**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

****

**Ngô Tùng Dương**

**PHÂN TÍCH AN NINH CÁC MÔDUN MỞ RỘNG CHO TRÌNH DUYỆT**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**Hà Nội – 2018**

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Ngô Tùng Dương**

**PHÂN TÍCH AN NINH CÁC MÔDUN MỞ RỘNG CHO TRÌNH DUYỆT**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Công nghệ thông tin**

**Cán bộ hướng dẫn: TS. Lê Đình Thanh**

**Cán bộ đồng hướng dẫn: TS. Lê Thị Hợi**

**Hà Nội - 2018**

**LỜI CẢM ƠN**

Sau quá trình học tập và rèn luyện tại trường Đại học Công nghệ, Đại học Quốc gia Hà Nội và một thời gian thực hiện việc nghiên cứu về đề tài “Phân tích an ninh các môdun mở rộng cho trình duyệt”, em xin bày tỏ lòng cảm ơn chân thành về sự giúp đỡ của các thầy, cô giảng viên, cán bộ các phòng, ban trong trường đối với em trong thời gian vừa qua.

Đặc biệt, em xin chân thành cảm ơn thầy hướng dẫn – Tiến sĩ Lê Đình Thanh và cô đồng hướng dẫn – Tiến sĩ Lê Thị Hợi, hai thầy cô đã hết sức nhiệt tình giúp đỡ em không chỉ trong việc thực hiện khóa luận mà còn trong suốt thời gian em tham gia học tập tại trường. Em xin gửi lời cảm ơn chân thành đến Ban giám hiệu Trường Đại học Công nghệ đã tạo điều kiện giúp đỡ em có thể thực hiện đề tài. Cảm ơn toàn thể các thầy cô công tác tại Trường Đại học Công nghệ - Đại học Quốc gia Hà nội đã dạy dỗ và truyền đạt những kiến thức quí báu cho em trong suốt thời gian học tập và rèn luyện. Em xin được gửi lời biết ơn tới cha mẹ đã nuôi nấng và dìu dắt để em có thể đi đến ngày hôm nay. Cuối cùng, xin cảm ơn tập thể lớp K59CD và đặc biệt cảm ơn những người bạn tốt đã ở bên, cho em những kỷ niệm không thể phai mờ trong thời gian học tập dưới mái trường này.

**TÓM TẮT**

**Tóm tắt**: Hiện nay, sự phát triển mạnh mẽ của các tiện ích mở rộng cho trình duyệt đã đặt ra một vấn đề lớn về an toàn đối với người dùng. Các tiện ích mở rộng đang trở thành công cụ cho những kẻ tấn công thực hiện các hành vi độc hại như ăn cắp thông tin người dùng, phát tán mã độc, lừa đảo, quảng cáo,… Các nhà phát triển trình duyệt web và các nhóm nghiên cứu đang không ngừng tìm ra cách phát hiện và xóa bỏ các tiện ích mở rộng có hành vi độc hại. Tuy nhiên, các phần mở rộng này vẫn tồn tại và thực hiện các hành vi độc hại dưới phương thức ngày một tinh vi hơn. Mục tiêu của khóa luận là tìm ra phương pháp phát hiện các hành vi mạng độc hại của các môdun mở rộng để bảo vệ người dùng trước mối đe dọa này. Nội dung đồ án sẽ nói qua về các hành vi độc hại của tiện ích mở rộng, sau đó sẽ tập trung vào phương pháp phát hiện hành mạng độc hại. Bao gồm hai phần chính là : phương pháp thu thập các hành vi mạng của tiện ích mở rộng và phương pháp phát hiện hành vi độc hại của tiện ích mở rộng. Từ đó xây dựng hệ thống phát hiện hành vi độc hại cài đặt trên máy người dùng để giúp bảo vệ họ trước các nguy cơ về an ninh do các môdun mở rộng cho trình duyệt tạo ra.

***Từ khóa****: hành vi độc hại, tiện ích mở rộng.*

**LỜI CAM ĐOAN**

Tôi xin cam đoan những nghiên cứu của tôi về phân tích an ninh các môđun mở rộng cho trình duyệt mà tôi viết trong khoá luận này là sự thật. Những gì tôi viết ra không sao chép từ các tài liệu, không sử dụng các kết quả của người khác mà không trích dẫn cụ thể.

Tôi xin cam đoan hệ thống tôi trình bày trong khoá luận là do tôi tự nghiên cứu và cài đặt , không sao chép mã nguồn của người khác, trong quá trình thực hiện khoá luận có sự hướng dẫn thực hiện của thầy Lê Đình Thanh và cô Lê Thị Hợi. Nếu sai tôi hoàn toàn chịu trách nhiệm theo quy định của Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội

Hà Nội, ngày tháng năm

Sinh viên

**BẢNG THUẬT NGỮ VÀ VIẾT TẮT**

**Từ viết tắt**

API Application Programming Interface

JS JavaScript

JSON JavaScript Object Notation

HTTP Hypertext Transfer Protocol

HTTP Hypertext Transfer Protocol Secure

URL Uniform Resource Locator

**Thuật ngữ**

Extension Tiện ích mở rộng cho trình duyệt

Malware Mã độc, trương trình độc hại

GET, POST Phương thức trong giao thức HTTP

Request  Thông tin gửi từ client lên server

Danh mục hình ảnh

[Hình 1.2: Extension Google translate với hơn 10 triệu người dùng 10](#_Toc512497235)

[Hình 1.3 : Extension Momentum với hơn 3 triệu người dùng 10](#_Toc512497236)

[Hình 2.1: Kiến trúc của một extension 16](#_Toc512497237)

[Hình 2.2: Phương thức hoạt động của Droidclub 18](#_Toc512497238)

[Hình 2.3: *Nội dung lệnh được gửi đến từ máy chủ điều khiển* 19](#_Toc512497239)

[Hình 2.4: Droidclub chèn quảng cáo và thay đổi nội dung trang web 19](#_Toc512497240)

[Hình 2.5: Demo mã độc ghi lại hành động của người dùng 20](#_Toc512497241)

[Hình 2.6: Mã độc đào tiền ảo của Droidclub 20](#_Toc512497242)

[Hình 2.7: Trang extension giả mà Droidclub tạo ra để đánh lừa ngươi dùng 21](#_Toc512497243)

[Hình 2.8 Hàm nhận và thực thi mã từ xa chứa trong extension 22](#_Toc512497244)

[Hình 2.9: Sơ đồ hoạt động của Extension Change HTTP Request Header 22](#_Toc512497245)

[Hình 2.10: Các Extension độc hại giả dạng Extension chặn quảng cáo 23](#_Toc512497246)

[Hình 2.11: Các Extension độc hại giả dạng Extension IDM 24](#_Toc512497247)

[Hình 2.12: Các thông tin mà Extension độc hại thu được 24](#_Toc512497248)

[Hình 2.13: Một chương trình crack chứa mã cài đặt Extension độc hại 25](#_Toc512497249)

[Hình 2.14: Trang web yêu cầu cài đặt Extension 26](#_Toc512497250)

[Hình 3.1: Kiến trúc hệ thống thu thập hành vi mạng của môdun mở rộng cho trình duyệt 29](#_Toc512497251)

[Hình 3.2 : Kiến trúc của bộ thu thập hành vi mạng tại trình duyệt 33](#_Toc512497252)

[Hình 3.3: Sơ đồ hoạt động của proxy với phương thức HTTP 35](#_Toc512497253)

[Hình 3.4: Sơ đồ hoạt động của proxy với phương thức HTTP 36](#_Toc512497254)

[Hình 3.5: Kiến trúc của bộ so sánh và trích xuất 37](#_Toc512497255)

[Hình 3.6: Bảng tỉ lệ hành vi mạng của các Extension 39](#_Toc512497256)

[Hình 4.1 Số trang web không an toàn được phát hiện mỗi tuần 41](#_Toc512497257)

[Hình 4.2 Số trang web bị coi là nguy hiểm theo dịch vụ duyệt web an toàn 41](#_Toc512497258)

[Hình 4.3: Mô tả nội dung chứa trong một URL 42](#_Toc512497259)

[Hình 4.4: Nội dung trong phần body của một request 42](#_Toc512497260)

[Hình 4.5 Nội dung chứa trong một URL có method GET 43](#_Toc512497261)

[Hình 4.6 Kiến trúc thuật toán so sánh format request 44](#_Toc512497262)

Mục lục

[Chương 1. Giới thiệu 9](#_Toc512500521)

[1.1. Giới thiệu môdun mở rộng cho trình duyệt và lợi ích của chúng 9](#_Toc512500522)

[1.2. Các rủi ro an ninh liên quan đến môđun mở rộng cho trình duyệt 11](#_Toc512500523)

[1.3. Mục tiêu của đề tài 12](#_Toc512500524)

[1.4. Sản phẩm của khóa luận 12](#_Toc512500525)

[1.5. Nội dung của báo cáo khóa luận 13](#_Toc512500526)

[1.6. Tiềm năng của đề tài 13](#_Toc512500527)

[Chương 2: Browser Extension – Môdun mở rộng cho trình duyệt. 16](#_Toc512500528)

[2.1 Kiến trúc của một browser extension. 16](#_Toc512500529)

[2.2 Một số extension độc hại trong thực tế và phương thức hoạt động của chúng 17](#_Toc512500530)

[2.3 Một số cách thức phát tán extension độc hại 25](#_Toc512500531)

[Chương 3. Thu thập nhật ký hành vi mạng của môđun mở rộng 27](#_Toc512500532)

[3.1. Phân loại hành vi của mô đun mở rộng cho trình duyệt 27](#_Toc512500533)

[3.2. Mục tiêu của thu thập nhật ký hành vi mạng 28](#_Toc512500534)

[3.3. Kiến trúc hệ thống thu thập nhật ký hành vi mạng của môdun mở rộng 29](#_Toc512500535)

[3.3.1 Bộ thu thập nhật ký mạng tại trình duyệt 30](#_Toc512500536)

[3.3.2 Bộ thu thập nhật ký mạng tại proxy 34](#_Toc512500537)

[3.3.3  Bộ so sánh và trích xuất 37](#_Toc512500538)

[3.4 Thử nghiệm 38](#_Toc512500539)

[Chương 4. Phát hiện hành vi mạng xấu độc của môđun mở rộng 40](#_Toc512500540)

[4.1. Phương pháp whitelist và blacklist 40](#_Toc512500541)

[4.2. Phương pháp dựa vào đánh giá URL 40](#_Toc512500542)

[4.3. Phương pháp dựa vào phân tích nội dung của request 42](#_Toc512500543)

[4.4 Phương pháp phân tích thông tin dựa vào người dùng 44](#_Toc512500544)

[4.5. Thử nghiệm 45](#_Toc512500545)

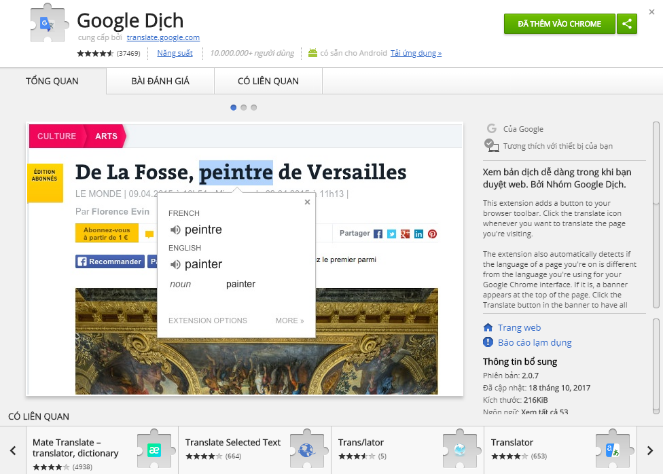
[CHƯƠNG 5: KẾT LUẬN 46](#_Toc512500546)

# Chương 1. Giới thiệu

## 1.1. Giới thiệu môdun mở rộng cho trình duyệt và lợi ích của chúng

 Ngày nay, nền tảng web đang ngày càng phát triển và đang dần trở thành môi trường cho rất nhiều ứng dụng, từ các công việc thường ngày như đọc báo, xem phim, nghe nhạc,chơi game, …cho đến các tác vụ quan trọng như online banking, email hay lưu trữ thông tin. Thậm chí, có rất nhiều ứng dụng trên máy tính cũng đang dần chuyển sang hoạt động trên nền tảng web, ví dụ như Office, ứng dụng vẽ và chỉnh sửa ảnh, ứng dụng quản lý,… Để cung cấp một môi trường mạnh mẽ cũng như nâng cao trải nghiệm cho người dùng, bên cảnh thiết kế một trình duyệt web an toàn với nhiều tính năng hữu ích, các nhà phát triền trình duyệt web còn cho phép bên thứ ba cung cấp các phần mở rộng cho trình duyệt - browser extension. Các phần mở rộng có rất nhiều tính năng hữu ích và là một phần không thể thiếu của trình duyệt web. Chúng có thể giúp người dùng quản lý mật khẩu, bookmark, kiểm tra mail, dịch văn bản,.... Chúng có thể can thiệp vào DOM để chặn quảng cáo, sửa đổi nội dung trang web, can thiệp vào các luồng request để thay đổi nội dung gửi đi, chặn trang web, thậm chí chúng có thể tùy biến giao diện hiển thị của các tab hay cả browser. Sau đây là một số extension phổ biến với người dùng:

Hình 1.1: Extension Adblock với hơn 40 triệu người dùng



#### Hình 1.2: Extension Google translate với hơn 10 triệu người dùng

#### Hình 1.3 : Extension Momentum với hơn 3 triệu người dùng

* Checker Plus for Gmail với hơn 1 triệu người dùng và 11 nghìn lượt đánh giá
* Trình chặn quảng cáo cho Youtube với gần 1 triệu người dùng hơn 30 nghìn lượt đánh giá
* Evernote Web Clipper với gần 5 triệu người dùng và khoảng 133 nghìn lượt đánh giá

## 1.2. Các rủi ro an ninh liên quan đến môđun mở rộng cho trình duyệt

Như đã nói ở phần mở đầu của bài viết, người dùng đang có xu hướng sử dụng trình duyệt web để thực hiện nhiều tác vụ quan trọng. Cho nên, trình duyệt web có thể là đầu mối cung cấp rất nhiều thông tin nhậy cảm của người dùng, ví dụ như: lịch sử duyệt web, cookie, nội dung tìm kiếm, các form mà người dùng submit, các tương tác xã hội, các thông tin liên quan đến tài chính,... Không những vậy, do trình duyệt web là môi trường cho các hoạt động mạng nên qua đó, nó có thể dễ dàng thực hiện giao tiếp với internet. Bên cạnh đó,  thiết kế của trình duyệt cho phép tiện ích mở rộng của bên thứ ba có khả năng can thiệp sâu vào phần lõi của trình duyệt qua các API đặc quyền. Nói cách khác, các tiện ích mở rộng dễ dàng thực hiện tự động các hành vi mạng hoặc tiếp xúc với thông tin nhậy cảm của người dùng trong khi người dùng khó có thể phát hiện ra. Bởi vì điều này, chúng đã trở thành công cụ chính để những kẻ tấn công sử dụng để thực hiện các hành vi không mong muốn hoặc độc hại đối với người dùng trên trình duyệt.

Cụ thể, một browser extension độc hại có thể thực hiện các khả năng:

* Thu thập tất cả thông tin mà người dùng tạo ra khi sử dụng trình duyệt như: lịch sử duyệt web, cookie, nội dung tìm kiếm, form submit,  tài khoản ngân hàng, …, sau đó gửi các thông tin này đến server của bên thứ ba - Ăn cắp thông tin của người dùng
* Thay đổi DOM, inject javascript code,... Qua đó, nó có thể thay đổi nội dung vốn có của trang web, chèn quảng cáo, lừa đảo hay các đoạn mã độc vào trang web.
* Tạo mới, chặn, sửa, chuyển hướng các luồng request xuất phát từ  trình duyệt. Mục đích hướng đến có thể là botnet, lừa đảo, phát tán phần mềm độc hại,...
* Nhận và thực hiện lệnh từ server.
* …...

Tóm lại, tiện ích mở rộng - browser extension đang trở thành mối nguy hiểm đáng lo ngại nhất của trình duyệt web. Không những vậy, chúng còn đang được những kẻ tấn công không ngừng nghiên cứu và phát triển thêm.

## 1.3. Mục tiêu của đề tài

Từ sự nguy hiểm mà các tiện ích mở rộng của trình duyệt gây ra, mục tiêu của đề tài là đề xuất mô hình và thuật toán phát hiện hành vi xấu, độc hại của môdun mở rộng cho trình duyệt, cụ thể, đó là hành vi độc hại có liên quan đến hoạt động mạng như ăn cắp thông tin, thay đổi các luồng request,... Mô hình này được thiết kế và xây dựng dành cho người dùng sử dụng.

## 1.4. Sản phẩm của khóa luận

Nghiên cứu lý thuyết:

* Tìm hiểu phương pháp hoạt động của các extension độc hại, cách chúng thu thập thông tin của người dùng, cách chúng làm việc với các luồng request, response,...
* Phát triển mô hình và thuật toán phát hiện các hành vi độc hại có liên quan đến hoạt động mạng của tiện ích mở rộng cho trình duyệt:
  + Nghiên cứu phương pháp thu thập các hành vi liên quan đến hoạt động mạng của một extension
  + Từ bộ các hành vi đã thu thập được ở trên, đưa ra thuật toán phân tích để xác định các hành vi độc hại

Phát triển sản phẩm: Dựa trên nghiên cứu lý thuyết, sản phẩm của khóa luận gồm:

* Công cụ và hệ thống thu thập nhật ký hành vi mạng của tiện ích mở rộng cho trình duyệt
* Công cụ phân tích nhât ký hành vi mạng và phát hiện hành vi độc hại

## 1.5. Nội dung của báo cáo khóa luận

Phần còn lại của báo cáo sẽ trình bầy những nội dung sau:

* Chương 2: Browser Extension - Tiện ích mở rộng cho trình duyệt.

Chương này sẽ trình bày về cấu trúc của một extension, giới thiệu và phân tích một số hành vi độc hại của extension trong thực tế. Đồng thời chương này cũng giới thiệu một số cách mà những kẻ tấn công sử dụng để phát tán extension độc hại.

* Chương 3: Thu thập nhật ký hành vi mạng của tiện ích mở rộng cho trình duyệt

Chương này sẽ trình bày chi tiết về kiến trúc của hệ thống thu thập nhật ký hành vi mạng của tiện ích mở rộng cho trình duyệt

* Chương 4: Phát hiện hành vi độc hại của tiện ích mở rộng cho trình duyệt

Chương này sẽ trình bày các phương pháp được sử dụng để phát hiện hành vi độc hại của tiện ích mở rộng cho trình duyệt.

## 1.6. Tiềm năng của đề tài

Trong thực tế, có rất nhiều nghiên cứu và dự án về phát hiện hành vi độc hại của phần mở rộng cho trình duyệt. Cụ thể như:

* Ex-Ray: Detection of History-Leaking Browser Extensions

(<https://www.ftc.gov/system/files/documents/public_comments/2017/11/00043-141901.pdf>)

Dự án này đề cập đến việc xây dựng một hệ thống tự động cài đặt và kiểm tra các tiện ích mở rộng trên Chrome store sử dụng phương pháp Honeypot và machine leaning. Mục đích là phát hiện các hành vi ăn cắp lịch sử duyệt web của người sử dụng.

* Hulk: Eliciting Malicious Behavior in Browser Extensions

(<http://www.icir.org/vern/papers/hulk-usesec14.pdf>)

Dự án này giới thiệu Hulk, một hệ thống sử dụng HoneyPages kết hợp với việc phân tích tĩnh và phân tích động để tìm ra các hành vi nguy hiểm của extension

* Malicious Browser Extensions at Scale: Bridging the Observability Gap between Web Site and Browser

(<https://cseweb.ucsd.edu/~ldekoven/publications/malicious_browser-extensions_at_scale.pdf> )

Đây là một dự án của Facebook nhằm nỗ lực ngăn chặn các hành vi độc hại của tiện ích mở rộng cho trình duyệt đối với người dùng của họ như spam tin nhắn, mã độc,...

* Extended Tracking Powers: Measuring the Privacy Diffusion Enabled by Browser Extensions

(<https://www.securitee.org/files/extendedtracking_www2017.pdf> )

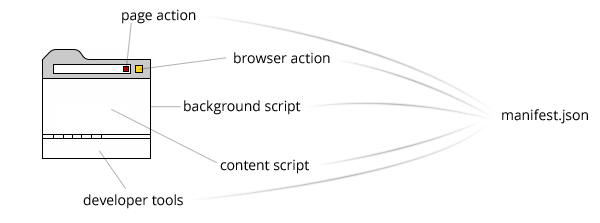
Đây là một dự án thông qua việc quan sát lưu lượng mạng để phát hiện hành vi độc hại của extension. Ngoài ra, họ còn thiết kế một Extension có tên gọi là BrowsingFog có tác dụng tạo ra các thông tin giả để che dấu các thông tin thật của người dùng khi bị một extension độc hại ăn cắp thông tin.

Tuy nhiên, hầu hết tất các nghiên cứu đó đều được đặt ở bên phía các nhà phát triển. Họ chủ động tải các tiện ích mở rộng của trình duyệt về rồi tiến hành phân tích, sau đó report kết quả và tiến hành gỡ bỏ trên store của trình duyệt. Phương pháp này có ưu điểm là số lượng extension được kiểm tra rất lớn, các thuật toán dùng để phát hiện hành vi độc hại được phát triển không ngừng và có độ chính xác ngày càng lớn. Mặc dù vậy, chính các nhà nghiên cứu phải thừa nhận rằng họ gặp phải một số vấn đề không nhỏ. Đầu tiên, có rất nhiều nguồn cung cấp tiện ích mở rộng bên ngoài store do các nhà phát triển trình duyệt web cung cấp, cho nên, các dự án phân tích hành vi độc hại không thể bao quát hết được tất cả các extension. Tiếp theo, thuật toán phát hiện hành vi độc vẫn có tỉ lệ nhất định bỏ sót extension độc hại do những lý do như: extension không thực hiện hành vi độc hại trong quá trình phân tích, extension sử dụng các hiện phát che dấu hành vi độc hại mà thuật toán phân tích không phát hiện ra,... Do đó, trong store của nhà phát triển trình duyệt web vẫn tồn tại một số extension có hành vi độc hại mà chưa bị phát hiện…

Phương phương được đưa ra trong đề tài này tiếp cận vấn đề phát hiện hành vi độc hại của extension dưới góc độ của người dùng. Do đó, nó có thể giải quyết được một phần vấn đề mà các nhà nghiên cứu hành vi độc hại của tiện ích mở rộng trình duyệt gặp phải đã nêu ở trên.

# Chương 2: Browser Extension – Môdun mở rộng cho trình duyệt.

## 2.1 Kiến trúc của một browser extension.



#### Hình 2.1: Kiến trúc của một extension

Về cơ bản, một extension gồm các thành phần chính:

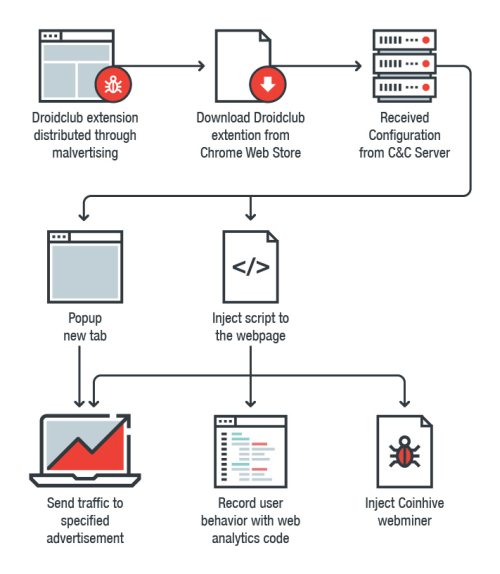
* Một file manifest. Đây là một file json nhằm cung cấp các thông tin cơ bản về extension như: Name, version, description, icon, backgroud, permisions, browser\_action,... Trong đó, trường “permisions” là một trong những trường quan trọng nhất. Một extension khi muốn sử dụng một API được trình duyệt cung cấp cần khai báo quyền tương ứng, ví dụ như nếu extension đó muốn thao tác với các luồng request và response của trình duyệt thì cần khai báo quyền “webRequests”, hay nếu extension muốn định vị vị trí của người dùng thì cần khai báo quyền “geolocation”. Quyền - permision có tác dụng làm rõ và hạn chế khả năng của một extension. Một số quyền sẽ được hiển thị cho người dùng trước khi cài đặt extension và nêu chi tiết trong “Cảnh báo quyền”.
* Trang nền: Mỗi extension có một trang nền (background page) không nhìn thấy được chạy bởi trình duyệt. Có hai loại - trang nền cố định và trang sự kiện. Cái đầu tiên hoạt động, mọi lúc. Cái thứ hai chỉ hoạt động khi cần. Google khuyến khích các nhà phát triển sử dụng các trang sự kiện, vì cái này giúp tiết kiệm bộ nhớ và cải thiện hiệu suất tổng thể của trình duyệt. Tuy nhiên, cũng cần phải biết rằng đây cũng là nơi bạn nên đặt logic chính và những khởi tạo của bạn. Thông thường, trang nền/script đóng vai trò là cầu nối giữa các phần khác nhau của extension.
* Script nội dung: nếu bạn cần truy cập vào DOM của trang hiện tại, bạn cần phải sự dụng một script nội dung (content script). Code được chạy bện trong ngữ cảnh của trang web hiện tại, có nghĩa là nó sẽ được thực thi với mỗi lần làm mới
* Gia diện người dùng: là một popup hoặc một trang tab mới để người dùng có thể tương tác với extension.

Có thể nói, extension có thiết kế tương tự như một trang web nhưng có thể sử dụng API của trình duyệt cung cấp và có thể chạy ngầm trên nền trình duyệt.

## 2.2 Một số extension độc hại trong thực tế và phương thức hoạt động của chúng

Các tiện ích mở rộng cho trình duyệt bị phát hiện là độc hại đang ngày càng nhiều. Chúng sử dụng các phương thức ngày càng đa dạng và tinh vi hơn trong việc thực hiện hành vi độc hại của mình. Nội dung bên dưới sẽ giới thiệu một số extension chứa các hành vi độc hại tiêu biểu trong thời gian gần đây

Cuối năm 2017, các nhà nghiên cứu của  Trend Micro Cyber Safety Solutions team đã phát hiện một dạng botnet mới dưới hình thức là một extension và đã đặt tên cho chúng là Droidclub. Botnet này được sử dụng để tiêm quảng cáo và mã độc đào tiền ảo vào các trang web mà nạn nhân truy cập. Ngoài các tính năng trên, Droidclub còn tiêm một đoạn script vào trang web để ghi lại nội dung mà người dùng tương tác với trang web. Qua đó, kẻ tấn công có thể biết được nội dung mà người dùng nhập vào form hay các tương tác khác trên trang web. Có tổng cộng 89 Droidclub extension được tìm thấy trên Chrome store và  uớc tính có gần 500000 người dùng bị nhiễm extension độc hại này.



#### Hình 2.2: Phương thức hoạt động của Droidclub

Cụ thể về các hành vi độc hại của Droidclub extension. Đầu tiên, Droidclub sẽ định kỳ bật lên một tab mới hiển thị quảng cáo trên web. URL và tần suất đều được gửi như một phần của thông tin cấu hình từ máy chủ C&C. Hiện tại, phần mềm độc hại này đang được sử dụng để hiển thị quảng cáo chất lượng thấp (chẳng hạn như quảng cáo cho trang web khiêu dâm) hoặc bộ công cụ khai thác.



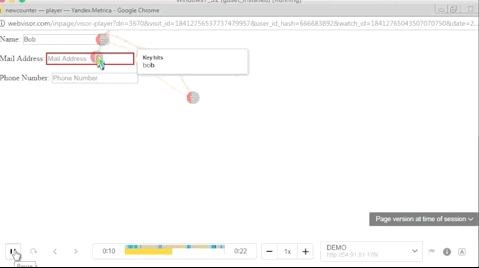
#### Hình 2.3: Nội dung lệnh được gửi đến từ máy chủ điều khiển

Bên cạnh đó, Droidclub cũng có thể sửa đổi nội dung của các trang web, một trong số các sửa đổi là thay link trong trang web bằng một liên kết khác. Các liên kết này cũng đi tới quảng cáo. Quảng cáo trong trang web gốc cũng được thay thế bằng quảng cáo do kẻ tấn công chọn. Kẻ tấn công đằng sau Droidclub có thể đang sử dụng botnet này để tăng số lần hiển thị quảng cáo một cách giả tạo, dẫn đến tăng lượt xem và doanh thu.



#### Hình 2.4: Droidclub chèn quảng cáo và thay đổi nội dung trang web

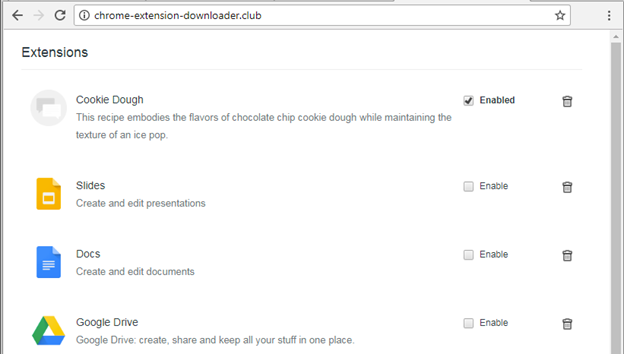
Tiếp theo, Droidclub  có thể tiêm một đoạn code javascript có tác dụng ghi lại hành động của người dùng như nhấp chuột, cuộn và tổ hợp phím. Qua đó, kẻ tấn công có thể lấy cắp dữ liệu được nhập vào các biểu mẫu như tên, số thẻ tín dụng, số CVV, địa chỉ email và số điện thoại.



#### https://blog.trendmicro.com/trendlabs-security-intelligence/files/2018/01/droidclub7.pngHình 2.5: Demo mã độc ghi lại hành động của người dùng

#### Hình 2.6: Mã độc đào tiền ảo của Droidclub

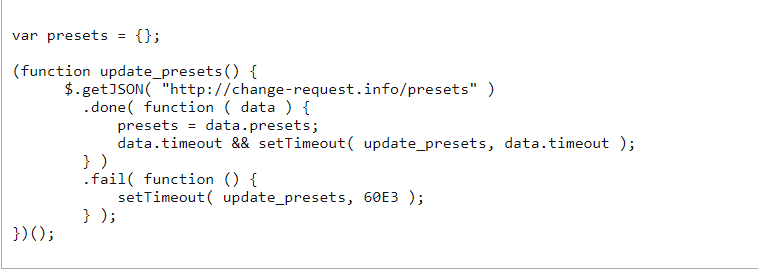
Một hành vi độc hại nữa của Droidclub  là tiêm mã đào tiền ảo Coinhive vào các trang web đã truy cập, biến trình duyệt thành trình khai thác Monero.



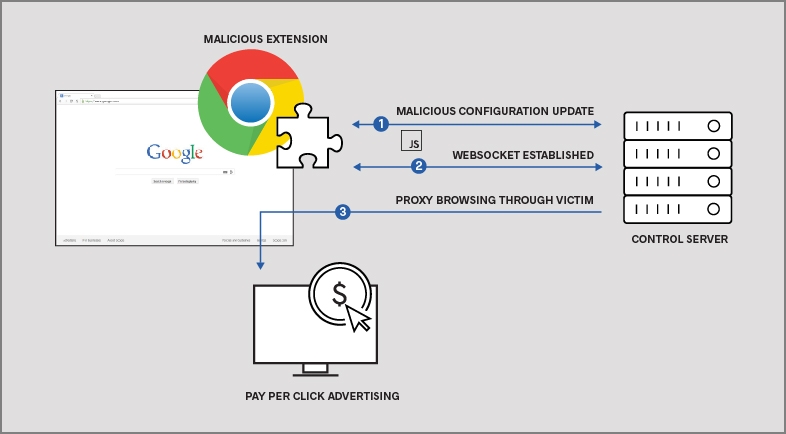
#### Hình 2.7: Trang extension giả mà Droidclub tạo ra để đánh lừa ngươi dùng

Tính năng cuối cùng của Droidclub được thiết kế để người dùng sẽ khó khăn hơn khi gỡ cài đặt và báo cáo phần mở rộng độc hại. Nếu tiện ích mở rộng phát hiện thấy người dùng đang cố gắng báo cáo tiện ích (bằng cách kiểm tra xem người dùng có đang truy cập URL khớp với chuỗi “https://chrome.google.com/webstore/report/([az]+)”) hay không, thay vào đó, người dùng được chuyển hướng đến trang giới thiệu của tiện ích mở rộng của họ. Cố gắng xóa tiện ích mở rộng thông qua trang quản lý tiện ích mở rộng của Chrome (nằm ở chrome: // extensions /), tiện ích chuyển hướng người dùng đến trang giả, dẫn người dùng tin rằng tiện ích mở rộng đã được gỡ cài đặt, ngay cả khi nó vẫn nằm trong trình duyệt của người dùng.

Đầu năm 2018, các nhà nghiên cứu tại ICEBRG Inc đã phát hiện ra 4 extension độc hại, ảnh hưởng đến hơn nửa triệu người dùng. Qua quá trình phân tích một extension tên là Change HTTP Request Header, họ phát hiện rằng bản thân extension không chứa bất kỳ mã độc hại công khai nào. Tuy nhiên, theo thiết kế, extension này có khả năng nhận và thực thi mã javascript từ server. Cụ thể, trong mã javascript của bản thân extension có một hàm được gọi là  ‘update\_presets()’, hàm này có chức năng downloads một file JSON từ server: ‘change-request[.]info’ rồi sau đó thực thi file json đó như một file javascript. Bên cạnh đó, một phần của mã được thực thi có sẵn trong extension sẽ kiểm tra sự hiện diện của các công cụ gỡ lỗi gốc của Chrome - debugging tools (chrome: // inspect / và chrome: // net-internals /) và nếu được phát hiện, extension sẽ tạm dừng thực hiện thực thi file json vừa tải. Điều này rất có thể là một kỹ thuật chống phân tích được thực hiện bởi kẻ tấn công để tránh phát hiện và kéo dài thời gian tồn tại của extension độc hại.



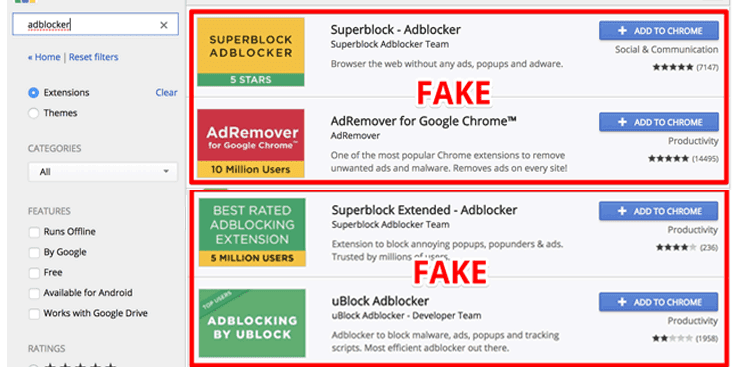
#### Hình 2.8 Hàm nhận và thực thi mã từ xa chứa trong extension

File JSON được tải về thực chất là một đoạn mã javascript đã bị obfuscate. Một khi inject vào extension thành công, mã JavaScript  độc hại này sẽ tạo một WebSocket tunnel với server ‘change-request[.]info’ sau đó bắt đầu nhận lệnh. Lệnh được gửi đến từ server sẽ chứa đường link dẫn đến trang web quảng cáo trả tiền cho mỗi nhấp chuột (PPC).

#### Hình 2.9: Sơ đồ hoạt động của Extension Change HTTP Request Header

Extension sẽ buộc hệ thống nạn nhân truy cập vào các trang web trên nhằm kiếm lợi

Gần đây nhất, ngày 17/4 vừa qua, Andrey Meshkov, nhà đồng sáng lập Adguard đã phát hiện 5 tiện ích mở rộng nguy hiểm là phiên bản ‘nhái’ của một số trình chặn quảng cáo nổi tiếng hợp pháp  và có ít nhất 20 triệu người dùng đã cài đặt các tiện ích này. Cụ thể đó là 5 extension:

* AdRemover cho Google Chrome ™ (hơn 10 triệu người dùng)
* uBlock Plus (hơn 8 triệu người dùng)
* [Fake] Adblock Pro (hơn 2 triệu người dùng)
* HD cho YouTube ™ (hơn 400.000 người dùng)
* Webutation (hơn 30.000 người dùng)

#### Hình 2.10: Các Extension độc hại giả dạng Extension chặn quảng cáo

Nhiệm vụ của các extension này đơn giản là ăn cắp thông tin của người dùng sau đó gửi lại cho máy chủ điều khiển

Tháng 6/2017, nhóm chuyên gia bảo mật đến từ phòng An toàn thông tin trực thuộc VCCorp đã tình cờ tìm ra và lần theo dấu vết của một đường dây chiếm đoạt thông tin quy mô lớn tại Việt Nam.


Extension Äá»c háº¡i giáº£ dáº¡ng Internet Download Manager ÄÃ£ tá»n táº¡i má»t thá»i gian khÃ¡ dÃ i trÃªn Chrome Web Store.
LÆ°u Ã½: tÃªn tÃ¡c giáº£ cá»§a extension nÃ y ÄÃ£ ÄÆ°á»£c xÃ¡c nháº­n lÃ  giáº£ máº¡o, má»¥c ÄÃ­ch hiá»n Äang ÄÆ°á»£c lÃ m rÃµ.


#### Hình 2.11: Các Extension độc hại giả dạng Extension IDM


Hacker ÄÃ£ chiáº¿m ÄÆ°á»£c lÆ°á»£ng lá»n tÃ i khoáº£n Äiá»n tá»­, trong ÄÃ³ cÃ³ tÃ i khoáº£n Vietcombank. Cáº§n pháº£i lÆ°u Ã½ rÃµ rÃ ng ráº±ng ngÃ¢n hÃ ng khÃ´ng cÃ³ lá»i trong chuyá»n nÃ y, há» thá»ng cá»§a há» váº«n an toÃ n. Hacker ÄÃ£ khai thÃ¡c tá»« chÃ­nh lá» há»ng trÃªn trÃ¬nh duyá»t cá»§a ngÆ°á»i dÃ¹ng Äá» táº¥n cÃ´ng.

 
Cụ thể, một Extension độc hại giả dạng Internet Download Manager  đã lấy cắp được thông tin đăng nhập (Username/Password) của khoảng 55.000 tài khoản Facebook, 6.000 tài khoản Google, 5.000 tài khoản Yahoo và đáng sợ nhất là hơn 5 triệu cookie các trang phổ biến như Facebook, Google Mail, Yahoo Mail, Hotmail hay cả PayPal.

#### Hình 2.12: Các thông tin mà Extension độc hại thu được

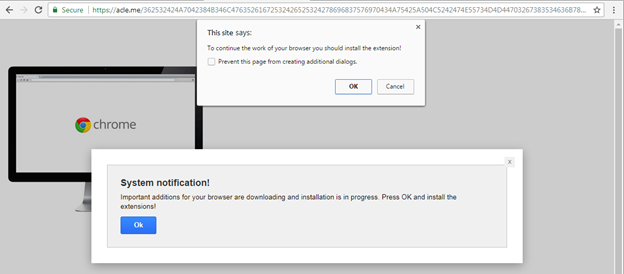
## 2.3 Một số cách thức phát tán extension độc hại

Với việc các nhà phát triển trình duyệt đang ngày càng thắt chặt chính sách bảo mật đối với tiện ích mở rộng, tỉ lệ xuất hiện của một extension độc hại trên store chính thức của nhà phát triển trình duyệt đã giảm đi đáng kể. Tuy nhiên, để đánh lừa người dùng cài đặt các extension độc hại, những kẻ tấn công nghĩ ra rất nhiều phương pháp, chúng ngày càng đa dạng và tinh vi hơn. Cụ thể, ngươi dùng có thể bị nhiễm extension độc hại thông qua sử dụng các phần mềm lậu (crack). Khi người dùng tải về phần mềm lậu từ trang mạng bất kỳ (do hacker tải lên), trong các file crack sẽ đính kèm một file thực thi nhiệm vụ theo trình tự sau: tắt trình duyệt nếu đang chạy, tạo kết nối tới trang chứa extension và cuối cùng là tải extension về và cài đặt trong máy nạn nhân.


Báº£n log ghi láº¡i viá»c file thá»±c thi tá»± Äá»ng táº£i vá» vÃ  cÃ i Äáº·t extension.


#### Hình 2.13: Một chương trình crack chứa mã cài đặt Extension độc hại

Một cách thực phát tán khác đó là sử dụng các trang web lừa đảo mời người dùng cài đặt một extension nào đó. Có thể là cài extension thì mới có thể xem được nội dung, cài đặt để lướt web nhanh hơn,... Nếu người dùng nhấn đồng ý, trình duyệt sẽ tự động tải về và cài đặt extension độc hại



#### Hình 2.14: Trang web yêu cầu cài đặt Extension

Một phương pháp phổ biến nữa đó là cài đặt extension độc hại thông qua mã độc trên máy tính. Cụ thể, khi máy tính bị nhiễm mã độc, mã độc đó sẽ thực hiện cài đặt extension độc hại lên trình duyệt của người dùng. Phương thức để một máy tính bị nhiễm mã độc rất đa dạng, có thể là mã độc được đính kèm trong phần mềm crack, đính kèm trong file word, pdf, hay lây lan qua mạng LAN,...

# Chương 3. Thu thập nhật ký hành vi mạng của môđun mở rộng

## 3.1. Phân loại hành vi của mô đun mở rộng cho trình duyệt

Các hành vi của một extension có thể được chia làm hai dạng chính, đó là:

Hành vi mạng: là tất cả các hành vi liên quan đến các luồng request và response, bao gồm:

* Chặn request gửi đi: xóa request trước khi request đi ra khỏi trình duyệt, thường được dùng để ngăn người dùng truy cập vào trang web ( block web). Với mục đích tốt, hành vi này giúp chặn web đen, kiểm soát truy cập internet và duyệt web an toàn. Tuy nhiên, hành vi có thể bị lạm dụng để thực hiện mục đích xấu.
* Chuyển hướng request gửi đi: thay đổi url của request trước khi request đi ra khỏi trình duyệt. Hành vi này tương tự như hành vi chặn request, có thể được sử dụng với nhiều mục đích khác nhau.
* Sửa đổi request: Hiện tại, các nhà phát triển trình duyệt web đã không cho phép extension sửa đổi body của request nữa. Cho nên, một extension chỉ có thể thay đổi các trường headers của request.
* Tạo request mới: sử dụng một số phương pháp như Ajax để tạo một request. Đây là hành vi đáng chú ý nhất của extension vì nó có thể sử dụng cách thức này để giao tiếp với máy chủ điều khiển, bao gồm các hành vi độc hại như nhận lệnh từ máy chủ hoặc gửi thông tin người dùng đã thu thập được.

Hành vi DOM: là tất cả các hành vi liên quan đến nội dung của trang web mà người dùng truy cập, bao gồm:

* Thay đổi nội dung hiển thị của trang web: sửa đổi các thẻ HTML nhằm thay đổi nội dung trang. Trong thực tế, extension chặn quảng cáo sẽ xóa thẻ HTML chứa quảng cáo, extension dịch sẽ thay đổi nội dung của các thẻ HTML chứa nội dung. Còn về phía một extension độc hại, hành vi này có thể dùng để thêm quảng cáo vào trang web, thay link trong trang web thành liên kết quảng cáo hoặc mã độc,...
* Thay đổi mã javascript của trang web: Xóa mã javascript ban đầu của trang web. Mục đích của hành động này có thể là xóa mã js chặn hiển thị trang web (Remove Render-Blocking JavaScript) hoặc đơn giản là làm trang web hoạt động không bình thường.
* Thêm mã javascript mới vào trang web. Đây có thể coi là hành động nguy hiểm. Mặc dù hành vi này vẫn có thể thực hiện với mục đích tốt nhưng một extension có thể thêm một số mã javascript độc hại như: mã đào tiền ảo, mã js theo dõi hành vi của người dùng,...

## 3.2. Mục tiêu của thu thập nhật ký hành vi mạng

Như đã nói ở phần đầu của báo cáo khóa luận, mục tiêu của đề tài là phát hiện hành vi độc hại có liên quan đến hoạt động mạng của extension. Do đó, ta cần thu thập các hành vi liên quan đến mạng của một extension để phân tích và xác định các hành vi độc hại lẫn trong đó. Các hành vi cần thu thập bao gồm:

* Các request bị extension chặn lại
* Các request bị extension chuyển hướng đến một địa chỉ khác. Thông tin thu thập bao gồm cả địa chỉ cũ và địa chỉ mới
* Các request bị extension sửa đổi. Thông tin thu thập bao gồm nội dung ban đầu và sau khi bị sửa đổi của request
* Các request do extension tạo ra

## 3.3. Kiến trúc hệ thống thu thập nhật ký hành vi mạng của môdun mở rộng

Hệ thống thu thập nhật ký hành vi mạng của extension gồm ba bộ phận chính, đó là: bộ thu thập nhật ký mạng tại trình duyệt, bộ thu thập nhật ký mạng tại proxy và bộ so sánh, trích suất thông tin. Cả hệ thống hoạt động theo sơ đồ sau:

#### Hình 3.1: Kiến trúc hệ thống thu thập hành vi mạng của môdun mở rộng cho trình duyệt

Cụ thể, trong trình duyệt có hai nguồn tạo ra request, đó là:

* Request do các tab trong trình duyệt tạo ra. Các tab này sinh ra request dựa theo các hành vi tương tác của người dùng như duyệt web, submit form, … Đây là nguồn tạo ra request chính trong trình duyệt
* Request do extension tạo ra. Một số extension khi hoạt động cần gửi thông tin đến máy chủ điều khiển để xử lý, thông tin cần gửi đi sẽ được chứa trong request.

Một extension khi được cấp quyền sẽ có khả năng tương tác với các luồng request do các tab tạo ra trước khi các request đó đi ra khỏi trình duyệt. Trong khi tương tác, nó có thể tiến hành sửa đổi header, chặn, chuyển hướng request hoặc bí mật thu thập các thông tin chứa trong request như URL ( lịch sử duyệt web), nội dung requests (thông tin người dùng từ form submit,...). Tại đây, các hành vi sửa đổi, chặn và chuyển hướng request sẽ được bộ thu thập nhật ký mạng tại trình duyệt ghi lại và gửi cho bộ so sánh và trích suất thông tin. Các luồng request ra khỏi trình duyệt sẽ bao gồm các request do các tab tạo ra và cả request do extension tạo ra, nó sẽ phải đi qua bộ thu thập nhật ký mạng tại proxy. Tại đây, tất cả các request sẽ được ghi lại sau đó chuyển cho bộ so sánh và trích suất thông tin để xử lý. Bộ so sánh và trích suất thông tin sẽ lọc ra các request do extension tạo ra. Tóm lại, đầu ra của bộ so sánh và trích suất thông tin sẽ là bốn hành vi mạng cần thu thập. Chi tiết về chức năng và kiến trúc của từng bộ phận sẽ được trình bày trong các phần sau.

### 3.3.1 Bộ thu thập nhật ký mạng tại trình duyệt

Bộ thu thập nhật ký mạng tại trình duyệt được thiết kế dưới dạng một extension. Nó có nhiệm vụ chính là thu thập ba loại hành vi sửa đổi header, chặn và chuyển hướng request.

Trước khi đi vào nguyên lý hoạt động của extension thu thập nhật ký hành vi mạng, ta cần hiểu rõ vòng đời của một requests sinh ra bởi trình duyệt - Life cycle of requests. Các nhà phát triển trình duyệt định nghĩa một API tên là webRequest dùng để quản lý vòng đời của request, theo đó, kể từ khi bắt đầu cho đến khi hoàn thành, một request sẽ có những sự kiện sau:

* **onBeforeRequest** :

Sự kiện xảy ra khi một request được khởi tạo. Sự kiện này được gọi trước khi bất kỳ kết nối TCP nào được thực hiện và có thể được sử dụng để hủy hoặc chuyển hướng các request.

* **onBeforeSendHeaders**:

Sự kiện này xảy ra khi tiêu đề ( headers) của request được khởi tạo. Sự kiện này nhằm mục đích cho phép các tiện ích mở rộng thêm, sửa đổi và xóa headers của request. Sự kiện onBeforeSendHeaders được chuyển cho tất cả các extension đã đăng ký quyền API webRequest, vì vậy những extension khác nhau có thể sửa đổi cùng một requests. Sự kiện này có thể được sử dụng để hủy yêu cầu

* **onSendHeaders:**

Sự kiện này xảy ra sau khi tất cả các tiện ích mở rộng đã hoàn tất sửa đổi các tiêu đề yêu cầu và trình bày phiên bản cuối cùng. Sự kiện này được kích hoạt trước khi các tiêu đề được gửi đi. Sự kiện này không cho phép sửa đổi hoặc hủy bỏ request.

* **onHeadersReceived**:

Sự kiện này xảy ra mỗi khi một tiêu đề phản hồi HTTP (S) được nhận. Do chuyển hướng và yêu cầu xác thực, điều này có thể xảy ra nhiều lần theo yêu cầu. Sự kiện này nhằm mục đích cho phép tiện ích mở rộng thêm, sửa đổi và xóa tiêu đề phản hồi, chẳng hạn như tiêu đề Set-Cookie. Các chỉ thị bộ nhớ đệm được xử lý trước khi sự kiện này được kích hoạt, do đó việc sửa đổi các tiêu đề như Cache-Control không ảnh hưởng đến bộ nhớ cache của trình duyệt. Nó cũng cho phép hủy bỏ hoặc chuyển hướng yêu cầu.

* **onAuthRequired**:

Sự kiện này xảy ra khi request yêu cầu xác thực người dùng. Sự kiện này có thể được xử lý đồng bộ để cung cấp thông tin đăng nhập xác thực. Lưu ý rằng tiện ích có thể cung cấp thông tin đăng nhập không hợp lệ. Cẩn thận không nhập vòng lặp vô hạn bằng cách liên tục cung cấp bằng chứng xác thực không hợp lệ. Điều này cũng có thể được sử dụng để hủy yêu cầu.

* **onBeforeRedirect**:

Sự kiện này xảy ra khi chuyển hướng sắp được thực hiện. Một chuyển hướng có thể được kích hoạt bởi một mã phản hồi HTTP hoặc bởi một phần mở rộng. Sự kiện này là thông tin và được xử lý không đồng bộ. Nó không cho phép bạn sửa đổi hoặc hủy bỏ yêu cầu.

* **onResponseStarted**:

Sự kiện này xảy ra khi byte đầu tiên của nội dung phản hồi được nhận. Đối với yêu cầu HTTP, điều này có nghĩa là dòng trạng thái và tiêu đề phản hồi có sẵn. Sự kiện này là thông tin và được xử lý không đồng bộ. Nó không cho phép sửa đổi hoặc hủy bỏ yêu cầu.

* **onCompleted**:

Sự kiện xảy ra khi một request được xử lý thành công

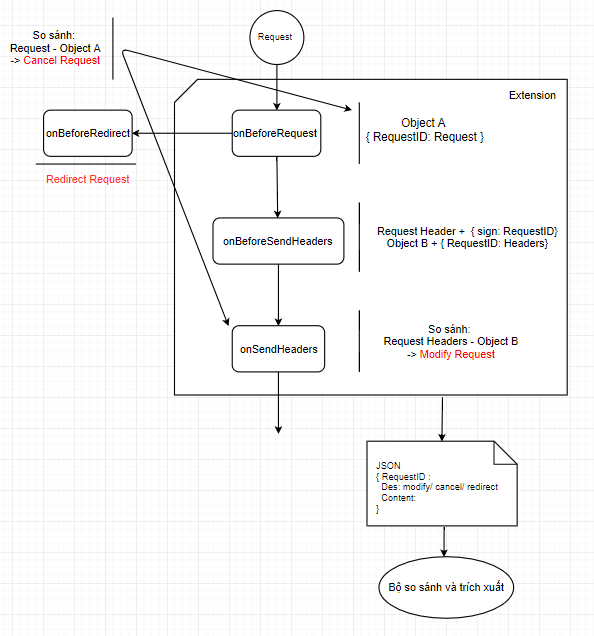
* **onErrorOccurred**:

Sự kiện xảy ra khi một request không được xử lý thành công

Một thông tin nữa cần lưu ý đó là phương pháp giải quyết xung đột giữa các extension trong trình duyệt web. Như đã biết, các extension đều có thể can thiệp vào vòng đời của một requests, vì vậy cần một bộ các quy tắc để giải quyết xung đột giữa các extension:

* Request được coi là bị hủy nếu ít nhất một tiện ích mở rộng yêu cầu hủy request . Nếu tiện ích hủy request , tất cả tiện ích sẽ được thông báo bởi sự kiện onErrorOccurred.
* Chỉ một tiện ích mở rộng được phép chuyển hướng một request  hoặc sửa đổi tiêu đề tại một thời điểm. Nếu có nhiều hơn một tiện ích mở rộng cố gắng sửa đổi yêu cầu, tiện ích mở rộng được cài đặt gần đây nhất sẽ thắng và tất cả các tiện ích khác sẽ bị bỏ qua. Một phần mở rộng không được thông báo nếu lệnh của nó để sửa đổi hoặc chuyển hướng đã bị bỏ qua.

Tiếp theo, chúng ta sẽ đi vào nguyên lý hoạt động của extension thu thập nhật ký mạng tại trinh duyệt. Extension hoạt động theo sơ đồ sau:



#### Hình 3.2 : Kiến trúc của bộ thu thập hành vi mạng tại trình duyệt

Cụ thể, khi sự kiện onBeforeRequest xảy ra, extension sẽ lưu thông tin của request vào Object javascript có tên là A. Mỗi một phần tử của object sẽ là một request với key là requestID của request còn value sẽ là nội dung của request đó. Tiếp theo, khi sự kiện onBeforeSendHeaders xảy ra, extension sẽ lưu nội dung của request đó vào một Object javascript khác có tên là B. Mỗi một phần tử của object sẽ là một request với key là requestID của request còn value sẽ là nội dung của request đó. Đồng thời, extension sẽ tiến hành thêm một trường giá trị vào headers của request đó. Trường giá trị đó có key là “Sign”, value là requestID của request đó. Cuối cùng, khi sự kiện onSendHeaders xảy ra, extension thực hiện kiểm tra request hiện tại bằng cách so sánh headers của request hiện tại với headers của request có requestID tương ứng trong Object B phía trên. Nếu hai headers khác nhau, ta có thể xác định rằng request headers đã bị sửa đổi. Còn nếu không, xác định đó là requets bình thường, tiến hành xóa bản lưu của request đó trong hai Object A và B. Bên cạnh đó, nếu như sự kiện onBeforeRedirect xảy ra, extension sẽ kiểm tra id của request đó với requestID tương ứng trong Object A. Từ đó có được thông tin request nào đã bị chuyển hướng. Cuối cùng, sau một khoảng thời gian nhất định, extension sẽ kiểm tra những phần tử bên trong Object có thời gian tồn tại trên 1 phút mà chưa bị xóa, đó là những request đã bị hủy. Như vậy, extension đã thu thập được ba hành vi đó là sửa đổi headers, chuyển hướng và chặn request. Sau đó, extension sẽ gửi thông tin thu thập được cho bộ so sánh và trích suất thông tin để xử lý. Thông tin gửi đi có định dạng của một file json, bao gồm các trường: “requestID” chứa id của request, “des” chứa tên hành vi: modify, cancel, redirect và cuối cùng là trường “content” chứa nội dung chi tiết của các hành vi kẻ trên.

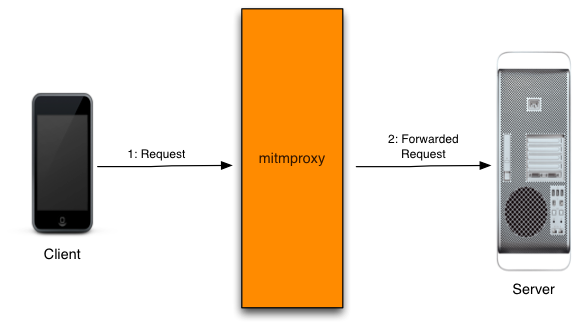
### 3.3.2 Bộ thu thập nhật ký mạng tại proxy

Bộ thu thập nhật ký mạng tại proxy là một proxy có tên gọi Mitmproxy. Nó có tác dụng ghi lại tất cả các luồng request đi qua nó, sau đó gửi các luồng request này đến bộ so sánh và trích xuất thông tin để xử lý. Tất cả dữ liệu mà proxy thu thập được là ở dưới dạng bản rõ, kể cả khi request sử dụng HTTPS.

Phương thức hoạt động của Mitmproxy:

* **Explicit HTTP:**

Với các luồng request sử dụng HTTP, proxy rất dễ dàng thu thập được nội dung giao tiếp giữa máy khách và máy chủ. Sơ đồ mitmproxy thực hiện giao tiếp bằng HTTP:

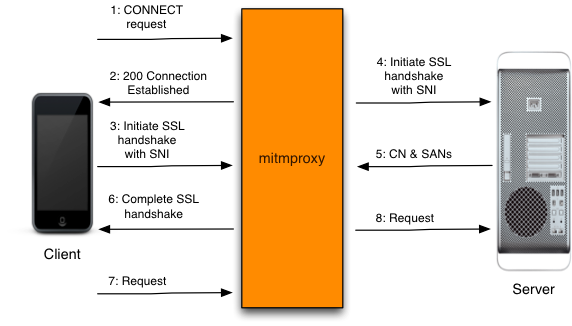


#### Hình 3.3: Sơ đồ hoạt động của proxy với phương thức HTTP

1. Máy khách kết nối với proxy và gửi đi một request
2. Mitmproxy kết nối với máy chủ và đơn giản là chuyển tiếp request đó.

* **Explicit HTTPS:**

Việc thu thập nội dung khi request sử dụng HTTPS sẽ khó khăn hơn. MITM trong tên của của Mitmproxy là viết tắt của Man-In-The-Middle. Ý tưởng cơ bản là giả vờ là máy chủ cho máy khách và giả vờ là máy khách với máy chủ, trong khi đó proxy ở giữa lưu lượng giải mã từ cả hai phía. Phần khó khăn là hệ thống chứng chỉ được thiết kế để ngăn chặn chính xác cuộc tấn công MITM, bằng cách cho phép bên thứ ba đáng tin cậy ký mã hóa chứng chỉ của máy chủ để xác minh rằng chúng là hợp pháp. Nếu chữ ký này không khớp hoặc là từ một bên không đáng tin cậy, một máy khách  bảo mật sẽ chỉ cần bỏ kết nối và từ chối tiếp tục. Tuy nhiên, Mitmproxy là một tổ chức phát hành chứng chỉ tin cậy, ta có thể đăng ký mitmproxy như một CA tin cậy với thiết bị theo cách thủ công. Sau khi đăng ký mitmproxy là một CA tin cậy, proxy này có thể thu thập được nội dung HTTPS dưới dạng rõ ràng. Sơ đồ mitmproxy thực hiện giao tiếp HTTPS:

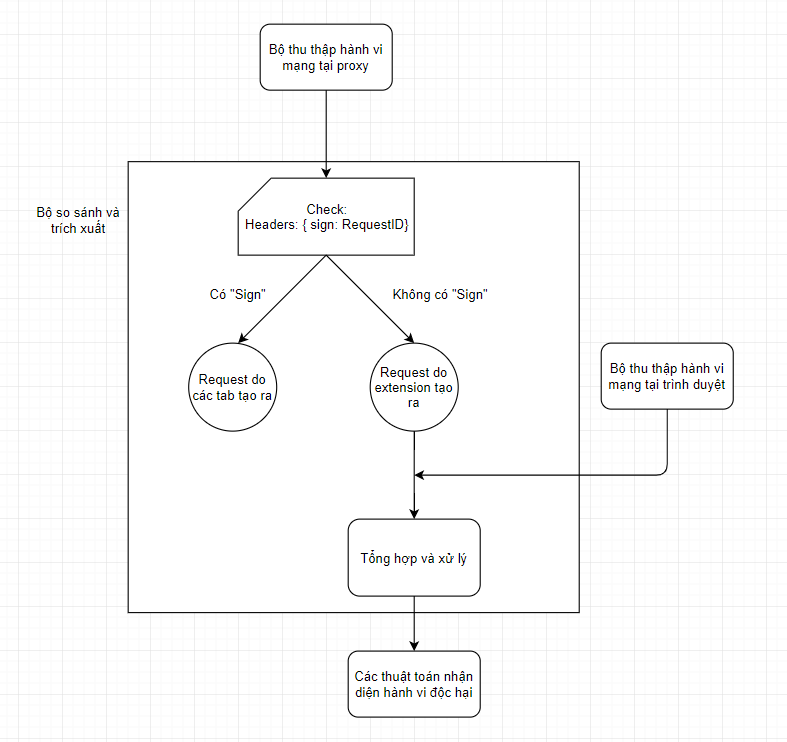


#### Hình 3.4: Sơ đồ hoạt động của proxy với phương thức HTTP

1. Máy khách tạo kết nối đến mitmproxy và đưa ra một HTTP CONNECT request
2. Mitmproxy trả về nội dung ”200 Connection Established”, như thể nó đã thiết lập một CONNECT pipe.
3. Máy khách tin rằng nó đang nói chuyện với máy chủ từ xa và khởi tạo kết nối TLS. Nó sử dụng SNI để chỉ ra tên máy chủ mà nó đang kết nối tới.
4. Mitmproxy kết nối với máy chủ và thiết lập kết nối TLS bằng tên máy chủ SNI được chỉ định bởi máy khách.
5. Máy chủ đáp ứng với chứng chỉ phù hợp, có chứa các giá trị CN và SAN cần thiết để tạo chứng chỉ chặn.
6. Mitmproxy tạo ra cert đánh chặn, và tiếp tục bắt tay TLS client bị tạm dừng trong bước 3 - Hoàn thành SSL HandShake
7. Máy khách request qua kết nối TLS đã được thiết lập.
8. Mitmproxy chuyển request đến máy chủ qua kết nối TLS được bắt đầu ở bước 4

Sau khi đã hoàn thành cài đặt Mitmproxy, proxy này sẽ ghi lại tất cả các luồng request xuất phát từ trình duyệt, sau đó gửi đến bô so sánh và trích xuất thông tin để xử lý.

### 3.3.3  Bộ so sánh và trích xuất



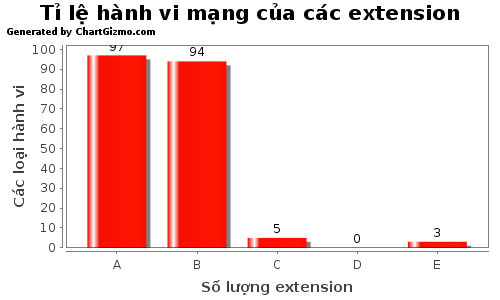
#### Hình 3.5: Kiến trúc của bộ so sánh và trích xuất

Bộ so sánh và trích xuất dựa vào dữ liệu mà hai bộ công cụ phía trên gửi về để tiến hành tìm ra các hành vi mạng của tiện ích mở rộng cho trình duyệt. Bộ thu thập nhật ký mạng tại trình duyệt sẽ gửi về ba hành vi mạng gồm: sửa đổi headers, chặn, chuyển hướng request dưới dạng json. Bộ so sánh và trích xuất sẽ đọc nội dung json đó và lưu hành vi mạng đã thu thập được lại để các thuật toán phát hiện hành vi độc hại phía sau xử lý tiếp. Bộ thu thập nhật ký mạng tại proxy gửi đến tất cả các request xuất phát từ trình duyệt. Các request đó bao gồm hai loại là request do các tab người dùng tạo ra và request do extension tạo ra. Có một quy tắc do nhà phát triển trình duyệt web đặt ra là một extension không thể tương tác với các luồng request do extension khác tạo ra. Nói cách khác, một extension chỉ có thể thực hiện hành vi sửa đổi headers đối với các request do các tab trong trình duyệt tạo ra. Bên cạnh đó, phần 3.3.1 đã đề cập đến việc extension thu thập nhật ký mạng có thêm một trường “sign: requestID” vào trong headers của tất cả các request nó có thể tương tác. Kết hợp các điều trên, ta có thể thấy rằng, khi một request đi ra khỏi trình duyệt, nếu trong headers của nó có chứa trường “sign” thì có nghĩa là request đó do các tab người dùng tạo ra, còn nếu như không có trường “sign”, đó là các request do extension tạo ra. Rất đơn giản, bộ so sánh và trích xuất sẽ kiểm tra các request đến từ trình duyệt và lọc ra các request không có “sign”, đó là các request do extension tạo ra. Như vậy, bộ so sánh và trích xuất đã thu thập được bốn hành vi mạng của extension bao gồm: sửa đổi headers, chặn, chuyển hướng và tạo mới request.

## 3.4 Thử nghiệm

Tiến hành cài đặt bộ các công cụ thu thập hành vi mạng của tiện ích mở rộng đồng thời cài đặt 100 extension ngẫu nhiên trong cửa hàng Google Chrome để thử nghiệm.

Sau đây là một số số liệu thống kê thu thập được



#### Hình 3.6: Bảng tỉ lệ hành vi mạng của các Extension

1. tỷ lệ extension có hành vi mạng
2. tỷ lệ extension có hành vi tạo thêm request
3. tỷ lệ extension có hành vi xóa request
4. tỷ lệ extension có hành vi chỉnh sửa request
5. tỷ lệ extension có hành vi chuyển hướng request

Trong đó, các hành vi xóa và chuyển hướng request chỉ xảy ra ở các extension đặc thù như chặn trang web, chặn quảng cáo. Không ghi nhận hành vi chỉnh sửa headers nào từ các extension thử nghiệm. Ngoài ra, nội dung gửi đi của các extension hầu như không bị mã hóa hoặc làm rối.

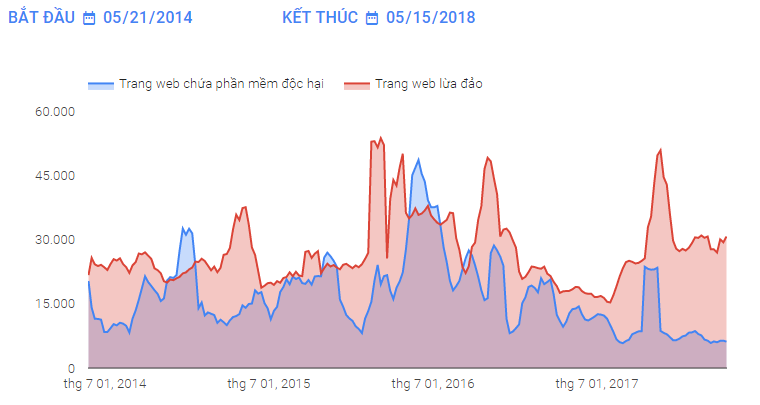
# Chương 4. Phát hiện hành vi mạng xấu độc của môđun mở rộng

Hiện tại ta đã thu thập được các hành vi của extension, bao gồm các hành vi chặn, chuyển hướng, thay đổi headers và tạo mới request. Sau đó, chúng ta sẽ sử dụng một bộ các phương pháp phân tích. Một hành vi được cho là độc hại nếu có ít nhất một trong các phương pháp nhận định hành vi đó độc hại.

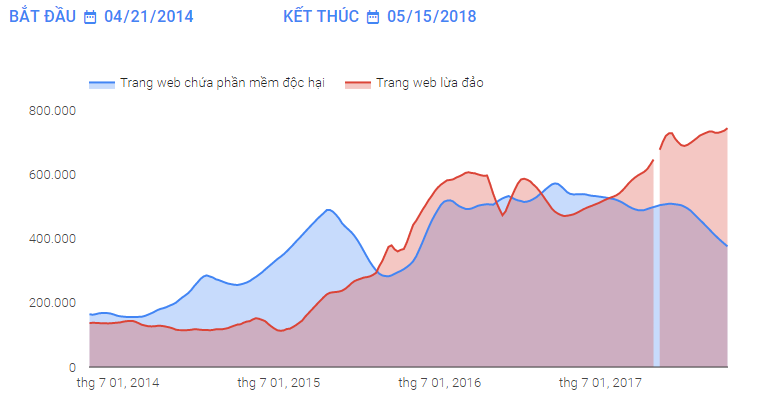
## 4.1. Phương pháp whitelist và blacklist

Điểm chung của hai phương pháp này là đều lấy địa chỉ có trong hành vi của extension để so sánh với một list các địa chỉ có sẵn. Cụ thể, phương pháp whitelist sẽ sử dụng một list các địa chỉ đáng tin cậy nhất, gồm có các địa chỉ thuộc về google, microsoft, kaspersky,... Danh sách whitelist là hữu hạn, chỉ có một số địa chỉ chắc chắn an toàn mới được cho vào danh sách. Khác với danh sách whitelist, danh sách blacklist chứa các địa chỉ dẫn đến các trang quảng cáo, các trang web chứa mã độc, hay các trang web chứa nội dung không lành mạnh,... Danh sách blacklist được lấy từ các nguồn cung đáng tin cậy về an toàn duyệt web và được cập nhật không ngừng. Nếu địa chỉ có trong hành vi của extension nằm trong danh sách whitelist, hành vi đó sẽ được coi như an toàn và không cần phân tính thêm. Ngược lại, nếu một địa chỉ nằm trong blacklist, hành vi đó sẽ được nhận định là độc hại.

## 4.2. Phương pháp dựa vào đánh giá URL

Phương pháp này sử dụng dịch vụ do bên thứ ba cung cấp để đánh gia độ an toàn của một URL chứa trong hành vi của extension. Ở đây, chúng ta sử dụng dịch vụ Safe Browsing của Google. Dịch vụ này của Google kiểm tra hàng tỷ URL, phần mềm và nội dung trên các trang web đó trong quá trình tìm kiếm các trang web không an toàn. Mỗi ngày, Duyệt web an toàn phát hiện hàng nghìn trang web mới không an toàn. Nhiều trang web trong số này là trang web hợp pháp đã bị kẻ tấn công xâm phạm. Trang web không an toàn nằm trong hai danh mục đe dọa quyền riêng tư và bảo mật của người dùng: lừa đảo và phần mềm độc hại.

#### Hình 4.1 Số trang web không an toàn được phát hiện mỗi tuần

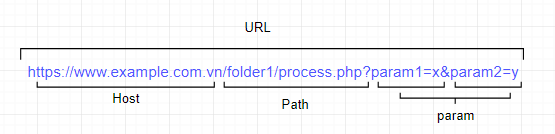


#### Hình 4.2 Số trang web bị coi là nguy hiểm theo dịch vụ duyệt web an toàn

Phương pháp này sẽ gửi URL lên dịch vụ duyệt web an toàn để tiến hành đánh  giá. Nếu dịch vụ này nhận định đây là một URL độc hại, có thể dẫn đến một trang web chứa mã độc hoặc lừa đảo, hành vi của extension sẽ bị coi là độc hại.

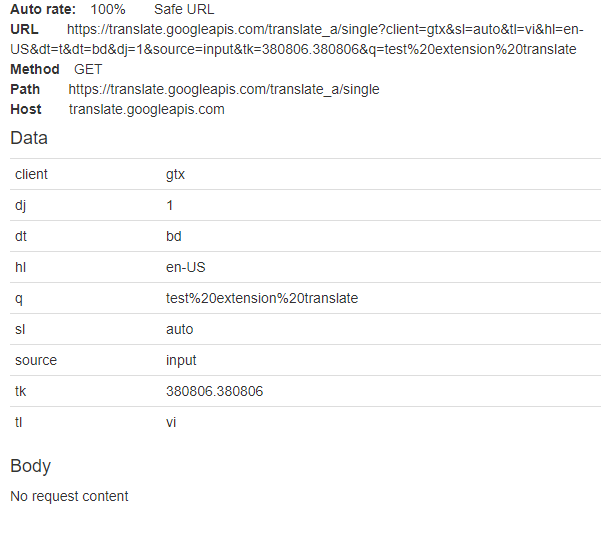
## 4.3. Phương pháp dựa vào phân tích nội dung của request

Phương pháp này được tạo ra để kiểm soát nội dung mà các extension gửi đi. Cụ thể, trong các request do extension tạo ra, chúng ta sẽ tiến hành lọc lấy các thông tin chứa trong đó:

* Nếu request có method GET, ta sẽ lọc lấy các param chứa trong URL
* Nếu request có method POST, ta sẽ lấy nội dung chứa trong phần body của request. 

#### Hình 4.3: Mô tả nội dung chứa trong một URL

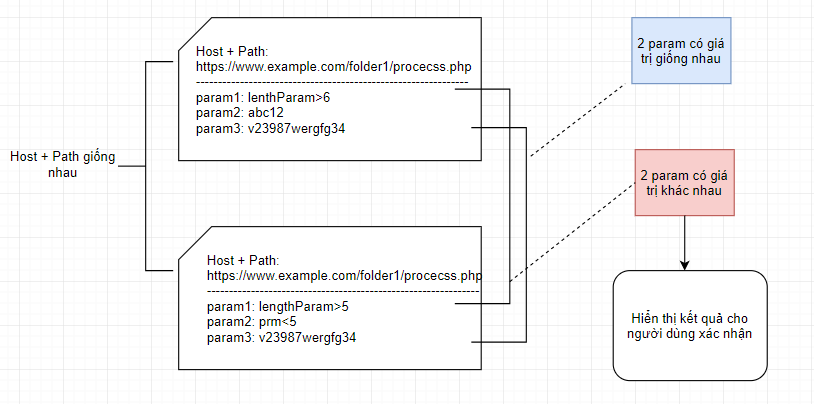
#### Hình 4.4: Nội dung trong phần body của một request



#### Hình 4.5 Nội dung chứa trong một URL có method GET

Sau khi đã lọc được các thông tin này, chúng ta sẽ sử dụng một số thuật toán để kiểm tra độ an toàn của dữ liệu gửi đi. Thuật toán trước hết sẽ tìm kiếm các chuỗi có độ dài lớn hơn 5 ký tự để tiến hành phân tích tiếp. Theo phân tích từ dữ liệu đã thu thập được các extension từ bộ thu thập nhật ký hành vi mạng, ta nhận thấy các tham số có độ dài nhỏ hơn 5 sẽ không thể chứa thông tin của người dùng, các trường email, tên đăng nhập hay password, số điện thoại đều quy định ít nhất có 6 ký tự. Việc lọc ra các chuỗi ký tự có độ dài lớn hơn 5 đã loại bỏ phần lớn các thông tin chứa trong request gửi đi của một extension. Tiếp theo, chúng ta sẽ tiến hành tìm kiếm các nội dung nhạy cảm có trong lượng thông tin còn lại như email, số điện thoại, url,... Nếu một extension gửi đi các thông tin nhậy cảm đó, có khả năng cao extension đó là độc hại. Các chuỗi ký tự có độ dài lớn hơn 5 mà hệ thống không nhận định được thông tin nhậy cảm chứa trong đó sẽ được lưu lại và chuyển cho phương pháp cuối cùng.

Ngoài ra, theo quan sát từ nội dung gửi đi của các extension, mỗi một đường dẫn (path) của URL sẽ chỉ tương ứng với một format các tham số nhất định.



#### Hình 4.6 Kiến trúc thuật toán so sánh format request

Ý tưởng của thuật toán là lưu lại định dạng gói tin gửi đi đầu tiên mà hệ thống thu thập được. Các request tiếp theo gửi đến cùng một địa chỉ sẽ được tiến hành so sánh với mẫu request đã lưu. Các tham số được gửi đi trong request nếu không thay đổi so với request ban đầu sẽ được coi là an toàn, còn nếu thay đổi, rất có khả năng đó là thông tin của người dùng. Phương pháp này nhằm loại bỏ một số tham số do bản thân extension tạo ra

Kết hợp hai thuật toán phân tích nội dung trên, lượng nội dung không xác định có trong request còn lại rất it, chúng sẽ được gửi đến phương pháp phân tích cuối cùng.

## 4.4 Phương pháp phân tích thông tin dựa vào người dùng

Như đã nói ở phần trước, có một số thông tin do extension gửi đi mà hệ thống không phân tích được, người dùng có thể xem danh sách các thông tin này để quyết định xem thông tin nhậy cảm của bản thân có nằm trong số đó hay không. Lượng thông tin không xác định này tồn tại rất ít nếu như trong trình duyệt của người dùng không có extension độc hại. Điều này sẽ không ảnh hưởng đến trải nghiệm của người dùng. Ngoài ra, tất cả các cảnh báo độc hại sẽ được thông báo cho người dùng để họ có quyền quyết định cuối cùng. Ví dụ như:

* Hệ thống thông báo một request đến một trang web nào đó đã bị chặn lại bởi extension, nếu người dùng không cài đặt bất kì extension chặn trang web nào, hành vi chặn request đó là độc hại.
* Hệ thống thông báo request headers đã bị sửa, nếu người dùng không phải đang dùng một extension sửa đổi headers để làm việc, hành vi đó chắc chắn là độc hại.
* Hệ thống thông báo một extension gửi đi một chuỗi thông tin chứa email, nếu người dùng đang không sử dụng một extension để đăng nhập một trang web nào đó, hành vi đó là hành vi độc hại. Người dùng có thể xem toàn bộ thông tin request gửi đi để có quyết định chính xác nhất.

## 4.5. Thử nghiệm

Kết quả phân tích nhật ký mạng của nhiều tiện ích mở rộng khác nhau chưa ghi nhận hành vi độc hại nào. Tuy nhiên, hệ thống đã đưa ra cảnh báo cho người dùng trong vài trường hợp:

* Cảnh báo website bị chặn, tuy nhiên người dùng đang dùng một extension có tên là “Block site” để chặn trang web
* Cảnh báo tìm thấy email trong request gửi đi của một extension. Tuy nhiên người dùng đang sử dụng chức năng đăng nhập của extension “Speed test”

Từ kết quả trên trong 100 extension thử nghiệm, ta có thể thấy được tỉ lệ extension có hành vi mạng độc hại là rất thấp. Tuy nhiên người dùng vẫn phải cẩn thận và chú ý khi sử dụng extension cho trình duyệt web.

# CHƯƠNG 5: KẾT LUẬN

Đối mặt với vấn đề về an ninh của các môdun mở rộng cho trình duyệt hiện nay, việc tìm ra phương pháp phát hiện hành vi độc hại của các extension và việc xây dựng hệ thống phát hiện hành vi độc hại là rất quan trọng và thiệt thực. Nó giúp bảo vệ người dùng khỏi các extension có hành vi độc hại cũng như những kẻ lợi dụng tiện ích mở rộng với mục đích xấu.

Sau một quá trình dài tìm hiểu những lý thuyết liên quan đến các tiện ích mở rộng , cách chúng thực hiện các hành vi độc hại, hệ thống phát hiện hành vi độc hại của tiện ích mở rộng cho trình duyệt đã được xây dựng và đi vào hoạt động. Hệ thống này mang ý nghĩa thiết thực, góp phần bảo vệ người dùng khỏi các extension độc hại cũng như giúp họ kiểm soát chặt chẽ các hành vi mạng của chúng.

Mục dù khóa luận đã kết thúc tại đây nhưng những nghiên cứu về hành vi độc hại của extension vẫn sẽ được tôi tiếp tục thực hiện. Hệ thống phát hiện hành vi độc hại của tiện ích mở rộng cho trình duyệt vẫn sẽ được tôi phát triển và cải tiến. Trong thời gian tới, tôi mong muốn xây dựng một thuật toán tốt hơn nhằm xác định chính xác hành vi mạng độc hại và mở rộng ra toàn bộ các hành vi độc hại của extension.

**Tài liệu tham khảo**

**Tiếng Anh:**

[1] Michael Weissbacher , Enrico Mariconti, Guillermo Suarez-Tangil, Gianluca Stringhini, William Robertson, Engin Kirda, “Ex-Ray: Detection of History-Leaking Browser Extensions”,  *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2017

[2] Alexandros Kapravelos, University of California, Santa Barbara; Chris Grier, University of California, Berkeley, and International Computer Science Institute; Neha Chachra, University of California, San Diego; Christopher Kruegel and Giovanni Vigna, University of California, Santa Barbara; Vern Paxson, University of California, Berkeley, and International Computer Science Institute, “Hulk: Eliciting Malicious Behavior in Browser Extensions“, *USENIX Security Symposium*, 2014

[3] Louis F. DeKoven, Stefan Savage, and Geoffrey M. Voelker, UCSD; Nektarios Leontiadis, Facebook, “Malicious Browser Extensions at Scale: Bridging the Observability Gap between Web Site and Browser“, *USENIX Security Symposium*, 2017

[4] [Oleksii Starov](https://www.semanticscholar.org/author/Oleksii-Starov/2448867), [Nick Nikiforakis](https://www.semanticscholar.org/author/Nick-Nikiforakis/1679711), “Extended Tracking Powers: Measuring the Privacy Diffusion Enabled by Browser Extensions“, *WWW*, 2017

[5]  [Cortesi](https://twitter.com/cortesi), [Maximilianhils](https://twitter.com/maximilianhils), và [Raumfresser](https://twitter.com/raumfresser), *project Mimtproxy*

[6] Google, *Google Safe Browsing*

[7] Google, *Learn Extension Basic*