng tùy chọn block_list, có thể chặn các yêu cầu trang web hoặc yêu cầu cụ thể. Thay vào đó mitmproxy trả về mã trạng thái HTTP cố định hoặc hoàn toàn không phản hồi.

Một mẫu block_list có thể như sau:

/flow-filter/status-code

Flow-filter là một biểu thức bộ lọc mitmproxy tùy chọn mô tả yêu cầu nào nên bị chặn.

Status-code là mã trạng thái HTTP được phân phối bởi mitmproxy cho các yêu cầu bị chặn. Một mã trạng thái đặc biệt là 444 hướng dẫn mitmproxy bị treo và không gửi bất cứ phản hồi nào.

Bảng 3.2 Ví dụ blocklist.

:~d google-analytics.com:404	Chặn tất cả các yêu cầu đến google-analytics.com và trả về " 404 not found"
:~d example.com$:444	Chặn tất cả các yêu cầu tới example.com và không phản hồi gì.
:!~d ^example\.com$:403	Chỉ cho phép các yêu cầu HTTP đến example.com . Điều này không an toàn vì có thể bị bỏ qua bằng cách sử dụng giao thực khác không phải HTTP.

3.2.3 Client-side replay

Cung cấp một cuộc giao tiếp HTTP đã lưu trước đó và mitmproxy phát lại lần lượt từng yêu cầu của máy khách. Mitmproxy tuần tự các yêu cầu, chờ phản hồi từ máy chủ trước khi bắt đầu các yêu cầu tiếp theo. Khác với giao tiếp được ghi lại, nơi các yêu cầu có thể được thực hiện đồng thời.

Có thể sử dụng tính năng Client-side replay kết hợp anticache để đảm bảo máy chủ phản hồi với dữ liệu đầy đủ.

3.2.4 Map Local

Tùy chọn map_local cho phép bạn chỉ định một số mẫu tùy ý xác định chuyển hướng của các yêu cầu HTTP đến các tệp hoặc thư mục cục bộ. Tệp cục bộ được tìm nạp thay vì tài nguyên ban đầu và được trả lại cho máy khách một cách trong suốt với người dùng.

Mẫu map_local có thể như sau:

|url-regex|local-path

|flow-filter|url-regex|local-path

Local-path: là tệp hoặc thư mục sẽ được cung cấp cho máy khách.

url-regex: là một biểu thức chính quy được áp dụng trên url yêu cầu. Nó phải khớp để chuyển hướng được diễn ra.

Flow-fillter: là một biểu thức bộ lọc mitmproxy tùy chọn hạn chế thêm yêu cầu nào sẽ được chuyển hướng.

Bảng 3.3 Ví dụ về map local.

|example.com/main.js|~/main-local.js	Thay thế example.com/main.js bằng ~/main-local.js
|example.com/static|~/static	Thay thế example.com/static/foo/bar.css bằng ~ / static / foo / bar.css
|example.com/static/foo|~/static	Thay thế example.com/static/foo/bar.css bằng ~ / static / bar.css.
|~m GET|example.com/static|~/static	Thay thế example.com/static/foo/bar.css bằng ~ / static / foo / bar.css (nhưng chỉ dành cho các yêu cầu GET).

3.2.5 Detail

Nếu local-path là một tệp, tệp này sẽ luôn được cung cấp. Thay đổi tệp sẽ được phản ánh ngay lập tức, không có bộ nhớ đệm.

Nếu local-path là một thư mục, url-regex được sử dụng để chia URL yêu cầu thành hai phần và một phần ở bên phải được nối vào local-path, ngoại trừ chuỗi truy vấn. Tuy nhiên, nếu url-regex chứa một nhóm thu thập regex, hành vi này sẽ thay đổi và thay vào đó, nhóm thu thập đầu tiên sẽ được thêm vào (và các chuỗi truy vấn không bị loại bỏ). Các ký tự đặc biệt được ánh xạ tới _. Nếu không tìm thấy tệp, /index.html sẽ được thêm vào và chúng tôi thử lại. Không thể duyệt qua thư mục bên ngoài thư mục được chỉ định ban đầu.

Để minh họa điều này, hãy xem xét ví dụ sau đây ánh xạ tất cả các yêu cầu cho example.org/css* tới thư mục cục bộ ~ / static-css.

```
                  ┌── url regex ──┬─ local path ─┐
map_local option: |example.com/css|~/static-css
```

```
                              |
                              |    URL is split here
                              ▼           ▼
HTTP Request URL: https://example.com/css/print/main.css?timestamp=123
                                                  |              ▼
                                                  ▼              query string
is ignored
Served File:      Preferred: ~/static-css/print/main.css
                   Fallback: ~/static-css/print/main.css/index.html
                  Otherwise: 404 response without content
```

Nếu tệp phụ thuộc vào chuỗi truy vấn, chúng ta có thể sử dụng các nhóm regex. Trong ví dụ này, tất cả các yêu cầu GET cho example.org/index.php?page= <page-name> được ánh xạ tới ~ / static-dir / <page-name>:

```
                     flow
                  ┌filter┐──────────── url regex ──────────┌─ local path ─┐
map_local option: |~m GET|example.com/index.php\\?page=(.+)|~/static-dir
                              |                          |
                              |                          | regex group =
suffix
                              ▼                          ▼
HTTP Request URL: https://example.com/index.php?page=aboutus
                                                         |
                                                         ▼
Served File:                  Preferred: ~/static-dir/aboutus
                               Fallback: ~/static-dir/aboutus/index.html
                              Otherwise: 404 response without content
```

3.2.6 Map remote

Tùy chọn map_remote cho phép chỉ định một số mẫu tùy ý xác định các thay thế trong URL yêu cầu HTTP trước khi chúng được gửi đến máy chủ. URL được thay

thế được tìm nạp thay vì tài nguyên ban đầu và phản hồi HTTP tương ứng được trả lại một cách minh bạch cho máy khách. Lưu ý rằng nếu đích ban đầu sử dụng HTTP2, thì đích được thay thế cũng cần hỗ trợ HTTP2, nếu không, yêu cầu được thay thế có thể không thành công. Một giải pháp khác là bạn có thể bắt đầu mitmproxy với cờ --no-http2 để tắt HTTP2. các mẫu map_remote trông như thế này:

```
|flow-filter|url-regex|replacement

|url-regex|replacement
```

flow-filter là một biểu thức bộ lọc mitmproxy tùy chọn xác định yêu cầu nào mà tùy chọn map_remote áp dụng.

url-regex là một biểu thức chính quy Python hợp lệ xác định những gì được thay thế trong URL của yêu cầu.

thay thế là một ký tự chuỗi được thay thế trong.

Vd: Ánh xạ tất cả các yêu cầu kết thúc bằng .jpg tới https://placedog.net/640/480?random. Lưu ý rằng điều này có thể không thành công nếu đích yêu cầu HTTP ban đầu sử dụng HTTP2 nhưng đích thay thế không hỗ trợ HTTP2.

```
      |.*\.jpg$|https://placedog.net/640/480?random
```

Định tuyến lại tất cả các yêu cầu GET từ example.org đến mitmproxy.org (sử dụng | làm dấu phân tách):

```
      |~m GET|//example.org/|//mitmproxy.org/
```

3.2.7 Modify Body

Tùy chọn modify_body cho phép bạn chỉ định một số lượng mẫu tùy ý để xác định các thay thế bên trong các phần nội dung của luồng. Các mẫu modify_body trông như thế này:

```
/flow-filter/body-regex/replacement

/flow-filter/body-regex/@file-path

/body-regex/replacement

/body-regex/@file-path
```

flow-filter là một biểu thức bộ lọc mitmproxy tùy chọn xác định dòng thay thế áp dụng cho.

body-regex là một biểu thức chính quy hợp lệ trong Python xác định những gì được thay thế.

Replace là một ký tự chuỗi được thay thế. Nếu ký tự thay thế bắt đầu bằng @ như trong @ file-path, thì nó được coi là một đường dẫn tệp mà từ đó thay thế được đọc.

Sửa đổi hooks kích hoạt khi nhận được yêu cầu của khách hàng hoặc phản hồi của máy chủ. Chỉ thành phần luồng phù hợp mới bị ảnh hưởng: vì vậy, ví dụ: nếu một hook được kích hoạt trên phản hồi của máy chủ, thì việc thay thế chỉ được chạy trên đối tượng Phản hồi, giữ nguyên Yêu cầu. Kiểm soát việc hook kích hoạt theo yêu cầu, phản hồi hay cả hai bằng cách sử dụng mẫu bộ lọc. Nếu cần kiểm soát chi tiết hơn mức này, thật đơn giản để tạo một tập lệnh bằng cách sử dụng API thay thế trên các Flow Component.

Vd

Thay thế foo bằng bar trong yêu cầu:

```
/~q/foo/bar
```

Thay thế foo bằng dữ liệu đọc từ ~/xss-exploit

```
mitmdump --modify-body :~q:foo:@~/xss-exploit
```

3.2.8 Modify Headers

Tùy chọn mod_headers cho phép chỉ định một tập hợp các tiêu đề sẽ được sửa đổi. Các tiêu đề mới có thể được thêm vào và các tiêu đề hiện có có thể được ghi đè hoặc xóa. Các mẫu mod_headers như sau:

```
/flow-filter/name/value

/flow-filter/name/@file-path

/name/value

/name/@file-path
```

flow-filter là một biểu thức lọc mitmproxy tùy chọn xác định luồng nào để sửa đổi các tiêu đề.

name là tên tiêu đề được đặt, thay thế hoặc loại bỏ.

giá trị là giá trị tiêu đề được đặt hoặc thay thế. Giá trị trống sẽ xóa các tiêu đề hiện có với tên. Nếu chuỗi giá trị bắt đầu bằng chữ @ như trong @ tệp-đường dẫn, thì nó được coi là đường dẫn tệp mà từ đó thay thế được đọc.

Các tiêu đề hiện tại được ghi đè theo mặc định. Điều này có thể được thay đổi bằng cách sử dụng một biểu thức bộ lọc.

VD

Đặt tiêu đề Host lưu trữ thành example.org cho tất cả các yêu cầu (tiêu đề Host lưu trữ hiện có được thay thế):

```
/~q/Host/example.org
```

Đặt tiêu đề Host lưu trữ thành example.org cho tất cả các yêu cầu không có tiêu đề Host lưu trữ hiện có:

```
/~q & !~h Host:/Host/example.org
```

Đặt tiêu đề User-Agent thành dữ liệu được đọc từ ~ / useragent.txt cho tất cả các yêu cầu (tiêu đề User-Agent có được thay thế):

```
/~q/Host/@~/useragent.txt
```

Loại bỏ tất cả tiêu đề Host từ tất cả các yêu cầu:

```
/~q/Host/
```

3.2.9 Proxy Authentication

Tùy chọn ủy quyền yêu cầu người dùng xác thực trước khi họ được phép sử dụng ủy quyền. Các tiêu đề xác thực bị loại bỏ khỏi các luồng, vì vậy chúng không được chuyển đến các máy chủ upstream. Hiện tại, chỉ có Xác thực cơ bản HTTP được hỗ trợ. Xác thực proxy không hoạt động tốt trong chế độ transparent proxy theo thiết kế vì máy khách không biết rằng nó đang giao tiếp với proxy. Mitmproxy sẽ yêu cầu lại thông tin đăng nhập cho mọi miền riêng lẻ. Xác thực proxy SOCKS hiện là chưa hoàn thành.

3.2.10 Server-side replay

Tùy chọn server_replay cho phép mitmproxy phát lại phản hồi của máy chủ từ các cuộc hội thoại HTTP đã lưu. Để thực hiện việc này, mitmproxy sử dụng một tập hợp các phương pháp phỏng đoán để đối chiếu các yêu cầu đến với các phản hồi đã lưu. Theo mặc định, mitmproxy loại trừ các tiêu đề yêu cầu khi đối sánh các yêu cầu đến với phản hồi từ tệp phát lại và chỉ sử dụng URL và phương thức yêu cầu để đối sánh. Điều này hoạt động trong hầu hết các trường hợp và giúp người dùng có thể

phát lại phản hồi của máy chủ trong các tình huống mà tiêu đề yêu cầu sẽ thay đổi một cách tự nhiên, ví dụ: sử dụng tác nhân người dùng khác.

Có rất nhiều cách để tùy chỉnh phù hợp, bao gồm chỉ định tiêu đề để bao gồm, yêu cầu tham số để loại trừ, v.v. Các tùy chọn này được thu thập dưới tiền tố server_replay.

3.2.11 Response refreshing

Việc chỉ phát lại các phản hồi của máy chủ mà không sửa đổi thường sẽ dẫn đến hành vi không mong muốn. Ví dụ: thời gian chờ cookie trong tương lai tại thời điểm cuộc trò chuyện được ghi lại có thể ở trong quá khứ tại thời điểm cuộc trò chuyện được phát lại. Theo mặc định, mitmproxy làm mới các phản hồi của máy chủ trước khi gửi đến máy khách. Ngày hết hạn và các tiêu đề được sửa đổi lần cuối đều được cập nhật để có cùng thời gian bù trừ với thời điểm ghi. Vì vậy, nếu các phản hồi ở quá khứ vào thời điểm ghi lại, các phản hồi sẽ ở quá khứ vào thời điểm phát lại, và ngược lại. Thời gian hết hạn cookie được cập nhật theo cách tương tự.

Có thể tắt hành vi này bằng cách đặt tùy chọn server_replay_refresh thành false.

3.2.12 Sticky auth

Tùy chọn stick auth tương tự như tùy chọn sticky cookie, trong đó tiêu đề Authorization chỉ được phát lại đến máy chủ sau khi chúng được nhìn thấy. Điều này đủ để cho phép bạn truy cập tài nguyên máy chủ bằng xác thực HTTP Basic thông qua proxy. Lưu ý rằng mitmproxy không (chưa) hỗ trợ phát lại xác thực Digest HTTP.

3.2.13 Sticky cookies

Khi tùy chọn stickcookie được đặt, mitmproxy sẽ thêm cookie được máy chủ đặt gần đây nhất vào bất kỳ yêu cầu cookie nào. Hãy xem xét một dịch vụ đặt cookie để theo dõi phiên sau khi xác thực. Sử dụng sticky cookie, người dùng có thể kích hoạt mitmproxy và xác thực dịch vụ như khi thường làm khi sử dụng trình duyệt. Sau khi xác thực, ta có thể yêu cầu các tài nguyên đã được xác thực thông qua mitmproxy như thể chúng chưa được xác thực, vì mitmproxy sẽ tự động thêm cookie theo dõi phiên vào các yêu cầu. Trong số những thứ khác, điều này cho phép người dùng tạo kịch bản cho các tương tác với các tài nguyên đã được xác thực (sử dụng các công cụ như wget hoặc curl) mà không phải lo lắng về xác thực.

Cookie sticky đặc biệt mạnh mẽ khi được sử dụng kết hợp với client-replay, có thể ghi lại quá trình xác thực một lần và chỉ cần phát lại khi khởi động mỗi khi bạn cần tương tác với các tài nguyên được bảo mật.

3.2.14 Streaming

Theo mặc định, mitmproxy sẽ đọc toàn bộ một yêu cầu / phản hồi, thực hiện bất kỳ thao tác nào được chỉ định trên đó và sau đó gửi thông báo cho bên kia. Điều này có thể có vấn đề khi tải xuống hoặc tải lên các tệp lớn. Khi tính năng phát trực tiếp được bật, nội dung thư không được lưu vào bộ đệm trên proxy mà thay vào đó được gửi trực tiếp đến máy chủ / máy khách. Các tiêu đề HTTP vẫn được lưu vào bộ đệm đầy đủ trước khi được gửi đi.

Tính năng phát trực tuyến yêu cầu / phản hồi được bật bằng cách chỉ định giới hạn kích thước trong tùy chọn stream_large_bodies.

Bạn cũng có thể sử dụng một tập lệnh để tùy chỉnh chính xác những yêu cầu hoặc phản hồi nào được truyền trực tuyến. Yêu cầu / phản hồi phải được gắn thẻ để phát trực tuyến bằng cách đặt thuộc tính .stream của chúng thành True:

```
"""
Select which responses should be streamed.

Enable response streaming for all HTTP flows.
This is equivalent to passing `--set stream_large_bodies=1` to mitmproxy.
"""


def responseheaders(flow):
    """
    Enables streaming for all responses.
    This is equivalent to passing `--set stream_large_bodies=1` to
mitmproxy.
    """
    flow.response.stream = True
```

3.3 Các tiện ích mở rộng

Cơ chế mở rộng của mitmproxy là một phần đặc biệt mạnh mẽ của mitmproxy. Trên thực tế, phần lớn chức năng của mitmproxy được xác định trong một bộ các phần bổ trợ tích hợp sẵn, triển khai mọi thứ từ chức năng như chống bộ nhớ đệm và sticky cookie cho đến ứng dụng web tích hợp

Các tiện ích mở rộng tương tác với mitmproxy bằng cách phản hồi các sự kiện, cho phép chúng xâm nhập và thay đổi hành vi của mitmproxy. Chúng được cấu hình thông qua các tùy chọn, có thể được đặt trong tệp cấu hình của mitmproxy, được thay đổi tương tác bởi người dùng hoặc được chuyển qua dòng lệnh. Cuối cùng, chúng có thể hiển thị các lệnh, cho phép người dùng gọi các hành động của họ trực tiếp hoặc bằng cách liên kết chúng với các phím trong các công cụ tương tác.

Một tiện ích mở rộng cụ thể:

```
"""
Basic skeleton of a mitmproxy addon.

Run as follows: mitmproxy -s anatomy.py
"""
from mitmproxy import ctx


class Counter:
    def __init__(self):
        self.num = 0

    def request(self, flow):
        self.num = self.num + 1
        ctx.log.info("We've seen %d flows" % self.num)


addons = [
    Counter()
]
```

Trên đây là một tiện ích đơn giản giúp theo dõi số lượng luồng (hoặc cụ thể hơn là các yêu cầu HTTP) mà mitmproxy đã thấy. Mỗi khi nó nhìn thấy một luồng mới, nó sử dụng cơ chế ghi nhật ký nội bộ của mitmproxy để thông báo việc kiểm đếm. Đầu ra có thể được tìm thấy trong nhật ký sự kiện trong các công cụ tương tác hoặc trên bảng điều khiển trong mitmdump.

Cách dùng addon:

mitmdump -s ./anatomy.py

Một số lưu ý:

- Mitmproxy chọn nội dung của danh sách các tiện ích và tải những gì nó tìm thấy vào cơ chế bổ trợ.
- Tiện ích chỉ là các đối tượng - trong trường hợp này, tiện ích là một ví dụ của Counter.

- Phương thức yêu cầu là một ví dụ về một sự kiện. Các tiện ích chỉ cần triển khai một phương thức cho mỗi sự kiện mà chúng muốn xử lý. Mỗi sự kiện và chữ ký của nó được ghi lại trong tài liệu API.
- Cuối cùng, mô-đun ctx là một mô-đun giữ cho hiển thị một tập hợp các đối tượng tiêu chuẩn thường được sử dụng trong các chương trình bổ trợ. Có thể chuyển một đối tượng ctx làm tham số đầu tiên cho mọi sự kiện, nhưng nó gọn gàng hơn khi chỉ hiển thị nó như một toàn cục có thể nhập được. Trong trường hợp này, sử dụng đối tượng ctx.log để ghi nhật ký của.

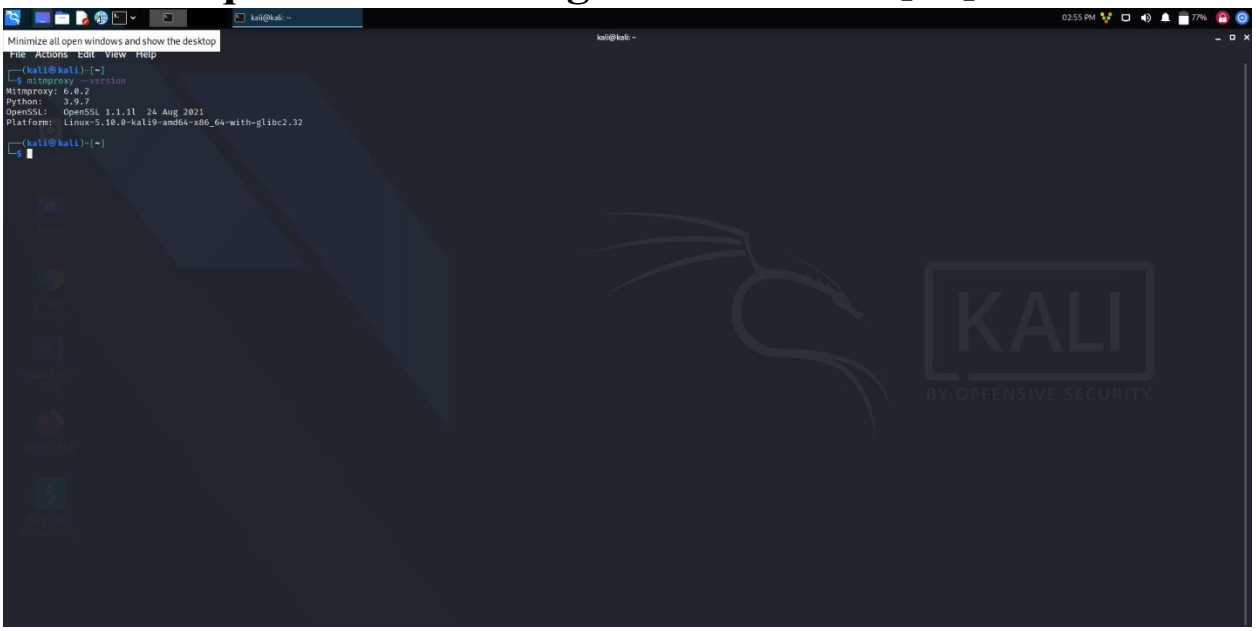
Cú pháp viết tắt :

Đôi khi, chúng tôi muốn viết một kịch bản nhanh chóng mà không gặp khó khăn khi tạo lớp. Cơ chế addon có một cách viết tắt cho phép một mô-đun nói chung được coi như một đối tượng addon. Điều này cho phép chúng tôi đặt các hàm xử lý sự kiện trong phạm vi mô-đun. Ví dụ: đây là một tập lệnh hoàn chỉnh thêm tiêu đề cho mọi yêu cầu:

```
"""An addon using the abbreviated scripting syntax."""


def request(flow):
    flow.request.headers["myheader"] = "value"
```

Chương 4: Demo

4.1 Sửa request HTTP đơn giản trên Linux [10]

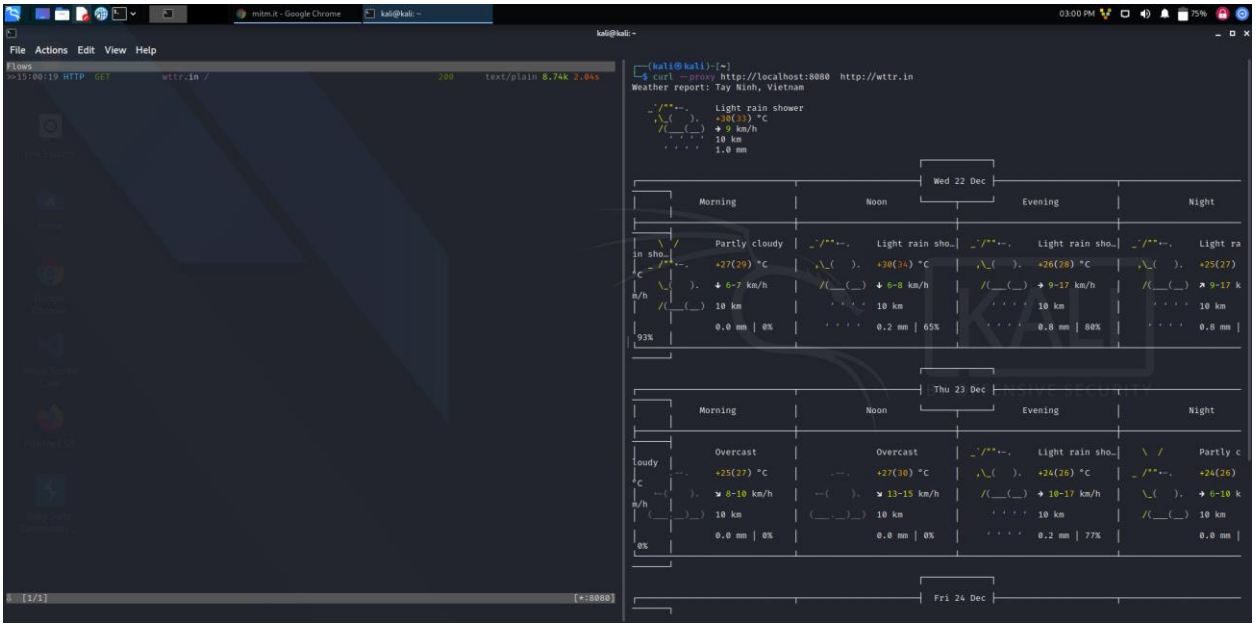


Hình 4.1 Phiên bản mitmproxy trên Kali Linux.

Trang web thử nghiệm http://wttr.in

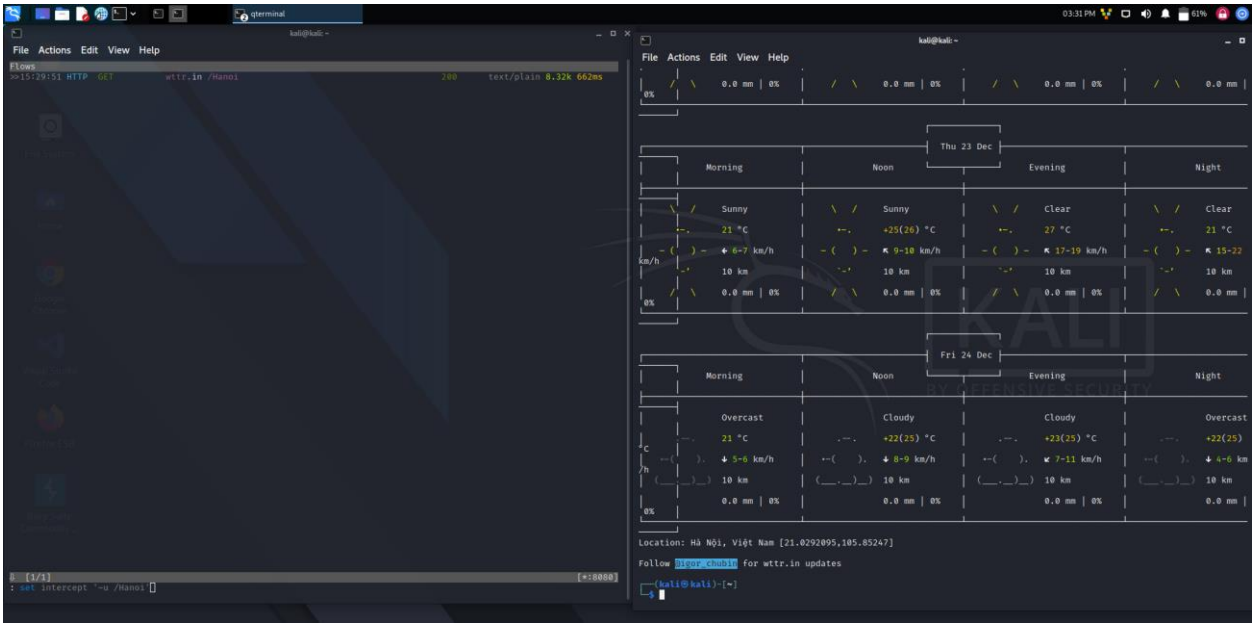
Sử dụng curl để kiểm tra lưu lượng qua mitmproxy

curl --proxy http://localhost:8080 http://wttr.in

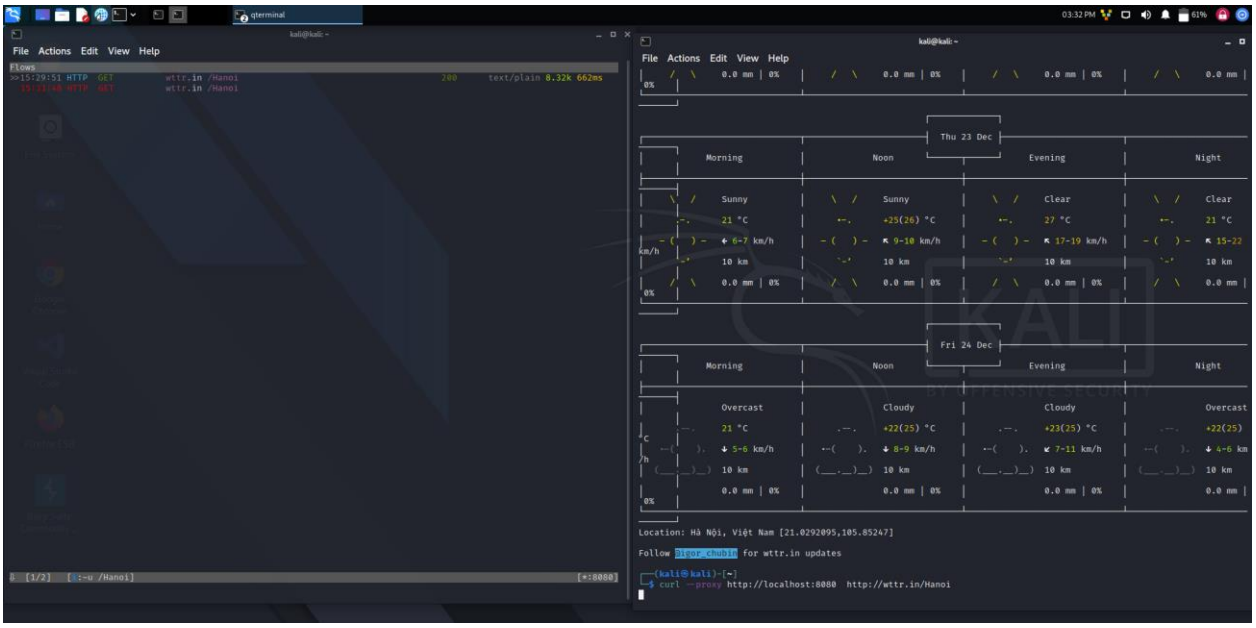


Hình 4.2Kiểm tra hoạt động mitmproxy bằng curl.

- Thiết lập mitmproxy chặn url
  Set intercept ‘ ~u/Hanoi‘

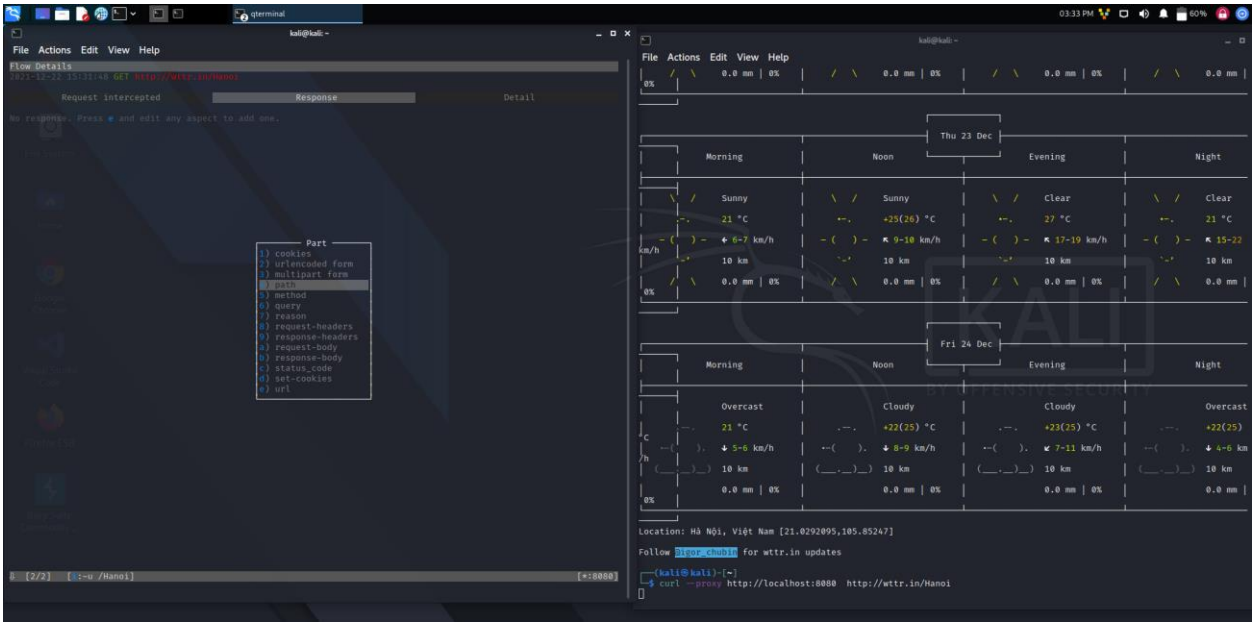


Hình 4.3 Thiết lập chặn yêu cầu chứa url /Hanoi.



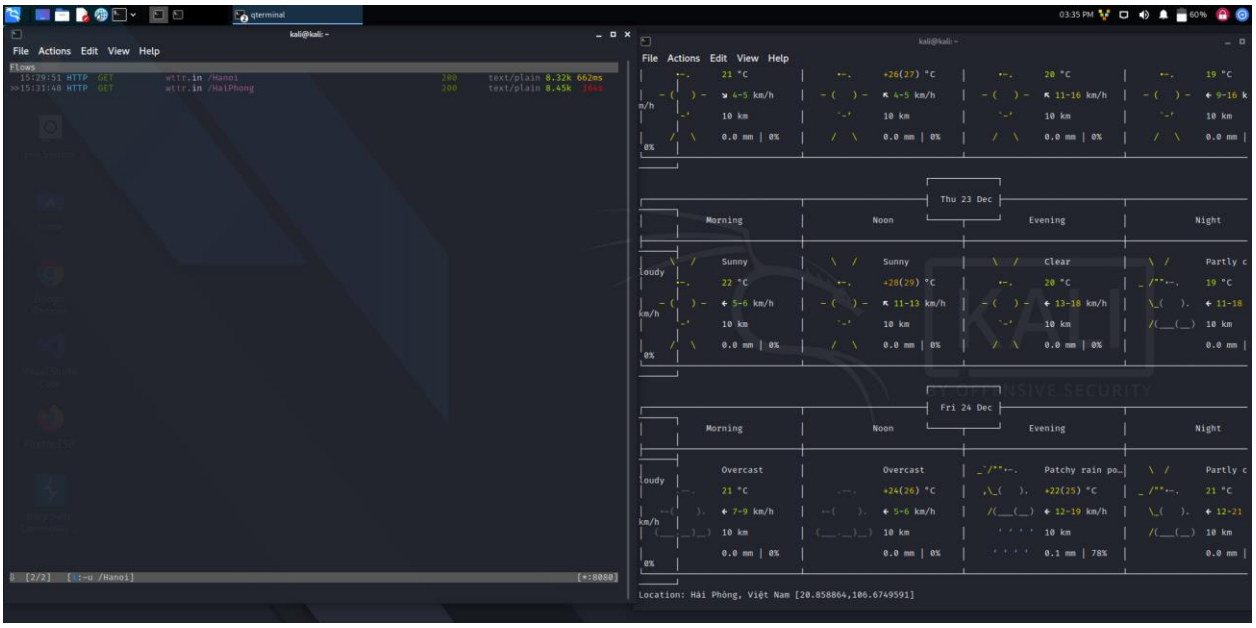
Hình 4.4 Chặn thành công yêu cầu.

- Sửa đổi request vừa chặn thành Haiphong



Hình 4.5 Sửa đổi request.

- Chuyển tiếp lưu lượng và thu được kết quả



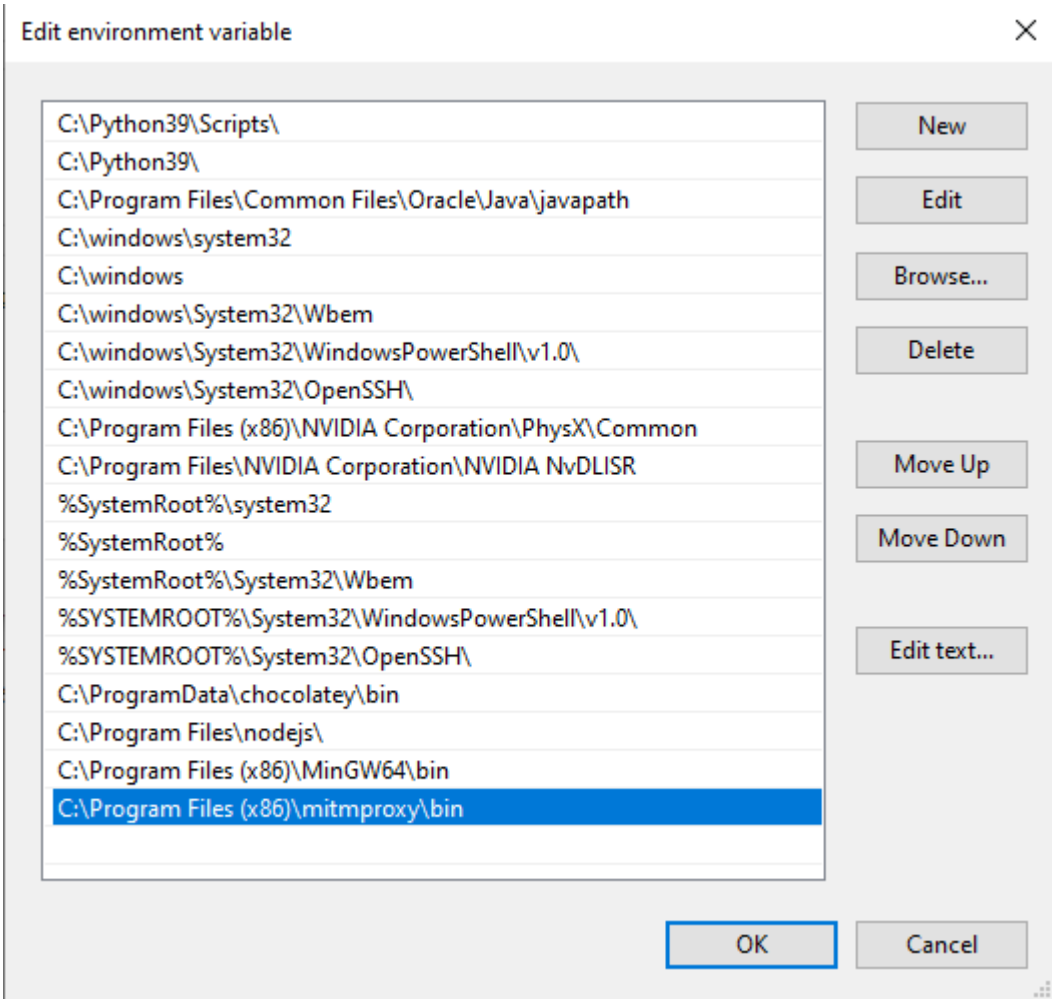
Hình 4.6 Chuyển tiếp yêu cầu thu được kết quả.

Hình 4.1.6 Chuyển tiếp yêu cầu thu được kết quả.

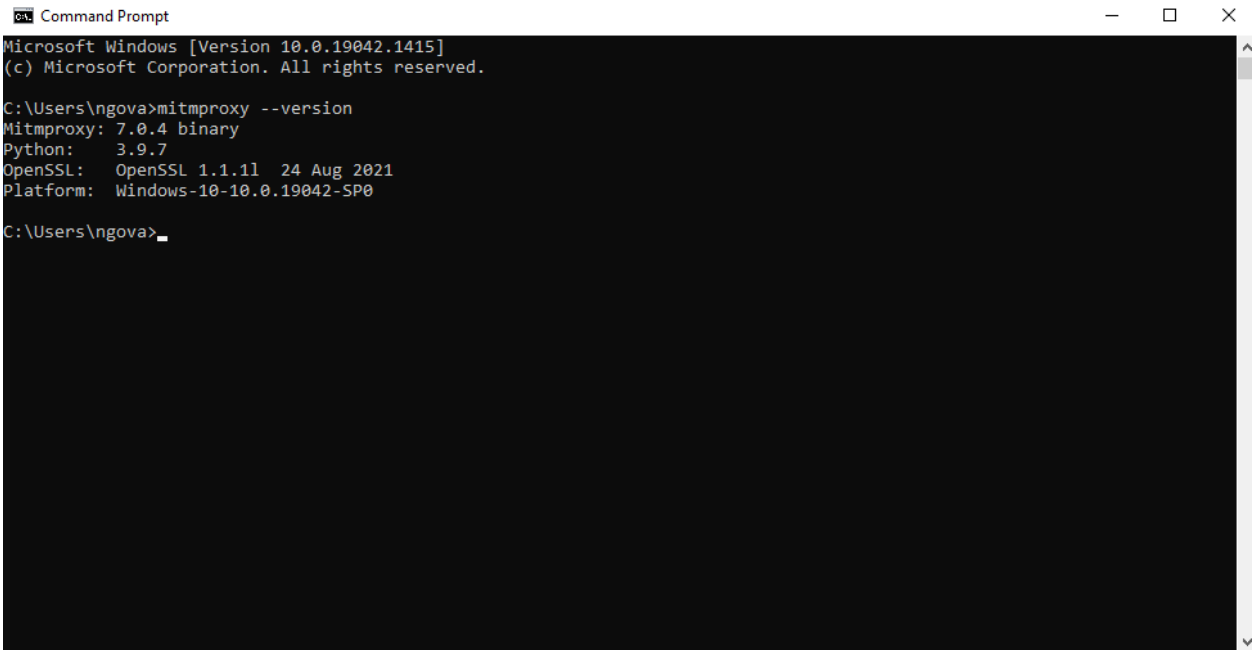
4.2 Cài đặt mitmproxy và sử dụng một addons đơn giản trên windows

Truy cập trang chủ mitmproxy.org chọn phiên bản tải về cho windows

Tiến hành cài đặt xong kiểm tra các biến môi trường được tạo :

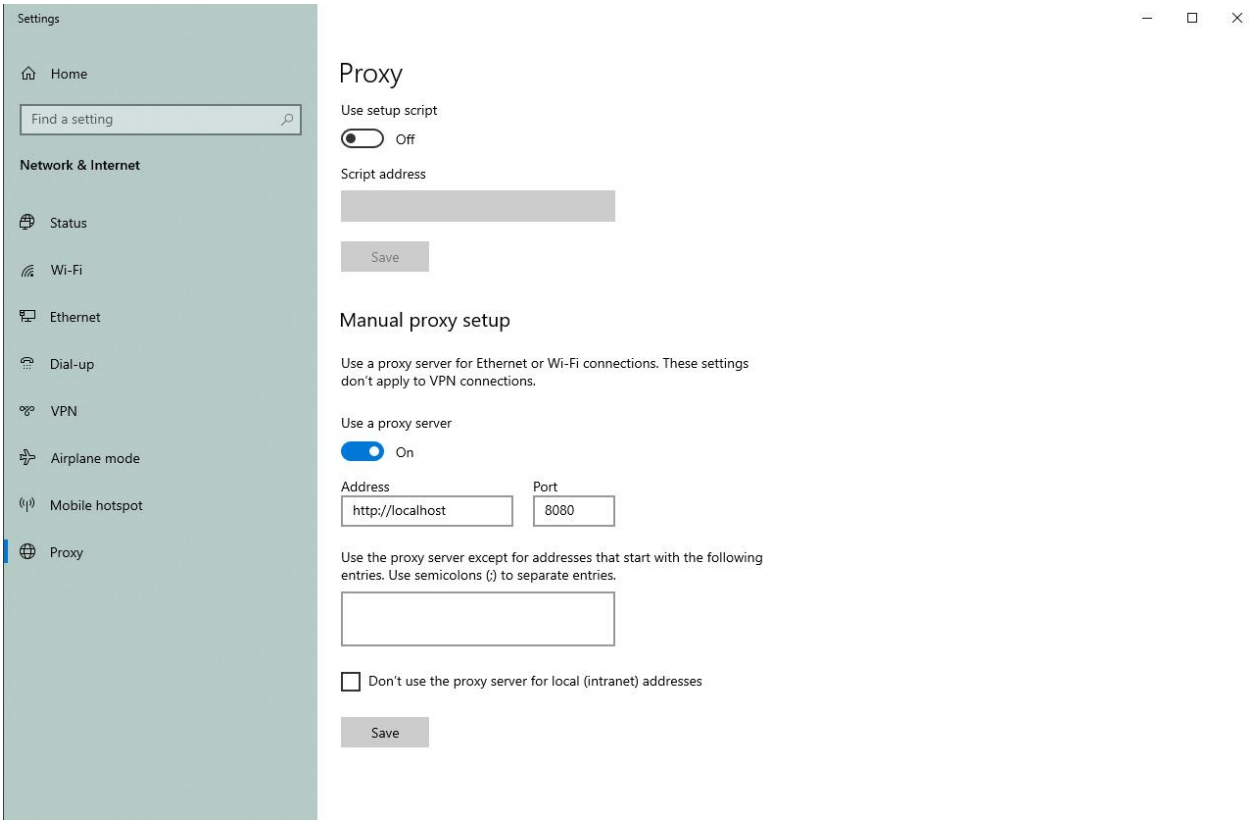


Hình 4.7 Kiểm tra các biến môi trường trên windows.



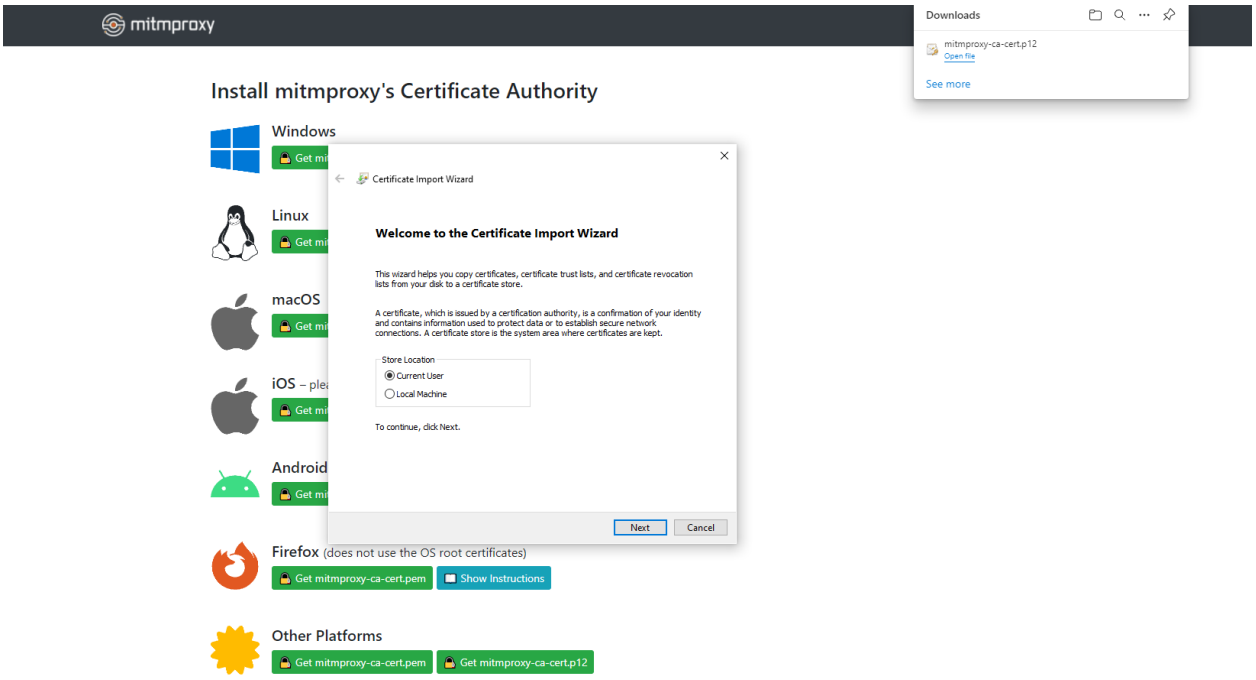
Hình 4.8 Kiểm tra phiên bản mitmproxy trên máy windows.

Thiết lập proxy trên chính thiết bị



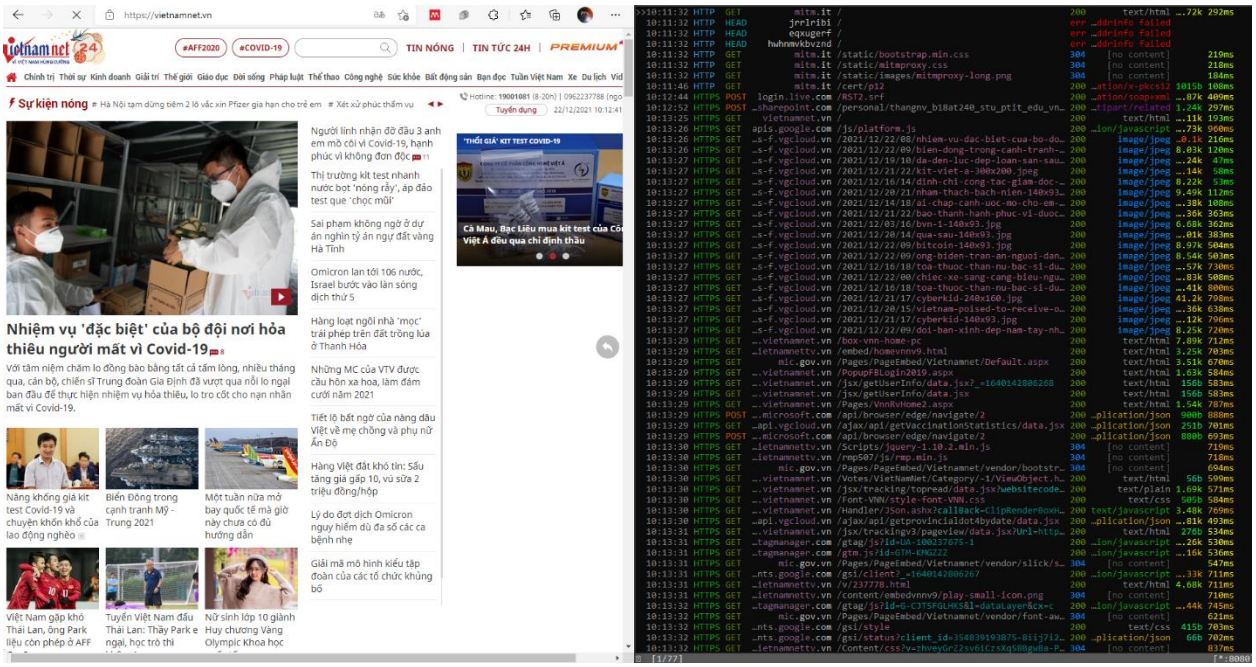
Hình 4.9 Thiết lập proxy trên windows.

Truy cập mitm.it để cài đặt chứng chỉ



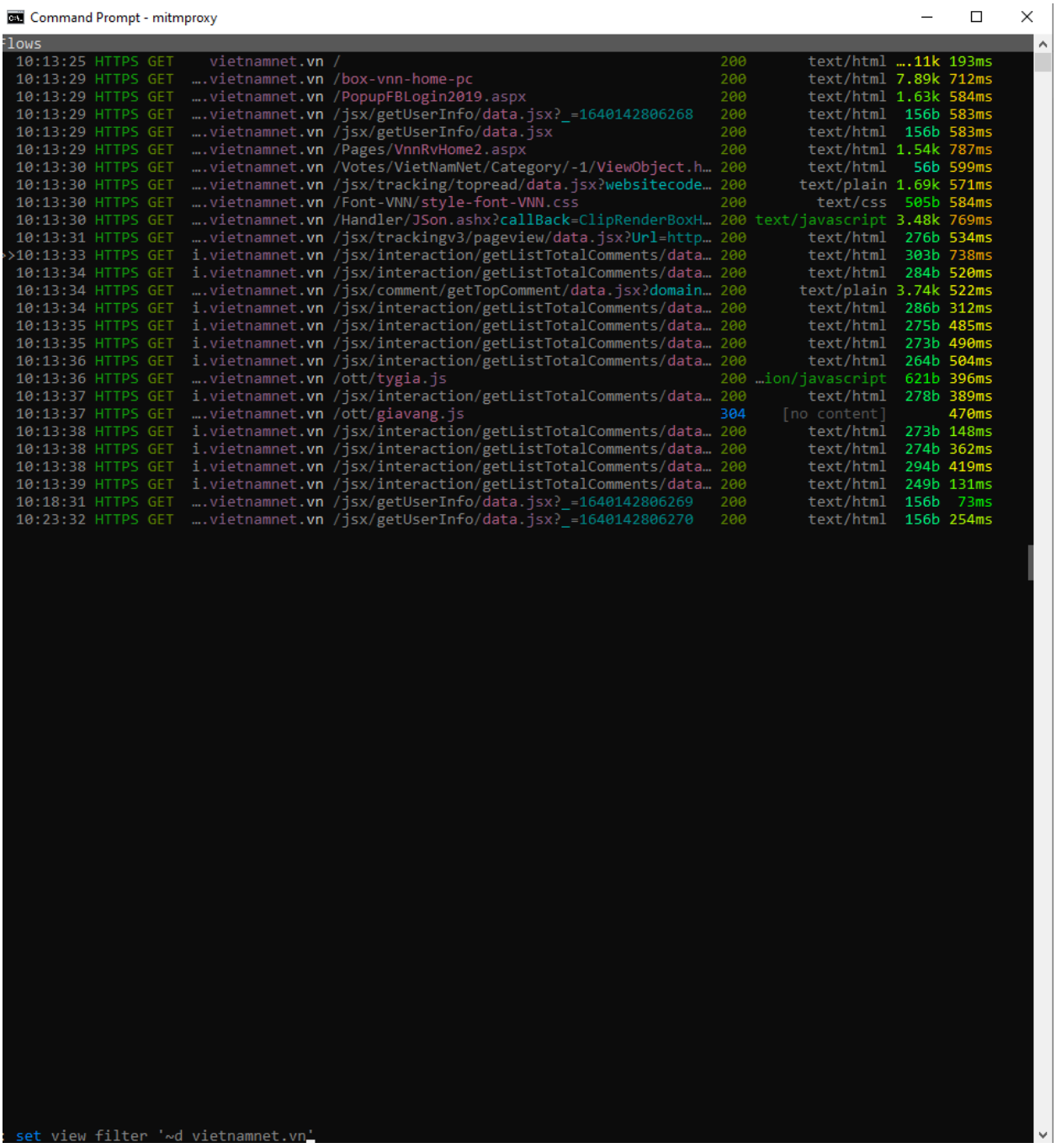
Hình 4.10 Cài đặt chứng chỉ trên windows.

Truy cập một trang vietnamnet.vn. Kiểm tra các lưu lượng trên mitmproxy



Hình 4.11 Kiểm tra hoạt động mitmproxy.

Sử dụng filter ~d vietnamnet.vn để lọc ra các lưu lượng từ tên miền vietnamnet.vn



Hình 4.12 Lọc hiển thị các lưu lượng HTTPS từ tên miền vietnamnet.vn.

Tạo một addon python script với tên redirect.py[11]. Mục đích của addons này là tự động chuyển hướng các trang web được chỉ định trong nó.

Nội dung của addons này như sau :

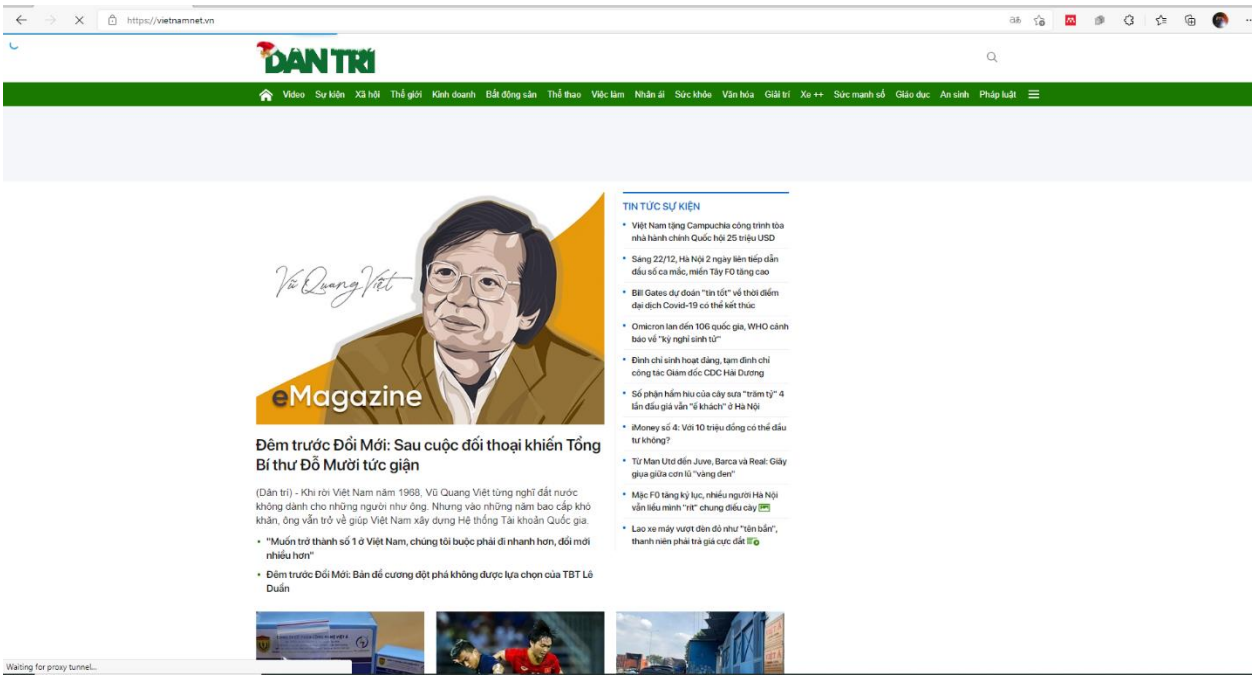
```python
from mitmproxy import http
from mitmproxy import ctx

class Redirect:
    redirect_rules = {
        "vietnamnet.vn": "dantri.com.vn", # replace `my-domain.com/images` with `my-domain.com/images-new` in requests url
        "bongda.com.vn": "bongda24h.vn"
    }

    def load(self, loader):
        ctx.options.http2 = False # HTTP2 won't let you update the url

    def request(self, flow: http.HTTPFlow) -> None:
        for init_domain, new_domain in self.redirect_rules.items():
            if (init_domain in flow.request.pretty_url):
                flow.request.url = flow.request.pretty_url.replace(init_domain, new_domain)

addons = [Redirect()]
```

Hình 4.13 Nội dung addon redirect.py.

Chạy addons bằng câu lệnh

mitmproxy -s redirect.py

Kiểm tra trên trình duyệt truy cập tên miền vietnamnet.vn tự động truy cập dantri.com.vn



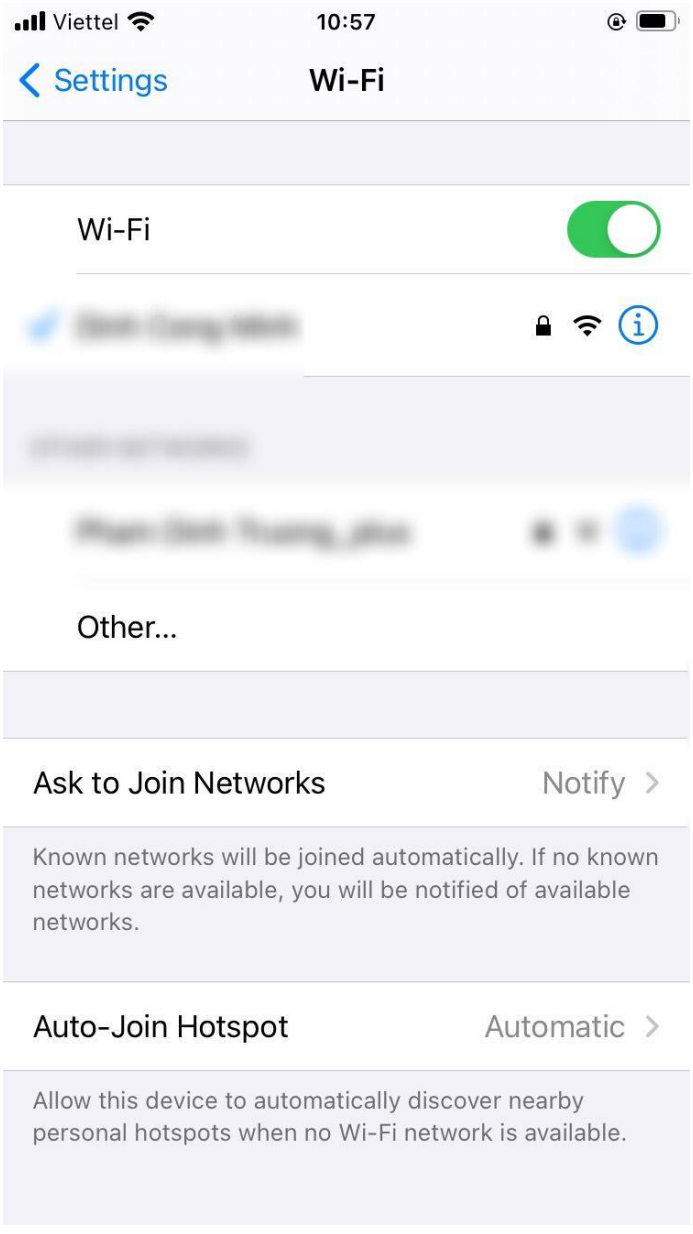
Hình 4.14 Kết quả khi chuyển hướng.

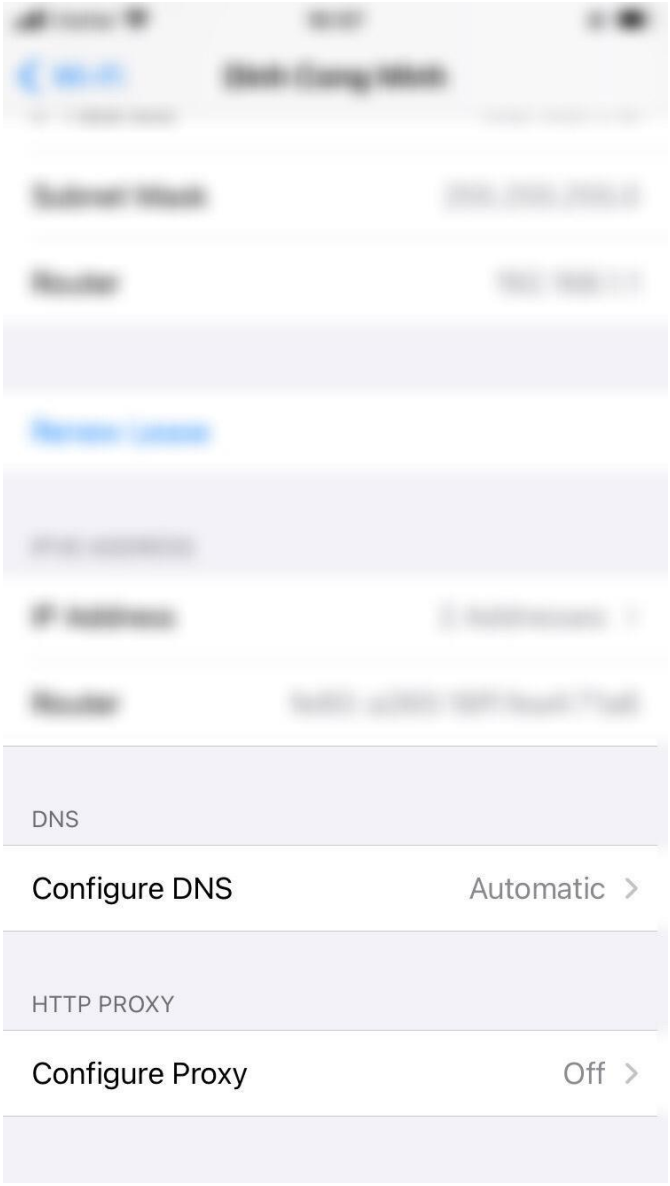
4.3.Theo dõi lưu lượng trên thiết bị chạy iOS bằng mitmproxy[12]

Địa chỉ thiết bị làm proxy server 192.168.1.12

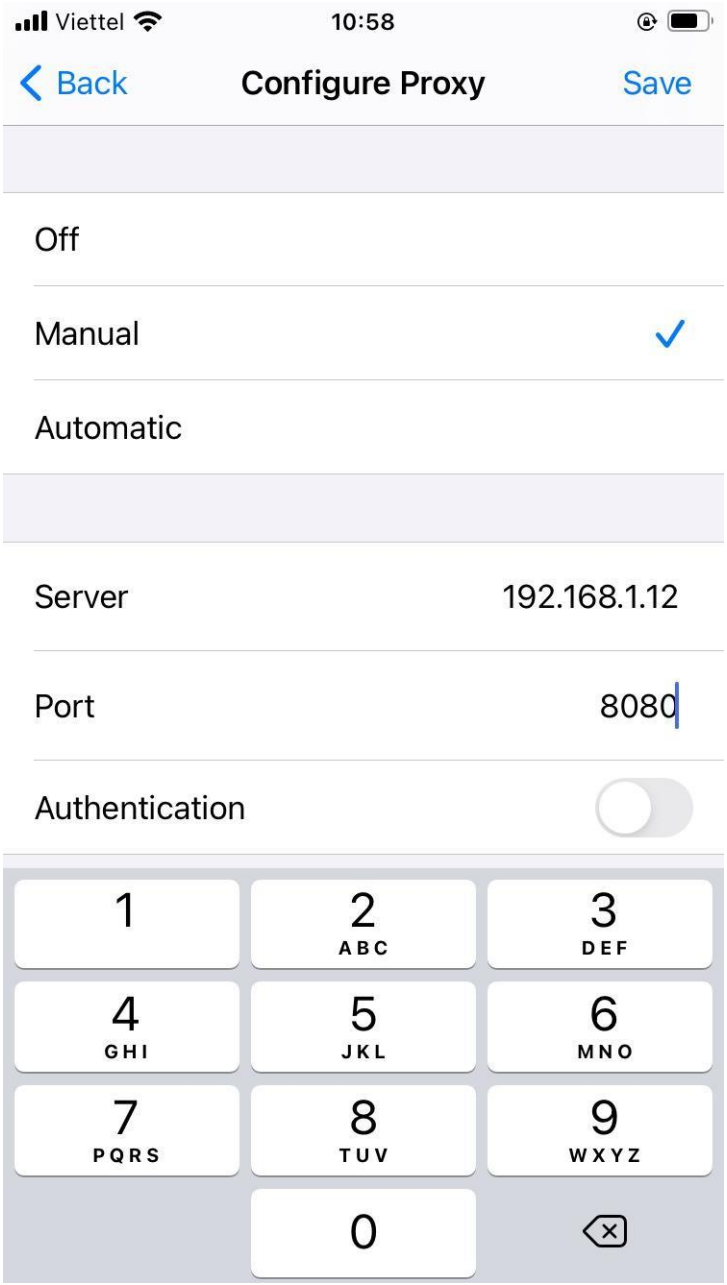
Cấu hình thiết bị chạy iOS 14.8.1

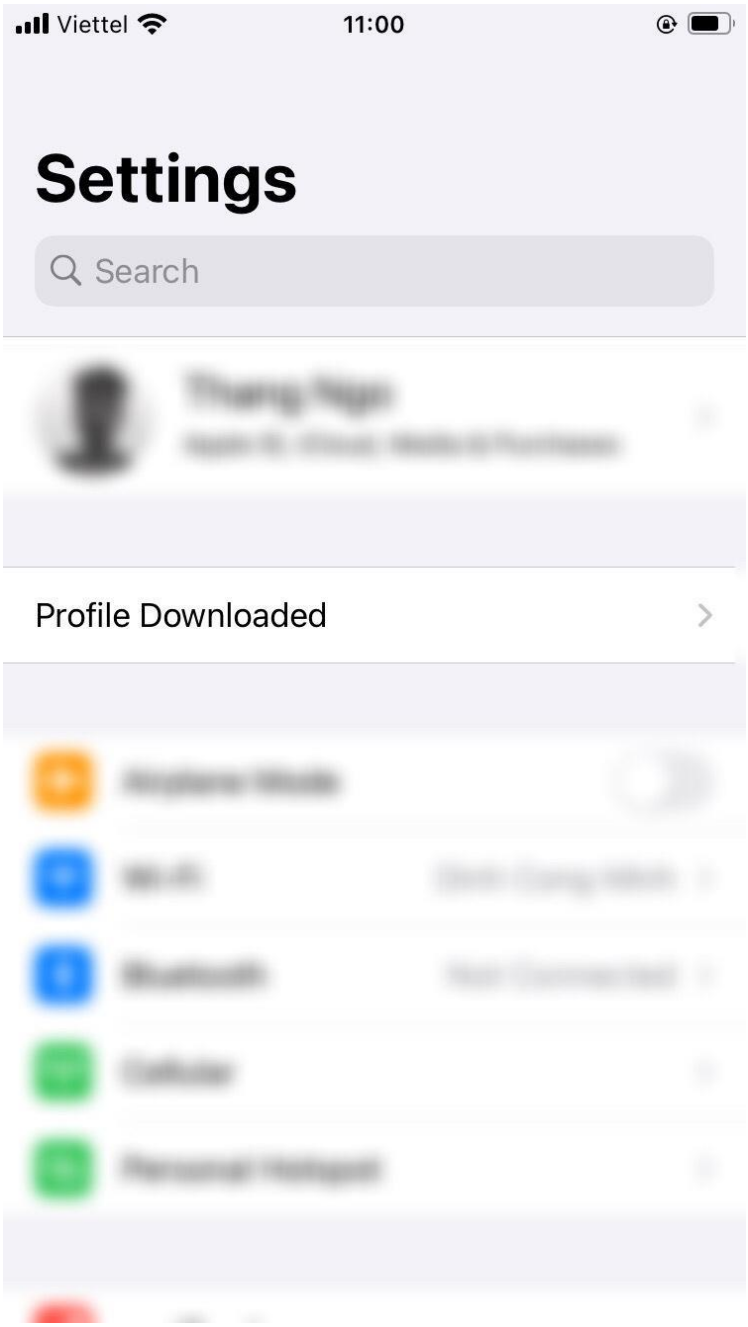
Kết nối với cùng mạng với thiết bị làm proxy server chạy mitmproxy



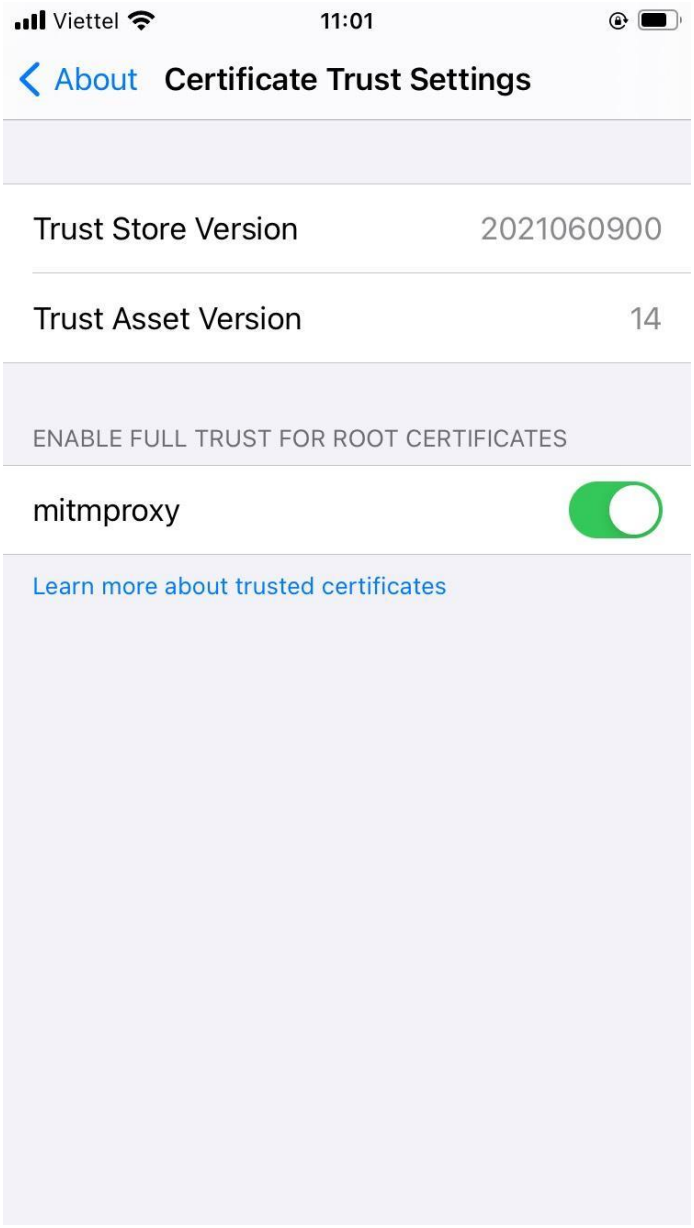


Hình 4.16 Cấu hình proxy.


Viettel
10:58
Back
Configure Proxy
Save
Off
Manual
Automatic
Server
192.168.1.12
Port
8080
Authentication
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
0

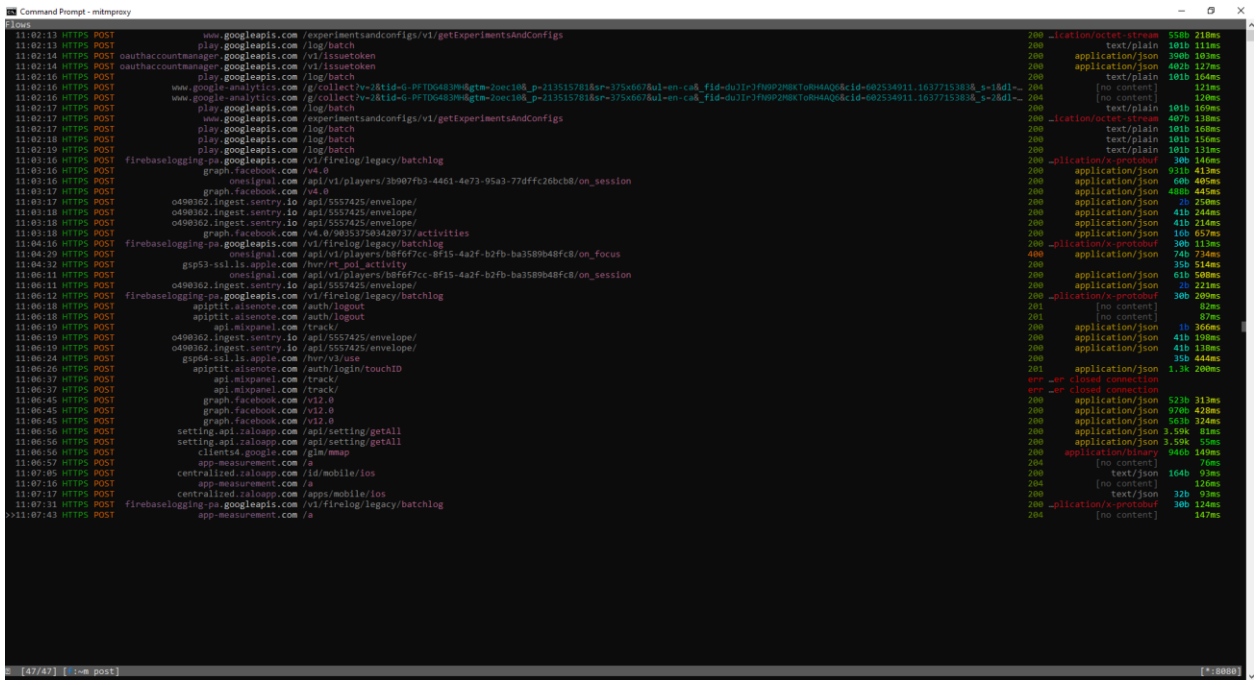


Hình 4.18 Tải chứng chỉ tại mitm.it và cài đặt.



Hình 4.19 Kích hoạt chứng chỉ..

Lọc các lưu lượng POST[13]



Hình 4.20 Kết quả sau khi lọc.

Kết luận

Bài báo cáo đã trình bày những tìm hiểu về công cụ mitmproxy. Đó là lịch sử hình thành, quá trình phát triển của mitmproxy; cách hoạt động, cách các nhà phát triển đã giải quyết các vấn đề khi kết nối với các giao thức an toàn hơn, các chế độ hoạt động có thể có ở mitmproxy, các vấn đề về chứng chỉ và các đặc trưng của công cụ này. Một tính năng quan trọng của công cụ này là các tiện ích mở rộng được viết bằng python script được thêm vào giúp người sử dụng có thể tự động và chỉnh sửa để mitmproxy hoạt động theo ý muốn.

Bài báo cáo cũng trình bày những thử nghiệm đơn giản, thực tế để chỉnh sửa giao tiếp mạng, sử dụng tiện ích mở rộng để chuyển hướng trang web mục tiêu và theo dõi lưu lượng mạng kết nối trên thiết bị di động.

Tuy còn một số hạn chế như phiên bản mitmweb chưa có đầy đủ các tính năng, giao diện chưa thân thiện với người dùng phổ thông, các phiên bản cũ chưa hỗ trợ hệ điều hành windows,… nhưng với điểm mạnh là miễn phí, mã nguồn mở, các tính năng khá đầy đủ thì đây vẫn là công cụ hữu ích cho các nhà phát triển kiểm tra sửa lỗi ứng dụng và nhiều mục đích khác.

Tài liệu tham khảo

[1] Aldo Cortesi, “Introducing mitmproxy: an interactive man-in-the-middle proxy,” Feb. 16, 2010. https://corte.si/posts/code/mitmproxy/announce0_1/ (accessed Dec. 18, 2021).

[2] “Releases · mitmproxy/mitmproxy.” https://github.com/mitmproxy/mitmproxy/releases?page=5 (accessed Dec. 18, 2021).

[3] Aldo Cortesi, “mitmproxy v1.0.0: Christmas Edition,” Dec. 26, 2016. https://corte.si/posts/code/mitmproxy/announce_1_0/ (accessed Dec. 18, 2021).

[4] Aldo Cortesi, “Mitmproxy 3,” Feb. 23, 2017. https://mitmproxy.org/posts/releases/mitmproxy3/ (accessed Dec. 18, 2021).

[5] Aldo Cortesi, “Mitmproxy 4,” May 15, 2018. https://mitmproxy.org/posts/releases/mitmproxy4/ (accessed Dec. 18, 2021).

[6] Maximilian Hils, “Mitmproxy 5,” Dec. 16, 2019. https://mitmproxy.org/posts/releases/mitmproxy5/ (accessed Dec. 22, 2021).

[7] Maximilian Hils, “Mitmproxy 6,” Dec. 13, 2020. https://mitmproxy.org/posts/releases/mitmproxy6/ (accessed Dec. 18, 2021).

[8] Maximilian Hils, “Mitmproxy 7,” Jul. 16, 2021. https://mitmproxy.org/posts/releases/mitmproxy7/ (accessed Dec. 18, 2021).

[9] “Documents,” *mitmproxy*, 2021. https://docs.mitmproxy.org/stable/ (accessed Dec. 22, 2021).

[10] NaveenKumar Namachivayam, “Learn mitmproxy #1 - Record, Replay, Intercept, and Modify HTTP Requests - YouTube,” Apr. 18, 2021. https://www.youtube.com/watch?v=igcsLKDfssw (accessed Dec. 22, 2021).

[11] Lucas Legname, “mitmproxy-helpers/Redirect at master · lucaslegname/mitmproxy-helpers,” Apr. 10, 2020. https://github.com/lucaslegname/mitmproxy-helpers/tree/master/Redirect (accessed Dec. 22, 2021).

[12] Petr Pátek, “Man-in-the-middle proxy to scrape data from mobile app API | Apify Blog,” Jul. 25, 2019. https://blog.apify.com/using-a-man-in-the-middle-proxy-to-scrape-data-from-a-mobile-app-api-e954915f979d/ (accessed Dec. 22, 2021).

[13] Sufiyan Yasa, “mitmproxy filters that will make you a better developer - Sufiyan Yasa,” Feb. 28, 2021. https://sufiyanyasa.com/blog/mitmproxy-better-filters/ (accessed Dec. 17, 2021).