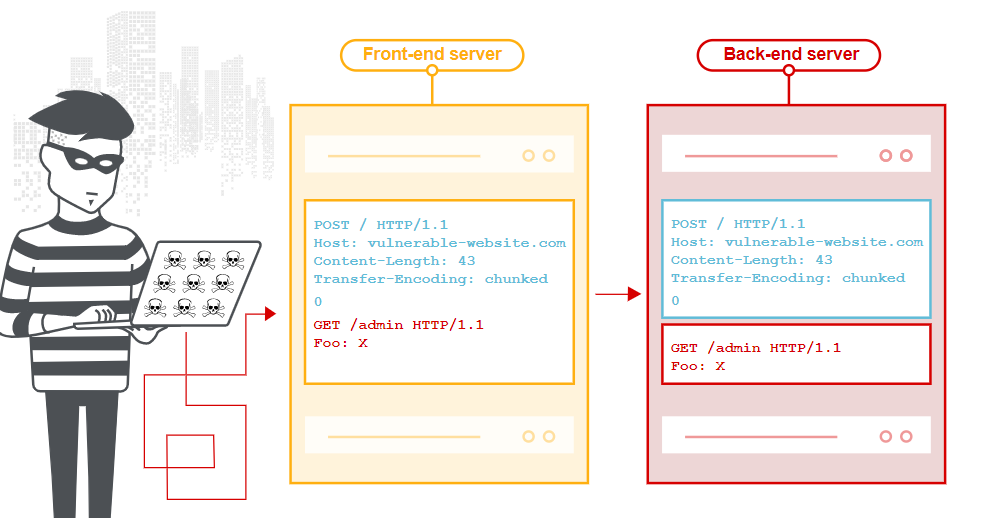
HTTP Request Smuggling

**What is HTTP Request Smuggling ?**

HTTP request smuggling (HRS) là 1 kỹ thuật tấn công nhằm vào các HTTP server(web server, proxy server). Bất cứ khi nào 1 HTTP requset của client được phân tích bởi nhiều hơn 1 hệ thống thì đều có khả năng bị HRS. HRS thường rất nghiêm trọng, cho phép kẻ tấn công vượt qua các kiểm soát bảo mật, truy cập trái phép vào dữ liệu nhảy cảm và xâm phạm vào ứng dụng. img



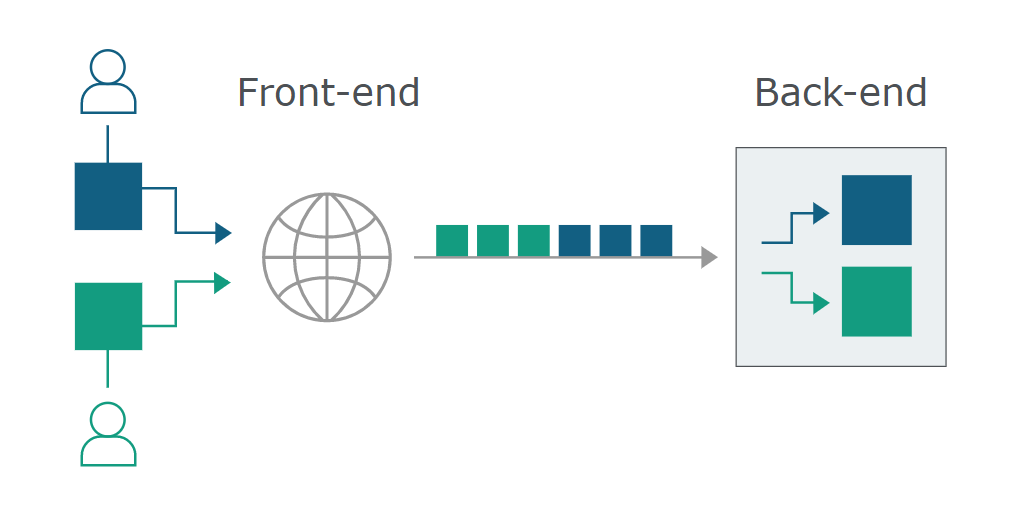
HRS có các hướng khai thác sau:

* khai thác một web cache server được triển khai giữa client và web server
* Bypass hệ thống firewall của web server
* Khai thác một web proxy server được triển khai giữa client và web server.

Để tiến hành HRS, không nhất thiết ứng dụng web phải có lỗ hổng (chẳng hạn SQL Injection, XSS,…), thay vào đó, chỉ cần một sự khác nhau nhỏ trong cách xử lý các HTTP request không hợp lệ của các HTTP server. Hacker sẽ gửi đi những HTTP request không hợp lệ để xem phản ứng của hai hệ thống, từ đó, tìm cách bypass filter của các hệ thống này.

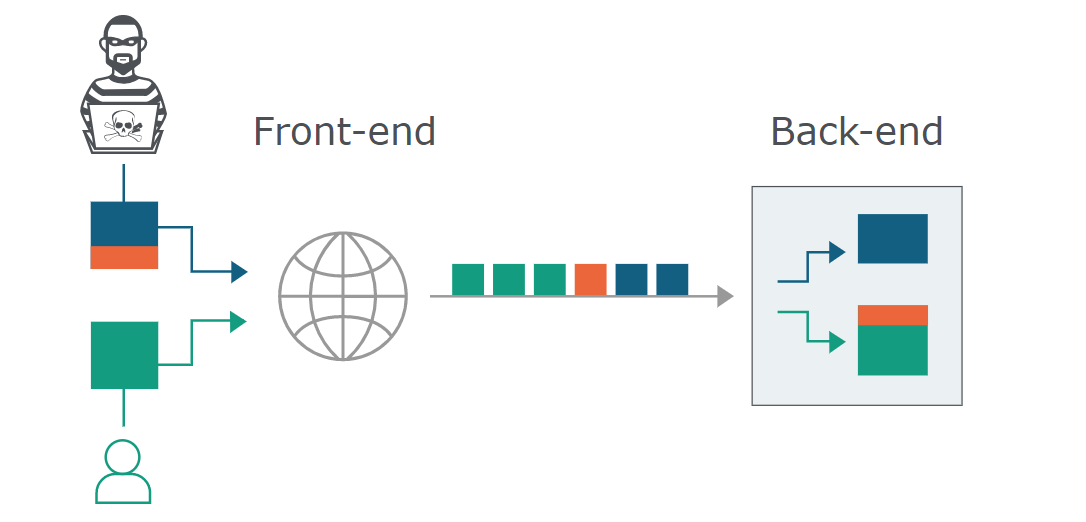
**HTTP request smuggling attack**

Khi máy chủ front-end chuyển tiếp các yêu cầu HTTP đến một máy chủ back-end, nó thường gửi một số request qua cùng một kết nối mạng back-end, vì điều này hiệu quả và hiệu quả hơn nhiều. Giao thức rất đơn giản: Các HTTP request được gửi lần lượt và máy chủ nhận phân tích cú pháp các HTTP request headers để xác định nơi một yêu cầu kết thúc và yêu cầu tiếp theo bắt đầu:



Trong tình hướng này, điều quan trọng là hệ thống front-end và back-end phải đồng ý về ranh giới giữa các yêu cầu. Nếu không, kẻ tấn công có thể gửi một yêu cầu không rõ ràng được hệ thống front-end và back-end diễn giải khác nhau:

hacker khiến 1 phần của front-end request của chúng được máy chủ back-end hiểu là phần bắt đầu của phần tiếp theo. Nó được thêm vào trước 1 cách hiệu quả cho request tiếp theo và do đó có thể ảnh hướng đến cách ứng dụng xử lý request đó



**Prevent HTTP request smuggling**

HTTP request smuggling phát sinh trong các tình huống trong đó server front-end chuyển tiếp nhiều yêu cầu đến server back-end qua cùng một kết nối mạng và giao thức được sử dụng cho các kết nối back-end có nguy cơ hai máy chủ không thống nhất về ranh giới giữa các request. Một số cách chung để ngăn chặn phát sinh các lỗ hổng bắt lậu yêu cầu HTTP như sau:

* Không sử dụng lại các kết nối back-end để mỗi yêu cầu back-end được gửi qua một kết nối mạng riêng biệt.
* Sử dụng HTTP / 2 cho các kết nối back-end, vì giao thức này ngăn chặn sự mơ hồ về ranh giới giữa các request.
* Thống nhất phần mềm giữa server back-end và front-end, thống nhất tiêu chuẩn giữa các yêu cầu.

**Cách thức tấn công HTTP request smuggling**

**Có thể chia ra làm 5 kiểu : CL, CL.CL, CL.TE, TE.CL và TE.TE**

CL và TE lần lượt viết tắt của Content-Length và Transfer-Encoding là 2 HTTP header. Content-Length cho biết kích thước của body của request, tính bằng byte, được gửi cho người nhận còn Transfer-Encoding chỉ định hình thức mã hóa (gzip, chunked.compress….) được sử dụng để đóng gói vận chuyển phần body một cách nhanh chóng, giải quyết được vấn đề của Content-Length với dữ liệu lớn. Nhưng tính năng quan trọng cần nhớ ở đây là 2 header này còn là cách xác định kết thúc một HTTP message. Đây là mấu chốt để thực hiện HRS !

**GET Request với CL**

Giả sử Front-End Server (Proxy Server) cho phép GET request mang Body và Back-end Server không cho phép GET request mang Body.Như vậy Back-end Server sẽ bỏ qua luônContent-Length header trong request GET và sẽ không xử lý nó.Điều này có thể dẫn đến request Smuggling.