**FINGERPRINTING MOBILE DEVICES**

Như đã trình bày trong phần trước, các kỹ thuật lấy dấu vân tay hiện tại thiếu độ chính xác cho thiết bị di động. Bây giờ chúng tôi đề xuất một bộ tính năng đặc biệt áp dụng cho thiết bị di động vân tay. Bộ tính năng này bao gồm các thuộc tính và thuộc tính đã được tổng hợp bằng cách tạo môi trường cho trình duyệt bằng JavaScript. Chúng tôi mong muốn nghiên cứu tính hiệu quả của bộ tính năng dành cho thiết bị di động, ngay cả khi không phải tất cả các tính năng này đều có sẵn. Chúng tôi chia các đặc điểm của một thiết bị di động thành bốn loại và thảo luận về từng loại sau.

1. **Thuộc tính trình duyệt**

Các ứng dụng trình duyệt đã cung cấp nhiều thông tin khác nhau đối với môi trường của hệ thống. Chúng tôi đã phát hiện ra rằng các trình duyệt web di động phổ biến của Android Trình duyệt Android Android, Google Chrome, Firefox, IE mobile, Opera, Opera Mini và Safari EZ tiết lộ thông tin về phiên bản trình duyệt, HĐH và công cụ kết xuất bên dưới. Hơn nữa, trình duyệt gốc Android, Chrome (hai trình duyệt được sử dụng thường xuyên nhất trên thiết bị Android [4]) và Safari cũng cung cấp cho nhà sản xuất thiết bị, kiểu máy và ngôn ngữ của trình duyệt. IE mobile và Opera cũng cho phép phát hiện nhà sản xuất thiết bị và kiểu máy.

1. **Thuộc tính hệ thống**

Do tiêu chuẩn hóa cao của trình duyệt di động, không thể chỉ phân biệt các thiết bị trên cơ sở thông tin trình duyệt. Vì lý do này, chúng tôi mong muốn thu thập thêm thông tin về chính hệ thống thiết bị. Tuy nhiên, chúng tôi vẫn sử dụng trình duyệt để lấy thông tin và do đó chúng tôi phải chịu một số hạn chế nhất định do hộp cát và quyền hạn chế. Hơn nữa, chúng tôi muốn sử dụng dấu vân tay có độ ồn thấp, tức là, chúng tôi không cài đặt bất kỳ ứng dụng nào hoặc thực hiện bất kỳ hoạt động nào gây ra sự nghi ngờ của người dùng. Với những hạn chế này, chúng tôi có thể có được thông tin trên toàn hệ thống sau đây

Từ đối tượng điều hướng, chiều rộng và chiều cao màn hình và hiển thị colordepth hiển thị được trích xuất. Ngoài ra, tên và phiên bản HĐH được cung cấp bởi bộ điều hướng rất hữu ích cho các mục đích của chúng tôi. Hầu hết các phiên bản hiện tại của trình duyệt phổ biến cũng mang lại thông tin về loại kết nối hiện tại. Chúng tôi có được thông tin khi thiết bị nằm trong mạng WiFi hoặc sử dụng kết nối di động như 3G hoặc 4G.

Bên cạnh loại kết nối, chúng tôi thu thập thông tin về môi trường của thiết bị di động. Cụ thể hơn, chúng tôi có được múi giờ của Thiết bị bằng cách tính toán thời gian bù cho 13 điểm thời gian khác nhau và xây dựng hàm băm của sự khác biệt. Chúng tôi cũng lưu trữ địa chỉ IP IP của thiết bị và tên máy chủ của nút mạng, ví dụ: bộ định tuyến WiFi. Để có cái nhìn tổng quát hơn, tên máy chủ được che dấu bằng ký tự đại diện có thể được sử dụng làm tính năng bổ sung. Tên máy chủ thường trông giống như ip-xxx-xxx-xxx-xxx.web.provider.com, bao gồm địa chỉ IP của thiết bị và tên miền của nhà cung cấp mạng (phụ). Ký tự đại diện tên máy chủ trong ví dụ này sẽ là \* .web. Carrier.com, cho phép nhóm các thiết bị dựa trên mạng mà họ đã đăng nhập. Chúng tôi sử dụng MaxMind GeoIP2 [17] để xác định thông tin địa lý về vị trí hiện tại của thiết bị.

Chúng tôi cũng đã triển khai một máy dò Apple AirPlay. Máy thu AirPlay lắng nghe các thiết bị mạng cục bộ để truyền phát nội dung phương tiện. Chúng tôi đã triển khai một chức năng yêu cầu truyền phát tệp âm thanh nếu thiết bị di động được kết nối với WiFi và đã được xác định là đang chạy iOS. Điều này làm cho giao thức AirPlay trả về danh sách các thiết bị khả dụng có thể phát tệp. Sau khi nhận được danh sách này, chúng tôi hủy bỏ và rút yêu cầu phát trực tuyến. Danh sách các thiết bị mạng được kích hoạt AirPlay có thể cung cấp thông tin về môi trường (ví dụ: nếu người dùng sở hữu AppleTV).

Các thuộc tính cụ thể của hệ thống như các tiện ích hoạt động, mã hóa điện thoại được bật / tắt hoặc tùy chọn nhà phát triển không thể truy cập được thông qua bất kỳ trình duyệt web nào. Chắc chắn, có thể kiểm tra các tùy chọn này khi chạy một ứng dụng như Trình theo dõi quảng cáo. Tuy nhiên, trong kịch bản của chúng tôi, chúng tôi bị hạn chế đối với các kỹ thuật trình duyệt. Ngoài ra, chúng tôi đã xem xét việc đo CPU và bộ nhớ của thiết bị cũng như vị trí dựa trên mạng và GPS. Chúng tôi cũng đã phát triển một trình quét mạng dựa trên JavaScript để xác định địa chỉ IP cục bộ của thiết bị bên cạnh các thiết bị mạng khác như bộ định tuyến. Tuy nhiên, vì không muốn khơi dậy sự nghi ngờ, chúng tôi quyết định bỏ qua các thuộc tính như vậy. Quét mạng cục bộ hoặc kiểm tra CPU và bộ nhớ sẽ dẫn đến mất tải và hiệu suất cao và việc xác định vị trí dựa trên GPS thường dẫn đến một cửa sổ bật lên yêu cầu người dùng cho phép truy cập.

1. **Thuộc tính phần cứng**

Các hạn chế nói trên về quyền của trình duyệt dẫn đến thực tế là hầu như không có bất kỳ dữ liệu meta phần cứng nào có thể được truy cập. Do đó, chúng tôi không có khả năng nhận được số nhận dạng như số sê-ri của các thành phần phần cứng cụ thể, ví dụ: mô-đun máy ảnh. Tuy nhiên, chúng tôi tổng hợp ba thuộc tính sau trong ngữ cảnh trình duyệt: nền tảng thiết bị, số lượng điểm tiếp xúc của thiết bị và tính khả dụng của động cơ rung.

Ngoài ra, chúng ta có thể truy cập một con quay hồi chuyển và gia tốc kế của thiết bị qua JavaScript, thường được sử dụng trong các trò chơi dựa trên trình duyệt. Công việc trước đây đã chỉ ra rằng các cảm biến này có sự không hoàn hảo khác nhau giữa các thiết bị khác nhau [10]. Để xác định những điểm không hoàn hảo này, chúng tôi đã triển khai một chức năng thu thập dữ liệu gia tốc kế và con quay hồi chuyển và sử dụng dữ liệu đó như một tính năng mô tả khác. Xin lưu ý rằng chúng tôi không có thông tin về các hoạt động hiện tại của người dùng và thiết bị có thể đang di chuyển trong khi thu thập dữ liệu này. Để tránh biến dạng, chúng tôi lọc các chuyển động này dựa trên biên độ gia tốc. Thông tin phần cứng khác như tính khả dụng của khe cắm thẻ SIM thứ hai hoặc thông số kỹ thuật của cảm biến không có sẵn cho trình duyệt

1. **Thuộc tính hành vi**

Chúng tôi cũng đã triển khai ba chức năng để thu thập thông tin chi tiết hơn về người dùng thiết bị và hành vi của cô ấy. Đầu tiên, khi chúng tôi muốn tìm hiểu về thói quen duyệt web của người dùng, chúng tôi đã thực hiện một kỹ thuật tấn công thời gian để đánh cắp lịch sử [22]. Kết xuất của các siêu liên kết được truy cập khác với kết xuất của các liên kết không mong muốn trong các trình duyệt khác nhau. Thực tế này được sử dụng để xác định xem người dùng có truy cập các trang web cụ thể hay không bằng cách đo thời gian kết xuất của các liên kết cụ thể bằng cách sử dụng hàm JavaScript requestAnimationFrame. Trong các thử nghiệm của chúng tôi, chúng tôi đã quyết định kiểm tra xem một người dùng đã truy cập các trang web của Amazon, Ebay, Facebook, Google, Twitter và Zalando, mỗi người có các tên miền cấp cao khác nhau để nhận thông tin về bản địa hóa người dùng. Chúng tôi đã chọn các trang web này bởi vì chúng tôi có thể hy vọng chúng có cơ sở người dùng lớn và do đó chúng tôi có thể thấy cơ hội công bằng cho người dùng Internet ngẫu nhiên đăng nhập tại một hoặc nhiều trong số họ. Như một hạn chế của tính năng này, mọi trang web được xác định là không bị chặn sau khi người dùng xóa lịch sử trình duyệt của mình.

Thứ hai, chúng tôi truy vấn các trang web phổ biến từ trình duyệt của người dùng cho các đối tượng chỉ có thể truy cập đối với người dùng đã đăng nhập, ví dụ: một hình ảnh cụ thể. Chính xác hơn, một URL được chuẩn bị để người dùng đăng nhập được chuyển hướng đến một nội dung cụ thể, trong khi màn hình đăng nhập của trang web sẽ hiển thị cho người dùng không đăng nhập. URL này được gọi trong nền. Nếu nó được tải chính xác, chúng tôi có thể giả sử rằng người dùng đã đăng nhập, trong khi cô ấy không phải là nếu xảy ra lỗi. Phương pháp này có thể được áp dụng cho một số trang web phổ biến, mặc dù các URL được xây dựng hơi khác nhau cho mỗi trang web. Ngoài ra, chúng tôi tải một hình ảnh có thể truy cập công khai để phát hiện trình duyệt dựa trên văn bản. Do đó, nếu người dùng vô hiệu hóa tải hình ảnh hoàn toàn, chúng tôi không phân loại cô ấy giống như người dùng chưa đăng nhập vào bất kỳ trang web nào được thử nghiệm.

Thứ ba, chúng tôi đã triển khai một chức năng để đo tốc độ gõ của người dùng. Do đó, một trường văn bản (ví dụ: CAPTCHA) được đặt trên trang web, sau đó có thể được theo dõi cho đầu vào của người dùng. Khi người dùng bắt đầu nhập, bộ hẹn giờ được kích hoạt dừng sau khi người dùng không nhấn phím trên bàn phím trong một thời gian nhất định. Số lượng chữ cái trung bình mỗi giây sau đó được tính toán và sử dụng làm thuộc tính cho dấu vân tay hành vi của người dùng. Vì tốc độ gõ có thể khác nhau đối với ngay cả một người dùng, điều này không có nghĩa là một định danh duy nhất cho một người. Tuy nhiên, kết hợp với các tính năng khác, tốc độ gõ có thể cải thiện phân loại của chúng tôi.

1. **Các tiếp cận chính thức**

Tóm lại, tính năng của chúng tôi được thiết lập cho thiết bị di động lấy dấu vân tay bao gồm một số thuộc tính của trình duyệt, hệ thống, phần cứng và một số lượng nhỏ hành vi của người dùng. Mục đích của chúng tôi là phát triển một hệ thống, dựa trên các tính năng được mô tả trước đây, có thể thực hiện hai thao tác:

Nhận biết các thiết bị mới, tức là các thiết bị chưa từng truy cập dịch vụ của chúng tôi trước đây.

If a device is not new, recognizing and associating it to a device that has already visited our service.

K: tâp hợp các thiết bị đã biết

Fi set các lần đăng nhập của thiết bị thứ I trong K

Ai = fi1, fi2, .. fin: n là số lần đăng nhập của thiết bị thứ i

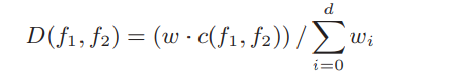
Fij = fij1, fij2, … fijd: d là số thuộc tính

Mỗi thiết bị Ki được liên kết với Fi

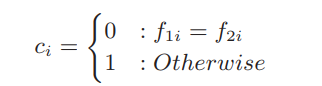
Với mõi thiết bị đăng nhập ku, sử dung Knn xác định vector feature gần giống nhất

1. **Đánh giá**
2. **Feature Distribution**
3. **Xác nhận thiết bị**

Trong phần này, chúng tôi mô tả việc triển khai hệ thống để nhận biết các thiết bị di động từ các tính năng được mô tả trong Phần 3. Chúng tôi cũng trình bày các thử nghiệm được thực hiện để đánh giá hiệu suất của hệ thống và bộ dữ liệu.







Vì tất cả các tính năng trong F1 và f2 được mã hóa dưới dạng số, chúng chỉ đóng góp vào khoảng cách nếu chúng có các giá trị khác nhau. Như được mô tả trong Phần 3.5, hệ thống xác định fmin vector tính năng với khoảng cách D thấp nhất từ fu. Nếu D (fu, fmin) <, các thiết bị được mô tả bởi fu và fmin sẽ được khớp. Mặt khác, fu fu tính năng đầu vào sẽ được liên kết với một thiết bị mới và được thêm vào cơ sở dữ liệu hệ thống.

Cài đặt thử nghiệm. Chúng tôi đã thực hiện một thử nghiệm trong đó hệ thống của chúng tôi được thiết kế để phát hiện các thiết bị không xác định và để khớp cùng lúc các thiết bị đã biết với các thiết bị chính xác. Chúng tôi chạy thử nghiệm như vậy trong hai tình huống có thể xảy ra:

1. Chế độ lặp đơn: Trong trường hợp này, chúng tôi cho rằng trang web đã được truy cập bởi một số thiết bị. Mục tiêu là nhận ra nếu các thiết bị mới đã truy cập hệ thống mà không cập nhật danh sách các thiết bị đã biết. Điều này được thực hiện để xác minh có bao nhiêu thiết bị mới có thể được hệ thống phát hiện chính xác trong một lần lặp. Đồng thời, hệ thống cũng phải có khả năng nhận ra nhiều truy cập của cùng một thiết bị\
2. Chế độ nhiều lần lặp: Trong kịch bản này, các lượt truy cập của từng thiết bị được xem xét lần lượt từng thiết bị và chúng được mô phỏng tại các thời điểm khác nhau. Sau mỗi lần lặp, danh sách các thiết bị đã biết được cập nhật bằng cách thêm các tính năng liên quan đến lượt truy cập mới. Quy trình này tái tạo hoàn toàn thuật toán mà chúng tôi đã mô tả trong phần trước.
3. **Chế độ lặp đơn**
4. **Chế độ lặp nhiều lần**

**Feature:**

