HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

VONGSAVANH VANPHATH

NGHIÊN CỨU PHƯƠNG PHÁP PHÁT HIỆN THAY ĐỔI NỘI DUNG BẢNG KẾT QUẢ CỦA TRANG TIN XỔ SỐ KIẾN THIẾT

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – NĂM 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

VONGSAVANH VANPHATH

NGHIÊN CỨU PHƯƠNG PHÁP PHÁT HIỆN THAY ĐỔI NỘI DUNG BẢNG KẾT QUẢ CỦA TRANG TIN XỔ SỐ KIẾN THIẾT

Chuyên ngành : HỆ THỐNG THÔNG TIN
Mã số : 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

Người hướng dẫn khoa học: PGS.TSKH. HOÀNG ĐĂNG HẢI

HÀ NỘI – NĂM 2020

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và dưới sự hướng dẫn của PGS.TSKH. Hoàng Đăng Hải. Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả luận văn

VONGSAVANH VANPHATH

LỜI CẢM ƠN

Học viên xin chân thành cảm ơn các thầy cô trong Khoa Đào tạo Sau Đại học, Học viện Công nghệ Bưu chính Viễn thông đã tạo điều kiện thuận lợi cho học viên trong quá trình học tập và nghiên cứu. Học viên xin chân thành cảm ơn PGS.TSKH. Hoàng Đăng Hải là người đã trực tiếp tận tình hướng dẫn học viên hoàn thành luận văn này.

Học viên xin chân thành cảm ơn các bạn bè đã sát cánh giúp học viên có được những kết quả như ngày hôm nay.

Đề tài nghiên cứu của luận văn có nội dung bao phủ rộng. Tuy nhiên, thời gian nghiên cứu còn hạn hẹp. Vì vậy, luận văn có thể có những thiếu sót. Học viên rất mong nhận được sự đóng góp ý kiến của các thầy cô và các bạn.

Xin chân thành cảm ơn!

Tác giả luận văn

VONGSAVANH VANPHATH

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT

STT	Từ viết tắt	Tiếng Anh	Tiếng Việt
1	API	Application Programming Interface	Giao diện lập trình ứng dụng
2	ATTT	Information security	An toàn thông tin
3	CNTT&TT	Information and Communication Technology	Công nghệ thông tin và truyền thông
4	CSRF	Cross Site Request Forgery	Kĩ thuật tấn công bằng cách sử dụng quyền chứng thực của người sử dụng đối với 1 website khác
	DHTML	Dynamic Hypertext Markup Language	Ngôn ngữ đánh dấu Siêu văn bản động
5	DOM	Document Object Model	Mô hình các đối tượng trong tài liệu HTML
6	FTP	File Transfer Protocol	Giao thức chuyển đổi file qua lại giữa trình duyệt và web server.
7	HTML	Hypertext Markup Language	Ngôn ngữ đánh dấu Siêu văn bản
8	HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
9	ID	Identification	Nhận dạng, nhận diện hoặc nhận biết
10	IP	Internet Protocol	Giao thức Internet
11	LCS	Longest common subsequence	Là việc thực hiện các thuật toán chia để trị để giải quyết bài toán tìm chuỗi con chung dài nhất.
12	LDAP	Lightweight Directory Access Protocol	Là một giao thức phát triển trên chuẩn X500, là một chuẩn cho dịch vụ thư mục chạy trên nền tảng OSI

13	MD5	Message-Digest algorithm 5	Thuật toán mã hóa theo chuẩn RFC 1321 để tạo ra 1 chuỗi 128 bit từ 1 chuỗi dữ liệu bất kỳ.
14	MIME	Multipurpose Internet Mail Extensions	Là một tiêu chuẩn Internet về định dạng cho thư điện tử
15	MITM	Man-in-the-middle attack	Tấn công xen giữa
16	OS	Operating System	Hệ điều hành
17	PDA	Persional Digital Assistant	Thiết bị trợ giúp kỹ thuật số cá nhân
18	SEO	Search Engine Optimization	Tối ưu hóa công cụ tìm kiếm
19	SHA-1	Secure Hash Algorithm-1	Thuật giải băm an toàn, trả lại kết quả dài 160 bit
20	SMS	Shortest Middle Snake	Phương pháp tìm đường ngắn nhất.
21	SQL	Structured Query Language	Ngôn ngữ truy vấn dữ liệu
22	SSO	Single sign-on	Đăng nhập một lần
23	Trang TTĐT	Portal	Trang thông tin điện tử
24	URL	Uniform Resource Locator	Đường dẫn hay địa chỉ dùng để tham chiếu đến các tài nguyên trên mạng Internet
25	WML	Wireless Markup Language	Ngôn ngữ đánh dấu kế thừa từ HTML, tuy nhiên WML dựa trên XML, do đó nó chặt chẽ hơn HTML.
26	XHTML	Extensible HyperText Markup Language	Ngôn ngữ Đánh dấu Siêu văn bản Mở rộng
27	XSS	Cross- Site-Scripting	Là kỹ thuật tấn công code injection ngay trên phía client

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

MỞ ĐẦU

Trong những năm gần đây, công nghệ thông tin và truyền thông có vai trò lớn đối với sự phát triển của mỗi quốc gia, mỗi doanh nghiệp. Ứng dụng CNTT&TT cũng có tác động không nhỏ đến đời sống kinh tế, xã hội của đại bộ phận người dân trên thế giới. CNTT&TT cũng góp phần quan trọng trong vấn đề an ninh và phát triển bền vững của mỗi quốc gia. Do vậy, ứng dụng CNTT&TT trở thành một phần không thể thiếu trong chiến lược phát triển của các doanh nghiệp và các quốc gia trên thế giới.

Với tốc độ phát triển và ứng dụng của CNTT&TT ngày càng nhanh như hiện nay, hàng ngày có một lượng lớn thông tin được lưu trữ, truyền tải thông qua các trang thông tin điện tử (TTĐT) cũng kéo theo nhiều rủi ro về sự mất an toàn thông tin. Thiệt hại do mất an ninh an toàn trên các trang TTĐT đã tăng rất nhanh và sẽ ảnh hưởng nghiêm trọng đến sự phát triển kinh tế- xã hội, nếu công tác đảm bảo an ninh an toàn không được triển khai đúng mức. Bởi các kỹ thuật của tội phạm mạng ngày càng cao và tinh vi hơn, số lượng điểm yếu an ninh ngày càng tăng, số vụ xâm phạm an toàn mạng ngày càng nhiều.

Trước những nguy cơ tấn công mạng ngày càng gia tăng vào các trang TTĐT, việc bảo đảm an toàn cho trang TTĐT là hết sức cần thiết. Một nguy cơ có thể xảy ra là nội dung thông tin trên trang có thể bị tin tặc tấn công, giả mạo bằng cách thay đổi thông tin. Ví dụ giả mạo kết quả trên trang tin kết quả xổ số có thể gây ra những tác hại rất lớn.

Do vậy, việc nghiên cứu phương pháp phát hiện thay đổi nội dung trang thông tin điện tử, cụ thể là cho một trang TTĐT về kết quả xổ số là hết sức cần thiết. Đó cũng là lý do học viên xin chọn đề tài: "**Nghiên cứu phương pháp phát hiện thay đổi nội dung bảng kết quả của trang tin xổ số kiến thiết"** làm đề tài cho luận văn nghiên cứu của mình.

Luận văn bao gồm 3 chương, bố cục các chương và các mục đi kèm như sau:

Chương 1: Tổng quan về vấn đề nghiên cứu

Khái niệm an toàn thông tin nhằm mục đích chính bảo vệ các khía cạnh tính bí mật, toàn vẹn và sẵn sàng của thông tin. Trong đó tính toàn vẹn chính là khía cạnh mà luận văn này muốn nghiên cứu, để xác định các nguy cơ thay đổi, giả mạo nội dung trang TTĐT. Các nội dung dự kiến:

- Vấn đề an toàn thông tin: cần nêu các mối nguy cơ, tác động đến trang thông tin điện tử nói chung.
- Vấn đề bảo đảm an toàn trang TTĐT nói chung.
- Nguy cơ thay đổi, giả mạo nội dung trang TTĐT nói chung. Tác hại.
- Trình bày về phương pháp thu thập thông tin từ trang TTĐT. Phương pháp chọn lọc thẻ tin, nội dung thông tin cần kiểm tra phát hiện thay đổi.
- Nghiên cứu về các mô hình, phương pháp, kỹ thuật liên quan đến thu thập thông tin, trích chọn dữ liệu, kiểm tra phát hiện thay đổi nội dung bảng kết quả của trang tin xổ số.

Chương 2: Nghiên cứu phương pháp kiểm tra phát hiện thay đổi nội dung trang tin xổ số

Đảm bảo tính toàn vẹn của thông tin, tức là thông tin chỉ được phép xóa hoặc sửa đổi bởi những đối tượng được phép và phải đảm bảo rằng thông tin vẫn còn chính xác khi được lưu trữ hay truyền đi. Ví dụ trường hợp tính toàn vẹn của thông tin bị phá vỡ: thay đổi kết quả xổ số trên trang xổ số kiến thiết từ một đối tượng không được phép dẫn đến nhiều hệ lụy. Chương này trình bày cụ thể về kiến trúc, cơ chế hoạt động của trang TTĐT cùng với mô hình tổng quát cho phương pháp kiểm tra phát hiện giả mạo nội dung trang tin. Bằng cách trình bày cụ thể phương pháp thu thập thông tin, chọn lọc nội dung thông tin cần kiểm tra đối với trang TTĐT, phân tích các công cụ thu thập thông tin sẽ đưa ra phương pháp kiểm tra phát hiện giả mạo nội dung trang kết quả xổ số. Các nội dung dự kiến:

- Khái quát về kiến trúc chung, cơ chế hoạt động của các trang TTĐT.
- Mô hình tổng quát cho phương pháp kiểm tra phát hiện thay đổi nội dung bảng kết quả của trang tin xổ số.
- Phương pháp thu thập thông tin từ trang TTĐT. Cụ thể cho một trang TTĐT về kết quả xổ số.
- Cách thức lập bảng dữ liệu về kết quả trang tin xổ số.
- Phân tích, đánh giá một số công cụ thu thập thông tin. Chọn một công cụ thu thập thông tin (dự kiến dùng bộ công cụ Scrapy).
- Phương pháp kiểm tra phát hiện thay đổi bảng kết quả của trang tin xổ số.
- Đánh giá, nhận xét

Chương 3: Cài đặt và thử nghiệm

Chương này nhằm hiện thực hóa các kết quả đã nghiên cứu, sẽ tiến hành triển khai thử nghiệm thu thập nội dung thông tin, ghi thông tin, kiểm tra phát hiện thay đổi nội dung trang tin kết quả xổ số. Các nội dung dự kiến:

- Xây dựng một kịch bản thử nghiệm.
- Kết quả thử nghiệm thu thập nội dung thông tin, ghi thông tin, kiểm tra phát hiện thay đổi nội dung trang tin kết quả xổ số.
- Phân tích, đánh giá kết quả thử nghiệm.

CHƯƠNG 1. TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU.

Khái niệm an toàn thông tin nhằm mục đích chính bảo vệ các khía cạnh tính bí mật, toàn vẹn và sẵn sàng của thông tin. Trong đó tính toàn vẹn chính là khía cạnh mà luận văn này muốn nghiên cứu, để xác định các nguy cơ thay đổi, giả mạo nội dung trang TTĐT.

1.1. Vấn đề an toàn thông tin: cần nêu các mối nguy cơ, tác động đến trang thông tin điện tử nói chung

Thông tin phản ánh các thuộc tính của đối tượng vật chất (tin tức về người, đồ vật, sự kiện, biến cố, hiện tượng và quá trình) và quan hệ giữa chúng nên có thể xem thông tin là đối tượng nhận thức và cũng là đối tượng cần bảo vệ. Thông tin được bảo vệ gồm thông tin mật và thông tin "nhạy cảm"...

Các nguy cơ đe dọa an toàn thông tin:

- An toàn thông tin được đánh giá bằng hai chỉ số: xác suất ngăn chặn các nguy cơ và thời gian đảm bảo mức độ an toàn xác định. Các chỉ số này phụ thuộc lẫn nhau. Trong những biện pháp bảo vệ thông tin cụ thể cho trước, có thể đảm bảo mức độ an toàn cao hơn trong khoảng thời gian ngắn hơn.
- Vì thông tin được chứa trong các tham số thông tin của vật mang, nên để đảm bảo an toàn thông tin, các tham số này phải giữ được giá trị của nó trong khoảng thời gian nhất định. Do tác động của những nhân tố (nhiễu) khác nhau mà giá trị tham số của vật mang tin cũng khác nhau.
- Thông tin thường bị đe dọa lấy cắp, thay đổi hay bị xóa một cách vô tình hay cố ý. Các nguy cơ này thường được thể hiện dưới dạng:
    - Những hoạt động của kẻ ác ý (khai thác thông tin vì mục đích tình báo quốc gia, tình báo thương mại, vì mục đích của những phần tử tội phạm, của những nhân viên không trung thành...);
    - Theo dõi các nguồn tin, nghe trộm các cuộc nói chuyện riêng và thu trộm các tín hiệu âm thanh của máy móc đang làm việc;
    - Chặn bắt các trường điện, trường từ, trường điện từ, các tín hiệu điện và các bức xạ phóng xạ;
    - Truyền trái phép vật mang tin dưới dạng vật chất ra ngoài cơ quan, đơn vị;
    - Những người nắm giữ thông tin riêng hoặc thông tin mật tiết lộ thông tin;

- Đánh mất vật mang tin (tài liệu, vật mang tin dưới dạng máy móc, mẫu nguyên vật liệu...);
- Lan truyền trái phép thông tin qua các trường và các tín hiệu điện xuất hiện ngẫu nhiên ở các thiết bị điện và thiết bị vô tuyến điện, vì các thiết bị đó quá cũ hoặc sản xuất kém chất lượng và vi phạm các nguyên tắc sử dụng;
- Những hỏng hóc do thiết bị, không thể sửa được trong quá trình làm việc của thu thập, xử lý, lưu giữ và truyền tin, những lỗi do vô ý của người dùng tin.
- Ảnh hưởng của thiên tai, sự cố về an toàn, ảnh hưởng của các loại nhiễu tự nhiên, nhiễu điện công nghiệp, nhiễu điện từ.

- Để bảo vệ thông tin có hiệu quả, cần ước lượng giá trị của nguy cơ đe dọa an toàn thông tin. Giá trị của một nguy cơ cụ thể đối với thành phần thông tin xem xét đầu tiên trong mọi trường hợp có thể biểu thị dưới dạng tích của các thiệt hại tiềm ẩn do thực trạng nguy cơ về yếu tố thông tin đầu tiên với xác suất thực tế thể hiện nó.
- Việc nhận giá trị định lượng tương đối chính xác và khách quan của các thành phần là phức tạp. Việc đánh giá gần đúng độ lớn của nguy cơ đe dọa an toàn thông tin có thể thực hiện được trong những điều kiện và giới hạn sau:
  - Thứ nhất, có thể giả thiết thiệt hại lớn nhất do thông tin bị đánh cắp tương ứng với giá trị của thông tin đó. Thực tế, trong trường hợp thông tin rơi vào tay đối thủ cạnh tranh thì người sở hữu thông tin có thể không những mất lợi nhuận được hưởng mà còn không thể bù được giá thành sản phẩm.
  - Thứ hai, trong trường hợp hoàn toàn không xác định được ý đồ của kẻ ác ý về khai thác thông tin thì sai số dự đoán là nhỏ nhất nếu chấp nhận giá một điều rõ ràng là giá trị thông tin lớn bao nhiêu và nguy cơ đe dọa an toàn thông tin cao bao nhiêu thì các nguồn lực để bảo vệ thông tin càng phải lớn bấy nhiêu.

Từ những phân tích trên đây có thể thấy rằng, việc đánh giá một cách đầy đủ các nguy cơ về an toàn thông tin đối với nguồn tài nguyên thông tin của mỗi cơ quan, tổ chức là bước đi cần thiết để có thể xây dựng các chính sách, giải pháp bảo vệ thông tin một cách hữu hiệu

1.2. Các hình thức tấn công mạng phổ biến

1.2.1. Tấn công bằng phần mềm độc hại (Malware attack)

Tấn công malware là hình thức phổ biến nhất. Malware bao gồm spyware (phần mềm gián điệp), ransomware (mã độc tống tiền), virus và worm (phần mềm độc hại có khả năng lây lan nhanh). Thông thường, tin tặc sẽ tấn công người dùng thông qua các lỗ hổng bảo mật, cũng có thể là *dụ dỗ người dùng click vào một đường link hoặc email (phishing)* để phần mềm độc hại tự động cài đặt vào máy tính. Một khi được cài đặt thành công, malware sẽ gây ra:

- Ngăn cản người dùng truy cập vào một file hoặc folder quan trọng (ransomware)
- Cài đặt thêm những phần mềm độc hại khác
- Lén lút theo dõi người dùng và đánh cắp dữ liệu (spyware)
- Làm hư hại phần mềm, phần cứng, làm gián đoạn hệ thống.

1.2.2. Tấn công giả mạo (Phishing attack)

Phishing là hình thức giả mạo thành một đơn vị/cá nhân uy tín để chiếm lòng tin của người dùng, thông thường qua email. Mục đích của tấn công Phishing thường là đánh cắp dữ liệu nhạy cảm như thông tin thẻ tín dụng, mật khẩu, đôi khi phishing là một hình thức để lừa người dùng cài đặt malware vào thiết bị (khi đó, phishing là một công đoạn trong cuộc tấn công malware).

1.2.3. Tấn công trung gian (Man-in-the-middle attack)

Tấn công trung gian (MitM), hay **tấn công nghe lén**, xảy ra khi kẻ tấn công xâm nhập vào một giao dịch/sự giao tiếp giữa 2 đối tượng. Khi đã chen vào giữa thành công, chúng có thể đánh cắp dữ liệu của giao dịch đó.

Loại hình này xảy ra khi:

- Nạn nhân truy cập vào một mạng Wifi công cộng không an toàn, kẻ tấn công có thể "chen vào giữa" thiết bị của nạn nhân và mạng Wifi đó. Vô tình, những thông tin nạn nhân gửi đi sẽ rơi vào tay kẻ tấn công.
- Khi phần mềm độc hại được cài đặt thành công vào thiết bị, một kẻ tấn công có thể dễ dàng xem và điều chỉnh dữ liệu của nạn nhân.

1.2.4. Tấn công từ chối dịch vụ (DoS và DDoS)

DoS (Denial of Service) là hình thức tấn công mà tin tặc "đánh sập tạm thời" một hệ thống, máy chủ, hoặc mạng nội bộ. Để thực hiện được điều này, chúng

thường tạo ra một lượng traffic/request khổng lồ ở cùng một thời điểm, khiến cho hệ thống bị quá tải, từ đó người dùng không thể truy cập vào dịch vụ trong khoảng thời gian mà cuộc tấn công DoS diễn ra.

Một hình thức biến thể của DoS là DDoS (Distributed Denial of Service): tin tặc sử dụng một mạng lưới các máy tính (botnet) để tấn công nạn nhân. Điều nguy hiểm là chính các máy tính thuộc mạng lưới botnet cũng không biết bản thân đang bị lợi dụng để làm công cụ tấn công. Đọc thêm: Sự nguy hiểm của Tấn công DDoS

1.2.5. Tấn công cơ sở dữ liệu (SQL injection)

Tin tặc "tiêm" một đoạn code độc hại vào server sử dụng ngôn ngữ truy vấn có cấu trúc (SQL), mục đích là khiến máy chủ trả về những thông tin quan trọng mà lẽ ra không được tiết lộ. Các cuộc tấn công SQL injection xuất phát từ các lỗ hổng của website, đôi khi tin tặc có thể tấn công chỉ bằng cách chèn một đoạn mã độc vào thanh công cụ "Tìm kiếm" là đã có thể tấn công website.

1.2.6. Khai thác lỗ hổng Zero-day (Zero day attack)

Lỗ hổng Zero-day (0-day vulnerabilities) là các lỗ hổng bảo mật chưa được công bố, các nhà cung cấp phần mềm chưa biết tới, và dĩ nhiên, chưa có bản vá chính thức. Chính vì thế, việc khai thác những lỗ hổng "mới ra lò" này vô cùng nguy hiểm và khó lường, có thể gây hậu quả nặng nề lên người dùng và cho chính nhà phát hành sản phẩm.

1.2.7. Các loại khác

Ngoài ra, còn rất nhiều hình thức tấn công mạng khác như: Tấn công chuỗi cung ứng, Tấn công Email, Tấn công vào con người, Tấn công nội bộ tổ chức, v.v. Mỗi hình thức tấn công đều có những đặc tính riêng, và chúng ngày càng tiến hóa phức tạp, tinh vi đòi hỏi các cá nhân, tổ chức phải liên tục cảnh giác & cập nhật các công nghệ phòng chống mới.

1.3. Vấn đề bảo đảm an toàn trang TTĐT nói chung

Đối với các doanh nghiệp, các cơ quan, tổ chức cổng/trang TTĐT là kênh cung cấp thông tin hiệu quả và nhanh chóng nhất. Không chỉ dừng lại ở việc cung cấp thông tin, các cổng/trang TTĐT còn là kênh quảng bá, giao dịch thương mại và mua bán rất phổ biến hiện nay. Cũng chính đặc điểm này, các cổng/trang TTĐT thường xuyên là mục tiêu tấn công của những kẻ xấu hay tin tặc để khai thác đánh cắp các thông tin liên quan bên trong. Phương thức tấn công phổ biến là khai thác các lỗi bảo mật trên các cổng/trang TTĐT nói riêng và các ứng dụng web nói chung.

Tin tặc có thể sử dụng nhiều biện pháp khác nhau để dò tìm và khai thác các lỗi bảo mật của ứng dụng web để thực hiện các cuộc tấn công.

Trong cổng/trang TTĐT thường có các thành phần cho người dùng nhập dữ liệu vào như mục đăng nhập, tìm kiếm, bình luận, liên kết đến bài viết, v.v. Ngoài việc giúp cho người dùng dễ dàng tương tác với cổng/trang TTĐT, các mục này nếu không được kiểm soát chặt chẽ sẽ trở thành một nguy cơ lớn để tin tặc thực hiện các cuộc tấn công. Bởi vậy, trước khi đưa cổng/trang TTĐT vào hoạt động chính thức cần sử dụng các công cụ phần mềm để tìm và kiểm tra tất cả các lỗ hổng có thể bị kẻ xấu khai thác. Từ đó tìm cách khắc phục những lỗ hổng trên cổng/trang TTĐT của mình để đảm bảo an ninh an toàn. Nhiều công cụ có thể tìm và phát hiện các loại lỗ hổng bảo mật từ những lỗi phổ biến, đến những lỗi ít gặp. Có những công cụ không chỉ giúp dà soát lỗ hổng của mã nguồn cổng/trang TTĐT, mà còn tìm và phát hiện lỗ hổng bảo mật trong việc thiết lập cấu hình máy chủ.

Ngoài ra có thể sử dụng biểu thức chính quy áp dụng cho tất cả các ngôn ngữ lập trình để thực hiện các công việc này. Chẳng hạn như sử dụng biểu thức chính quy để lọc siêu ký tự, để quy định giá trị mật khẩu nhập vào (ví dụ mật khẩu phải 8 ký tự trở lên và bao gồm chữ số, chữ hoa, chữ thường, ký tự đặc biệt, v.v.). Cũng có thể dùng biểu thức chính quy để lọc các tấn công.

Sau khi đã xác định được các lỗi trên cổng/trang TTĐT của mình, cũng cần phân loại để đưa ra những giải pháp phòng chống thích hợp. Việc phân loại các lỗi và các kiểu tấn công thành các nhóm khác nhau sẽ giúp người quản trị dễ dàng xác định các nguy cơ cũng như biện pháp đối phó. Sau đây là một số lỗi phổ biến trên các cổng/trang TTĐT nói riêng và ứng dụng web nói chung, có thể bị khai thác để tấn công.

- **Các lỗi Injection**

Các lỗi Injection cho phép tin tặc thực hiện các kiểu tấn công như SQL Injection, OS Injection, LDAP Injection. Kiểu tấn công này xảy ra khi tin tặc gửi các dữ liệu gây hại đến ứng dụng web. Những dữ liệu này có tác dụng như các câu lệnh với hệ điều hành hoặc các câu truy vấn với cơ sở dữ liệu, có thể lừa hệ thống biên dịch đi vào thực hiện những mã lệnh độc hại hoặc giúp kẻ tấn công truy cập đến những dữ liệu quan trọng một cách trái phép. Một trong những dạng phổ biến nhất của lỗi injection là lỗi "SQLInjection". Lỗi này được thực thi bằng cách chèn các câu truy vấn SQL vào dữ liệu tương tác giữa máy khách và trình ứng dụng. Đây là lỗ hổng trong việc kiểm tra dữ liệu nhập trong các ứng dụng web và các thông

báo lỗi của hệ quản trị quản trị cơ sở dữ liệu. Tin tặc có thể lợi dụng lỗ hổng này để chèn vào và thi hành những câu lệnh SQL để khai thác lỗi. Quá trình khai thác lỗi SQL Injection thành công có thể giúp tin tặc lấy được các dữ liệu nhạy cảm trong cơ sở dữ liệu, thay đổi cơ sở dữ liệu (thêm, xóa, sửa), thực thi các hành động với quyền của người quản trị và cao hơn có thể điều khiển được hệ điều hành máy chủ.

- **Các lỗi Cross-Site-Scripting (XSS)**

Các lỗi Cross-Site-Scripting (XSS) xảy ra khi một ứng dụng web bị lợi dụng để gửi những dữ liệu độc hại đến trình duyệt của người sử dụng. Những lỗ hổng này rất phổ biến và xảy ra trong bất cứ phần nào của ứng dụng web có sử dụng dữ liệu từ người dùng mà không kiểm tra tính hợp lệ. Tin tặc tấn công bằng cách chèn vào các ứng dụng web động những thẻ HTML hay những mã Script nguy hiểm có thể gây hại cho những người sử dụng. Trong đó, những đoạn mã nguy hiểm đươc chèn vào hầu hết được viết bằng các Client-Site Script như JavaScript, JScript, DHTML và cũng có thể là cả các thẻ HTML. Khi một người sử dụng kích vào các liên kết, các tập tin flash trên các ứng dụng web hay được gửi bởi tin tặc thì những đoạn mã độc sẽ được thực thi trên chính trình duyệt của người dùng. Hậu quả của tấn công dạng XSS có thể rất nguy hiểm, người dùng có thể bị chiếm quyền điều khiển phiên làm việc của mình, bị lộ các thông tin (cookie, tên đăng nhập, mật khẩu,...), lộ các tập tin của cá nhân. Tin tặc có thể thực thi những hành vi gây hại khác như, mạo danh người dùng, cài đặt các mã độc trên máy người dùng, thay đổi nội dung trên các trang web hoặc chuyển hướng người dùng đến các trang web chứa mã độc hại khác,...

- **Các lỗi quản lý xác thực và quản lý phiên làm việc**

Các lỗi liên quan đến quá trình quản lý xác thực và quản lý phiên làm việc: bao gồm tất cả các yếu tố quản lý xác thực người dùng và các phiên truy cập. Xác thực người dùng là một yếu tố quan trọng trong quy trình này, nhưng ngay cả những cơ chế xác thực mạnh nhất vẫn có thể bị mắc những lỗi liên quan đến các chức năng quản lý xác thực, bao gồm thay đổi mật khẩu, quên mật khẩu, lưu nhớ mật khẩu ở trình duyệt, cập nhật tài khoản và những chức năng khác. Xác thực người dùng trên ứng dụng web thường sử dụng tên đăng nhập và mật khẩu. Những phương pháp xác thực mạnh hơn bao gồm các giải pháp phần cứng hoặc phần mềm dựa trên các token key hoặc dùng phương pháp sinh trắc học (nhận dạng vân tay, v.v.). Tuy nhiên những phương pháp này có phần hạn chế do giá thành cao. Một số lượng lớn lỗi trong các hàm quản lý tài khoản có thể dẫn đến mối nguy cơ lộ tài khoản người

dùng và thậm chí là tài khoản của người quản trị. Các ứng dụng web thường phải theo dõi và duy trì phiên truy cập của người dùng nhằm phân biệt các truy cập từ người dùng khác nhau. Giao thức HTTP không cung cấp khả năng này và do đó ứng dụng web phải tự tạo cơ chế này. Thường thì, môi trường phát triển ứng dụng cung cấp cơ chế quản lý phiên truy cập (thường là dưới hình thức cookie token), tuy nhiên đa số các nhà lập trình nghiêng về phát triển cơ chế riêng của họ. Trong cả hai trường hợp, nếu token quản lý phiên truy cập không được bảo vệ, tin tặc có thể ăn cắp token truy cập tài khoản người khác.

- **Các lỗi đối tượng tham chiếu không an toàn**

Các lỗi đối tượng tham chiếu không an toàn (Insecure Direct Object References): xảy ra khi người phát triển ứng dụng web để lộ một tham chiếu đến những đối tượng trong hệ thống như các tập tin, thư mục hay chìa khóa dữ liệu. Nếu chúng ta không có một hệ thống kiểm tra truy cập, kẻ xấu có thể lợi dụng những tham chiếu này để truy cập dữ liệu một cách trái phép. Việc phân quyền yếu cho phép người dùng có thể truy cập dữ liệu của người khác. Trong trường hợp tấn công tin, tặc có thể xác định được cấu trúc truy vấn gửi đến máy chủ và có thể nhanh chóng thu thập được dữ liệu như thẻ tín dụng, mã khách hàng, thông tin cá nhân của khách hàng. Nguyên nhân là do các ứng dụng web thường xuyên sử dụng tham chiếu trực tiếp tên hoặc khóa của một đối tượng khi xây dựng các ứng dụng web và không kiểm tra kỹ người dùng thật sự có quyền truy cập hay không. Đây chính là mấu chốt dẫn đến các lỗi đối tượng tham chiếu không an toàn.

- **Các lỗi cấu hình thiếu an toàn**

Các lỗi cấu hình thiếu an toàn: cấu hình máy chủ và các phần mềm hỗ trợ dịch vụ web là một yếu tố quan trọng trong vấn đề bảo mật của ứng dụng. Máy chủ cung cấp nền tảng phục vụ cho việc cung cấp nội dung và các gói dịch mà ứng dụng web cần sử dụng, như dịch vụ lưu trữ, thư điện tử. Những vấn đề về cấu hình của máy chủ có thể dẫn đến vấn đề bảo mật của ứng dụng. Hiện nay nhiều ứng dụng web được lưu trữ trên các máy chủ đi thuê hoặc các máy chủ được chia sẻ, những người phát triển ứng dụng web thường không nắm được nhiều kiến thức về cấu hình máy chủ, lại thiếu sự liên kết với bên hỗ trợ triển khai ứng dụng web trên máy chủ. Vì vậy, thiếu sự thống nhất và liên lạc về phương hướng bảo mật giữa hai nhóm. Điều này dẫn đến những điểm yếu nghiêm trọng được tạo ra trên ứng dụng từ các lỗ hổng ở cả ứng dụng web và máy chủ.

- **Các lỗi lưu trữ dữ liệu thiếu an toàn:**

Các lỗi lưu trữ dữ liệu thiếu an toàn (Sensitive Data Exposure): Các dữ liệu nhạy cảm được lưu trữ không an toàn ảnh hưởng đến hệ thống máy chủ cũng như khách hàng, chẳng hạn như thẻ tín dụng, mã số thuế và các thông tin xác thực. Tin tặc có thể lợi dụng những kẽ hở này để đánh cắp hay sửa đổi những dữ liệu được bảo vệ kém, để gian lận thẻ tín dụng, trộm cắp thông tin cá nhân, hoặc thực hiện các hành vi phạm tội khác. Các sai lầm phổ biến nhất gây ra lỗi này chỉ đơn giản là không mã hóa dữ liệu nhạy cảm mà lưu trữ những dữ liệu đó ở dạng văn bản thông thường, có thể là tạo khóa, sử dụng các thuật toán mã hóa yếu phổ biến, đặc biệt là các kỹ thuật băm mật mã yếu. Cũng có khi là do điểm yếu của trình duyệt, tin tặc có thể nghe lén trên kênh truyền HTTPS và giải mã dữ liệu thông qua lỗ hổng.

- **Các lỗi Cross Site Request Forgery (CSRF)**

CSRF (Cross Site Request Forgery): CSRF khai thác lỗi xử lý phiên làm việc của các trang web. Tin tặc có thể lợi dụng người dùng để thực thi những hành động không mong muốn ngay trên phiên đăng nhập của họ. Thông qua việc gửi người dùng một liên kết qua email hay chát, tin tặc có thể hướng người dùng thực thi một số hành động ngay trên trình duyệt của người dùng (như gửi bài viết, xóa bài viết v.v…) Một cuộc tấn công CSRF lừa nạn nhân gửi một yêu cầu HTTP giả mạo trên phiên đăng nhập của mình kèm theo những thông tin xác thực, mà ứng dụng nghĩ là các yêu cầu của nạn nhân.

- **Các lỗi do ứng dụng sử dụng những thành phần chứa lỗi bảo mật:**

Các ứng dụng được sử dụng có các thành phần bị lỗi có thể làm suy yếu khả năng phòng thủ của ứng dụng. Do đó có thể bị một loạt các cuộc tấn công và các tác động. Các thành phần như là các thư viện, các thành phần mở rộng và các bản vá lỗi của các thành phần khác hầu hết luôn chạy với quyền đầy đủ. Nếu một thành phần dễ bị tấn công được khai thác, thì cuộc tấn công như vậy có thể làm cho mất dữ liệu nghiêm trọng hay mất quyền kiểm soát máy chủ.

- **Các lỗi trong việc kiểm soát quyền truy cập**

Các lỗi trong việc kiểm soát quyền truy cập (Missing Function Level Access Control) đối với người dùng trên ứng dụng web. Hiện nay, hầu hết các ứng dụng web có chức năng xác thực quyền truy cập. Các chức năng được quản lý thông qua việc cấu hình và hệ thống có những khi được cấu hình sai, là nguyên nhân dẫn đến lỗi này. Vì vậy, các ứng dụng cần phải thực hiện việc kiểm tra kiểm soát truy cập đồng thời trên máy chủ khi mà mỗi chức năng được truy cập. Nếu các yêu cầu không được xác thực, những tin tặc có thể giả mạo yêu cầu để truy cập trái phép vào các

chức năng mà chúng không có quyền. Những tin tặc này có thể là một người được cấp quyền trong hệ thống.

Chỉ cần thay đổi các URL hoặc một tham số để được sử dụng chức năng đặc quyền, hay những người dùng ẩn danh cũng có thể truy cập các trang tin cá nhân không được bảo vệ. Những chức năng quản trị là mục tiêu chính trong kiểu tấn công này.

- **Một số hình thức tấn công nhằm vào SSO:** tấn công tràn bộ đệm, chiếm phiên làm việc (Session Hijacking), tấn công Man in the Midle (MITM).

1.4. Nguy cơ thay đổi, giả mạo nội dung trang TTĐT nói chung.

Tấn công Deface là tấn công thay đổi nội dung, hacker sẽ thông qua một điểm yếu nào đó để thay đổi nội dung trang TTĐT của nạn nhân. Mục đích của việc đột nhập mà không được xin phép này là :

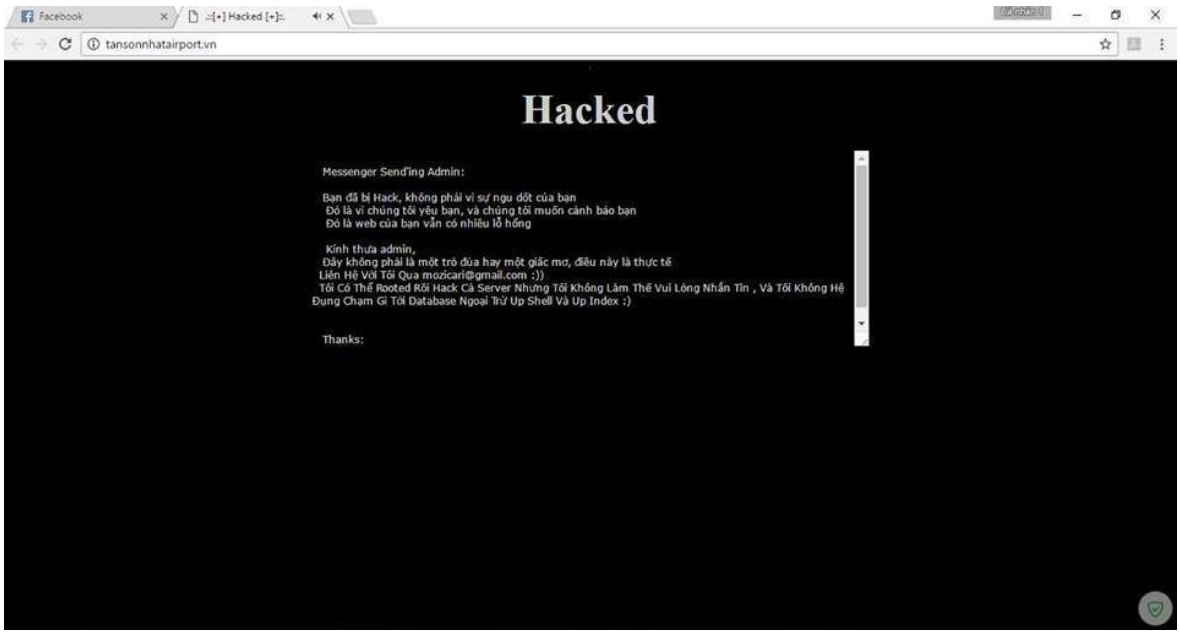
- Mục đích cao đẹp: cảnh báo quản trị viên biết trang TTĐT đang tồn tại lỗ hỏng bảo mật hoặc điểm yếu nghiêm trọng…
- Mục đích không đẹp: chứng tỏ năng lực bản thân, dạng này rất dễ gặp như kiểu hacked by…
- Mục đích xấu: thù hằn, nội dung thay đổi thường là lăng mạ nạn nhân hoặc nội dung liên quan đến chính trị, tôn giáo…

Có rất nhiều nguyên nhân trang TTĐT bị Deface, chủ yếu là do trang TTĐT tồn tại nhiều điểm yếu bảo mật nghiêm trọng mà hacker có thể upload file lên server hoặc có quyền đăng nhập vào trang quản trị trang TTĐT (Ví dụ : SQL Injection). Thậm chí nếu trang TTĐT trên hosting an toàn thuộc server bị tấn công thì cũng sẽ bị tấn công Deface luôn (Local Attack).

Các trường hợp trang TTĐT bị tấn công Deface: lỗi SQL injection, lỗi XSS (Cross Site Scripting), lỗ hổng Remote File Include, lỗ hổng Local file inclusion, không cập nhật phiên bản, mật khẩu quản trị yếu

Thông thường hacker tấn công Deface chủ yếu vào các trang mặc định như : index.php, index.html, home.html, default.html, trangchu.html… thì chỉ cần xử lý các trang mặc định này trang TTĐT sẽ hoạt động lại.

Nhưng nếu hacker không thay đổi nội dung những file trên thì khó thể phát hiện và sẽ nhận được cảnh báo từ việc truy cập trang TTĐT hoặc nhà quản lý Hosting.



Hình1.1: Màn hình một trang TTĐT bị tấn công

Cách khắc phục

- Luôn xem những thông tin nhật ký, file log của máy chủ và truy tìm xem, hacker đã làm gì và làm như thế nào trên hệ thống của mình.
- Ngoài ra, cũng có thể tham khảo 1 số biện pháp khuyến cáo:
  - Tiến hành scan shell, mã độc trên server khi sự cố xảy ra, xác định nguyên nhân, upload source phiên bản mới để khắc phục.
  - Thường xuyên kiểm tra dữ liệu trang TTĐT (để ý thời gian tập tin, thư mục bị thay đổi).
  - Có kế hoạch backup dữ liệu cụ thể hàng tuần để lúc cần có thể restore lại ngay.
  - Không nên cài đặt các module, plugin, extension,… không thật sự cần thiết và không rõ nguồn gốc (nên download module, plugin, extension,… từ các trang web uy tín).

Nên đổi mật khẩu quản trị theo 1 chu kỳ định sẵn 3 tháng/1 lần và lưu giữ cẩn thận.

1.5. Các mô hình, phương pháp, kỹ thuật liên quan đến thu thập thông tin, trích chọn dữ liệu.

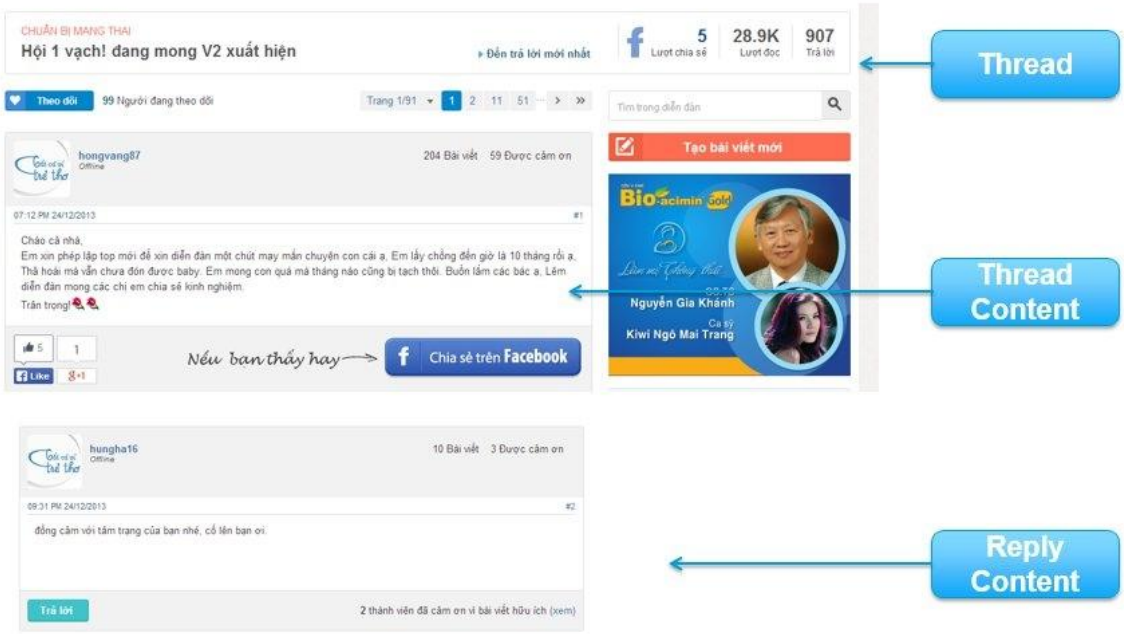
Hiện nay có 2 phương pháp chính dùng để thu thập dữ liệu: API và Trang (Sites).

- Thu thập dữ liệu bằng cổng giao thức lập trình (API): Phương pháp này được áp dụng đối với các global social networks như Facebook, Google Plus, Youtube, Twitter , Instagram… trong đó các công cụ social listening sẽ kết nối với các API (Application Programming Interface – Giao diện lập trình ứng dụng) của các social networks và yêu cầu hệ thống trả về những bài viết có chứa keywords. Phương pháp này theo nguyên tắc cho phép lấy dữ liệu của toàn bộ social network, bao gồm các trang cá nhân, nhưng trên thực tế phụ thuộc vào sự hạn chế của các social networks này. Với việc Facebook hạn chế organic reach cho các chủ fanpage và các nhà quảng cáo, Facebook cũng không trả lại đầy đủ và nhất quán các bài viết cá nhân cho Social Listening tool qua API. Hiện tại không có một thống kê rõ ràng việc lấy dữ liệu bằng API có thể lấy được bao nhiêu % thảo luận.
- Thu thập dữ liệu theo Sites: Hệ thống sẽ đi thu thập dữ liệu từ các nguồn dữ liệu theo trang như trang TTĐT báo điện tử, forum, Facebook fanpages, Youtube channels, Instagram pages,…. Phương pháp này giúp thu thập toàn bộ dữ liệu của các kênh được liệt kê. Việc thu thập dữ liệu được thực hiện bằng 2 cách: Tự động lan tỏa và Liệt kê danh sách trang (panel).
  - Thu thập theo phương pháp liệt kê trang:

Công đoạn xây dựng một social listening platform cho thị trường mới bắt đầu từ việc xây dựng một tổ hợp các trang fanpages mạng xã hội, báo điện tử, diễn đàn, blogs… của thị trường đó. Việc này thường tốn từ 6 tháng đến một năm. Từ danh sách các trang này, đội ngũ data team sẽ viết các con nhện (crawlers) để quét qua các trang liên tục và copy thảo luận người dùng về. Crawlers hành xử như một con người, tự động scan nội dung của trang, nhận điện đâu là bài viết (thread), nội dung của bài viết bao gồm bài viết đầu tiên (lead), tác giả (author), ngày giờ và các bình luận hay phản hồi (comments).

Khác với Search engine nhận diện toàn trang là một dòng dữ liệu, hệ thống Social Listening nhận diện mỗi comment là một dòng dữ liệu. Như hình minh họa dưới đây, bài viết này có 907 phản hồi, tương đương với hệ thống ghi nhận 908 dòng dữ liệu, hay 908 mentions, hay buzz, hay ý kiến người tiêu dùng. Crawlers chỉ có thể thấy những gì công chúng thấy, thu thập được những thảo luận để chế độ public, chứ không lấy được các thảo luận private, tuân thủ theo luật privacy. Tuy nhiên, crawlers có thể lấy được thảo luận trong closed Facebook group, bằng các

đăng nhập bằng một member ID của group đó, nhưng việc này cần có sự đồng ý của admin của group.



Hình 1.2 Hình minh họa trang TTĐT mà Social Listening nhận diện mỗi comment là một dòng dữ liệu.

Hệ thống thu thập toàn bộ dữ liệu có trong trang từ quá khứ đến hiện tại và liên tục quay lại cập nhật các dữ liệu mới tạo ra trên trang cứ 15 phút đến 1 tiếng một lần.

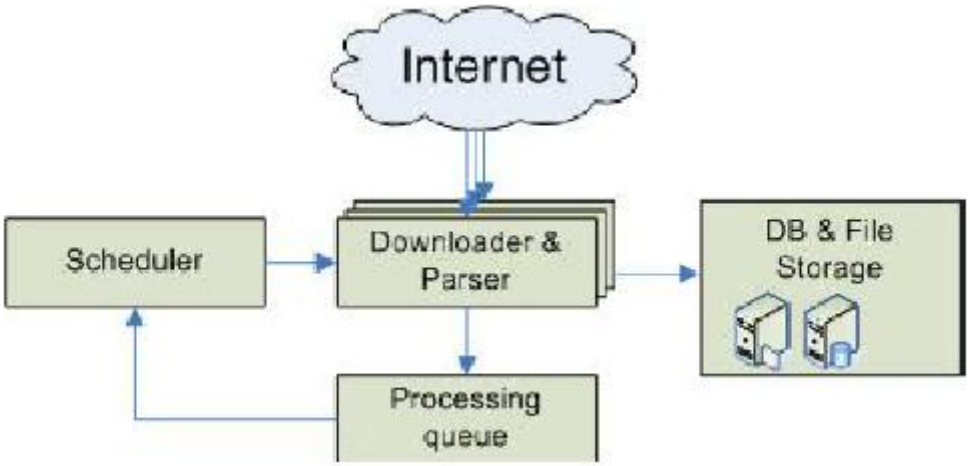
Hình 1.3 Dòng thời gian thể hiện thời điểm thu thập trang

Phương pháp thu thập theo trang phụ thuộc vào 4 yếu tố: đường truyền Internet, tốc độ trả dữ liệu của trang, sự nhận diện nội dung và cấu trúc trang của crawlers và khả năng ngăn chặn crawlers của trang. Các trang diễn đàn lớn thường có sự thay đổi về cấu trúc hàng năm nên khi crawlers khi gặp cấu trúc mới khác với thiết kế ban đầu thì sẽ dẫn đến việc thu thập bị gián đoạn. Đồng thời các publishers

thường có cơ chế nhận diện và chặn việc thu thập dữ liệu của máy tính gây ảnh hưởng đến băng thông. Các crawlers cũng thường xuyên phải cập nhật và nhảy tính danh để vượt qua các cơ chế chặn này. Vì những lý do trên, việc thiếu hay gián đoạn dữ liệu là điều không thể tránh khỏi với các Social Listening tool nên ở Buzzmetrics, một đội ngũ lập trình viên data team phải làm việc liên tục để cập nhật crawlers, thực hiện các biện pháp xử lý ngoài tình huống chuẩn để đảm bảo đầy đủ dữ liệu cho khách hàng, đặc biệt trong các trường hợp chạy chiến dịch hay xử lý khủng hoảng.

1.5.1. Web Crawler

Một Web Crawler là một chương trình máy tính có thể "duyệt web" một cách tự động và theo một phương thức nào đó được xác định trước. Vì là một chương trình nên quá trình " duyệt web" của các web crawler không hoàn toàn giống với quá trình duyệt web của con người (Web crawler phải sử dụng các phương thức dựa trên HTTP trực tiếp chứ không thông qua web browser như con người). Các web crawler thường bắt đầu với một danh sách URL của các web page để ghé thăm đầu tiên. Khi ghé thăm một URL, crawler sẽ đọc nội dung web page, tìm tất cả các hyperlink có trong web page đó và đưa các URL được trỏ tới bởi các hyperlink đó vào danh sách URL. Dựa vào danh sách URL này, Crawler lại tiếp tục quá trình duyệt đệ quy để ghé thăm tất cả các URL chưa được duyệt đến. Quá trình này được gọi là web crawling hoặc là web spidering, các web crawler còn được gọi là các robot (bot) hoặc nhện web (web spider). Thường thì các crawler được tạo ra để phục vụ cho một mục đích, tác vụ nào đó. Ví dụ các máy tìm kiếm (search engine) sử dụng crawler để tải các web page, các web page này sau đó được search engine đánh chỉ mục để có thể cho kết quả nhanh hơn khi được tìm kiếm.



Hình 1.4. Sơ đồ hoạt động của một web crawler đơn giản.

Về bản chất, web crawling chính là quá trình duyệt đệ quy một đồ thị cây có các node là các web page. Tùy thuộc vào chiến lược của crawler, các node có thể được duyệt theo chiều sâu hoặc duyệt theo chiều rộng. Trong thực tế, quá trình crawling web sẽ phải đối diện với rất nhiều vấn đề khó khăn như: kích thước khổng lồ của word wide web, các trang web HTML được viết không chuẩn, hạn chế ghé thăm một URL đã được ghé thăm trước đó, các trang web động, nội dung các trang web được cập nhật thường xuyên,...

1.5.2. Web Scraper

Các trang web chủ yếu được viết bằng các ngôn ngữ đánh dấu như HTML, XHTML và được nhắm đến đối tượng sử dụng là con người chứ không phải máy tính. Các trang web lại chứa đựng nhiều thông tin có ích mà con người có thể muốn thu thập và lưu trữ lại, chính vì thế mà các web scraper được ra đời. Web Scraper là một thuật ngữ để chỉ các phần mềm có khả năng bóc tách và trích xuất thông tin chứa trên các web page một cách tự động. Công việc này được gọi là web scraping, web harvesting hoặc web data extraction. Các web scraper khác với web crawler ở chỗ, trong khi web crawler tập trung vào việc duyệt các trang web thông qua các liên kết hyperlink, thì web scraper lại tập trung vào việc chuyển đổi nội dung có cấu trúc, sau đó bóc tách, trích xuất phần thông tin mong muốn và lưu trữ lại vào các cơ sở dữ liệu hoặc spreadsheet. Các web scraper cũng có thể thực hiện thêm các công đoạn phân tích dữ liệu sau khi đã trích xuất được để phục vụ cho một mục đích nào đó. Một số ứng dụng của web scraping bao gồm: so sánh giá cả thị trường trực

tuyến, nghiên cứu thị trường, thu thập thông tin để thống kê, theo dõi thông tin thời tiết trên các website dự báo thời tiết, tổng hợp tin tức từ nhiều website,...

Một số kỹ thuật được sử dụng trong web scraping có thể kể ra như:

- So trùng: một kỹ thuật đơn giản nhưng khá hiệu quả để tìm kiếm các phần nội dung chữ có sự tương đồng với nhau (do nội dung trang web chủ yếu là ở dạng ký tự). Kỹ thuật này thường sử dụng regular expression (biểu thức chính quy) để so trùng và tìm kiếm.
- Lập trình HTTP: ta có thể lấy được nội dung trang web bằng cách gửi một yêu cầu HTTP đến web server, cũng giống như cách web browser làm. Đây cũng là một kỹ thuật được sử dụng bởi các web clawer.
- Phân tích cấu trúc DOM : Phân tích nội dung HTML của web page và xây dựng một DOM (Document Object Model), giúp scraper có thể duyệt các node trên cây này và chỉ lấy ra phần nội dung mà nó cần.

Một số ứng dụng quan trọng của Web Scraping:

- E-commerce Websites (Website thương mại điện tử): Web Scraping có thể thu thập dữ liệu liên quan đặc biệt đến giá thành của một sản phẩm cụ thể từ các trang web thương mại điện tử khác nhau để so sánh.
- Content Aggregators (Bộ tổng hợp nội dung): Web Scraping được sử dụng rộng rãi bởi các bộ tổng hợp nội dung như bộ tổng hợp tin tức (news) và bộ tổng hợp việc làm (job) để cung cấp dữ liệu cập nhật cho người dùng.
- Marketing and Sales Campaigns (Chiến dịch tiếp thị và bán hàng): Web Scarping có thể được sử dụng để lấy dữ liệu như email, số điện thoại,... cho các chiến dịch tiếp thị và bán hàng.
- Search Engine Optimization- SEO (Tối ưu hóa công cụ tìm kiếm): được sử dụng rộng rãi bởi các công cụ SEO như SEMRush, Majestic,... để cho doanh nghiệp biết cách họ xếp hạng các từ khóa tìm kiếm quan trọng.
- Data for Machine Learning Project (Dữ liệu cho các dự án máy học): Việc truy xuất dữ liệu cho các dự án máy học từ Web Scraping.

1.5.3. Phân biệt Web Crawling và Web Scraping

Web Crawling và Web Scraping là hai khái niệm có liên quan với nhau và có thể có nhiều người nhầm lẫn hoặc chưa phân biệt được sự khác nhau giữa hai khái niệm này. Web Crawling là quá trình thu thập thông tin từ các Website trên mạng Internet theo các đường links cho trước. Các Web Crawler sẽ truy cập các

links này để download toàn bộ nội dung của trang web cũng như tìm kiếm thêm các đường links bên trong để tiếp tục truy cập và download nội dung từ các đường links này. Dữ liệu sau khi được tải về sẽ được đánh chỉ số (indexing) rồi lưu vào cơ sở dữ liệu.

Web Scraping cũng thực hiện việc tìm kiếm và thu thập thông tin nhưng khác với Web Crawling, Web Scraping không thu thập toàn bộ thông tin của một trang web mà chỉ thu thập những thông tin cần thiết, phù hợp với mục đích của người dùng. Trong Web Scraping chúng ta cũng phần nào sử dụng WebCrawler để thu thập dữ liệu, kết hợp với Data Extraction (trích xuất dữ liệu) để tập trung vào các nội dung cần thiết.

Ví dụ như đối với trang amazon.com, Web Crawling sẽ thu thập toàn bộ nội dung của trang web này (tên các sản phẩm, thông tin chi tiết, bảng giá, hướng dẫn sử dụng, các reviews và comments về sản phẩm,…). Tuy nhiên Web Scraping có thể chỉ thu thập thông tin về giá của các sản phẩm để tiến hành so sánh giá này với các trang bán hàng online khác.

1.6. Một số thuật toán kiểm tra phát hiện thay đổi nội dung trang TTĐT

1.6.1. Hàm băm

1.6.1.1. Giới thiệu hàm băm

Hàm băm (hash function) là giải thuật với đầu vào là những khối dữ liệu và kết quả đầu ra là các giá trị băm tương ứng với mỗi giá trị đầu vào. Ở đây giá trị băm có thể được coi như một khóa để phân biệt các dữ liệu với nhau, tuy vẫn còn hiện tượng trùng khóa hay còn gọi là đụng độ nhưng điều này vẫn được chấp nhận và mọi người vẫn đang tìm cách để cải thiện giải thuật nhằm giảm thiểu sự đụng độ đó. Để giảm chi phí tính toán khi tìm một khối dữ liệu trong một tập hợp, người ta sử dụng bảng băm.

Nhận thấy rằng hàm băm h không phải là một song ánh. Do đó, với một thông điệp m bất kỳ, tồn tại thông điệp m’ ≠ m sao cho h(m) = h(m’). Lúc này, ta nói rằng “có xảy ra sự đụng độ”.

Một hàm băm h được gọi là an toàn (hay “ít bị đụng độ”) khi khó có thể xác định được (bằng cách tính toán) cặp thông điệp m và m’ thỏa mãn m ≠ m’ và h(m) = h(m’). Trên thực tế, các thuật toán băm là hàm có tính một chiều, tức là, khi có thông điệp rút gọn rất khó để xây dựng lại thông điệp ban đầu.

Cúng ta có thể sử dụng hàm băm để xác định được tính toàn vẹn dữ liệu hay của thông tin, mọi thay đổi, dù là rất nhỏ, trên thông điệp cho trước, ví dụ như thay đổi giá trị 1 bit, đều làm thay đổi thông điệp rút gọn tương ứng. Đây là tính chất khá hữu ích trong việc kiểm tra chữ ký điện tử, chứng nhận các thông điệp...

Hàm băm là nền tảng cho nhiều ứng dụng mã hóa. Có nhiều thuật toán để thực hiện hàm băm, trong đó, phương pháp SHA-1 và MD5 thường được sử dụng khá phổ biến từ thập niên 1990 đến nay.

Hàm băm mật mã ứng dụng nhiều trong an ninh thông tin, điển hình là kiểm tra tính toàn vẹn của dữ liệu, mã hóa mật khẩu, xác minh tệp tin, chứng thực thông điệp, chữ ký số, và các dạng ứng dụng khác mà chứa hàm băm mật mã bên trong. Hàm băm mật mã còn được sử dụng như các hàm băm thường, phục vụ cho bảng băm như chúng ta đã biết. Vì mã băm của một đối tượng là duy nhất, mã băm đó có thể được dùng như dấu vân tay để nhận dạng đối tượng. Ví dụ để chỉ ra một người sử dụng, cách tốt nhất là sử dụng dấu vân tay là mã băm mật mã dành cho người đó, trong khi mã nhận dạng kiểu khác có thể không rõ ràng và bị trùng lặp.

1.6.1.2. Tính một chiều của hàm băm

Hàm băm được xem là hàm một chiều khi cho trước giá trị băm, khó có thể tái tạo lại thông điệp ban đầu, hay còn gọi là "tiền ảnh" ("pre-image"). Thật vậy, với bài toán tìm "tiền ảnh" tương ứng với một giá trị băm, trong trường hợp lý tưởng, cần phải thực hiện hàm băm cho khoảng 2^n thông điệp.

Cách tấn công nhằm tạo ra một thông điệp khác với thông điệp ban đầu nhưng có cùng giá trị băm gọi là tấn công "tiền ảnh thứ hai" (second pre-image attack).

Hàm băm mật mã phải có khả năng chống lại các loại tấn công mật mã, tối thiểu phải đảm bảo có 3 tính chất sau:

+ Kháng tiền ảnh (Pre-image resistance): Với một mã băm *h* bất kỳ, khó tìm được một thông điệp m nào mà $h = hash(m)$. Điều này làm chúng ta liên tưởng tới tính một chiều của hàm số. Trong góc độ hàm toán học, mã băm là ảnh còn thông điệp là tạo ảnh của mã băm, hay gọi là tiền ảnh. Sức kháng cự tấn công từ ảnh ngược về tiền ảnh gọi là kháng tiền ảnh. Một hàm băm có kháng tiền ảnh yếu là lỗ hổng cho các cuộc tấn công tiền ảnh.

+ Kháng tiền ảnh thứ hai (Second pre-image resistance): Với một thông điệp m_1 bất kỳ, khó tìm được một thông điệp thứ hai m_2 sao cho $m_1 \neq m_2$ và $hash(m_1) = hash(m_2)$. Xác suất xảy ra biến cố có thông điệp m_2 như thế tương tự biến cố

"Cùng ngày sinh". Một hàm băm có kháng tiền ảnh thứ hai yếu là lỗ hổng cho các cuộc tấn công tiền ảnh thứ hai.

+ Kháng xung đột (Collision resistance): Khó tìm được một cặp thông điệp m_1 và m_2 sao cho $m_1 \neq m_2$ và $hash(m_1) = hash(m_2)$. Cặp như thế được gọi là xung đột băm mật mã. Tính chất này đôi khi còn được gọi là kháng xung đột mạnh. Nó yêu cầu chiều dài băm ít nhất phải dài hơn hai lần so với yêu cầu của kháng tiền ảnh, nếu không xung đột có thể xảy ra bởi một cuộc tấn công "Ngày sinh".

1.6.1.3. Cấu trúc hàm băm

Các hàm băm hầu hết đều có chung cấu trúc giải thuật như sau:

+ Cho dữ liệu đầu vào *M* có độ dài bất kỳ. Có thể thêm vào M một số bit để nhận được dữ liệu có độ dài là bội của hằng số cho trước. Chia nhỏ thông điệp thành từng khối có kích thước bằng nhau: $M_1, M_2, \ldots M_s$

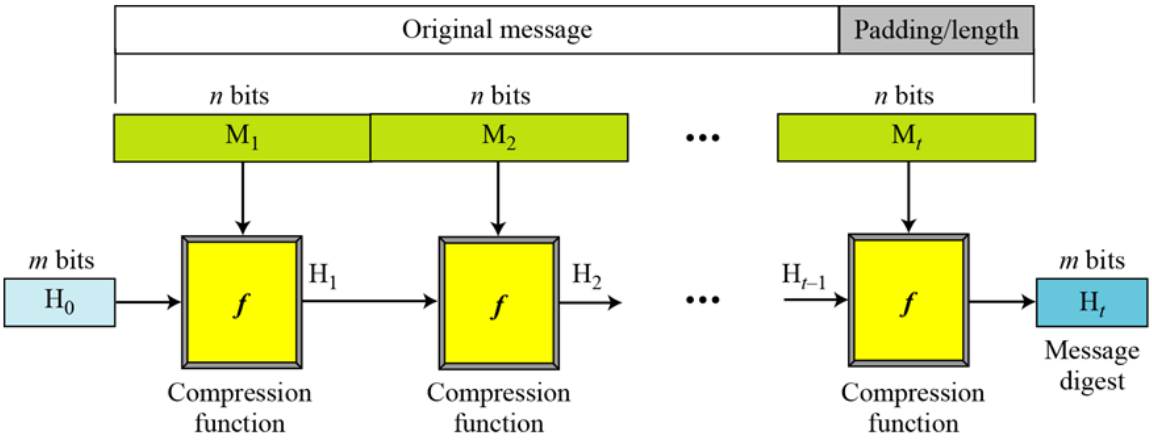
+ Gọi *H* là trạng thái có kích thước *n* bit,

+ Gọi *f* là hàm dùng để trộn khối dữ liệu với trạng thái hiện hành

➢ - Khởi tạo, gán H_0 bằng một vector khởi tạo nào đó

➢ - $H_i = f(H_{i-1}, M_i)$ với $i = 1, 2, 3, \ldots, s$

+ H_s chính là thông điệp rút gọn của thông điệp *M* ban đầu



Hình 1.5 Sơ đồ Merkel-Damgard

Quá trình băm thực hiện trên từng khúc dữ liệu, mỗi khúc được băm một số vòng (số vòng lặp đang sử dụng hiện nay là 64 vòng hoặc 80 vòng). Thuật toán duy trì các biến tương ứng với các từ của mã băm, mỗi từ 32 bits hoặc 64 bits, mỗi vòng thực hiện tính toán các biến trên cơ sở khúc dữ liệu và các thao tác khác. Khúc dữ liệu, tùy theo loại hàm băm có thể được sơ chế hoặc không, được chia thành các mảnh nhỏ cỡ 32 bits hoặc 64 bits và được "trộn" dần vào các vòng băm. Trong mỗi vòng băm, "gia vị" được bổ sung là hằng số, đối với đa số hàm băm hằng số được

lấy từ một hay hai mảng cho trước, trong đó có mảng các giá trị của một hàm phi tuyến hoặc mảng của các số nguyên tố đầu tiên. Dữ liệu được "băm" sử dụng kết hợp các phép toán trên bit. Cuối một vòng, thuật toán thực hiện "trộn" dữ liệu bằng cách gán một số biến đầu ra bằng với một số biến đầu vào nhưng hoán vị. Các biến với giá trị mới lại trở thành biến đầu vào cho vòng tiếp theo. Chúng ta thấy rằng dữ liệu được băm là dữ liệu hỗn hợp, nhưng mô tả rõ hơn thì, thuật toán thực hiện băm các biến trong khi thả mảnh dữ liệu và gia vị vào, rồi trộn các biến, tính toán, khởi tạo các biến cho vòng sau, rồi lại băm tiếp. Khi kết thúc tất cả các vòng băm của một khúc, khúc dữ liệu coi như được nén vào các biến.

Việc tính toán được bắt đầu với một giá trị hash khởi tạo, coi như mã băm hiện hành của khúc dữ liệu đầu tiên. Ở đầu chu kỳ của một khúc, các biến được gán với các từ tương ứng của mã băm hiện hành. Sau khi đi qua hết các vòng băm, các biến kết quả được cộng tương ứng vào các từ của mã băm hiện hành để tạo ra mã băm mới. Quá trình băm thực hiện băm đi băm lại trên từng khúc dữ liệu, mã băm mới của chu kỳ này được dùng làm mã băm hiện hành của chu kỳ tiếp theo, cho tới khúc cuối cùng. Giá trị nối các từ của mã băm sau cùng là mã kết quả của hàm băm.

1.6.2. Thuật toán đối sánh chuỗi

Đối sánh chuỗi là việc so sánh một hoặc vài chuỗi (thường được gọi là mẫu hoặc pattern) với toàn bộ văn bản để tìm ra nơi và số lần xuất hiện của chuỗi đó trong văn bản.

Đối sánh chuỗi là một chủ đề rất quan trọng trong lĩnh vực rộng hơn của xử lý văn bản. Các thuật toán đối sánh chuỗi là những thành phần cơ bản được sử dụng trong việc triển khai các phần mềm thực tiễn tồn tại trong hầu hết các hệ điều hành. Hơn nữa, chúng làm nổi bật các phương pháp lập trình phục vụ như là các mẫu trong các lĩnh vực khác của khoa học máy tính (thiết kế hệ thống và phần mềm). Cuối cùng, các thuật toán đối sánh chuỗi cũng đóng một vai trò quan trọng trong khoa học máy tính lý thuyết bằng cách cung cấp những bài toán thách thức.

1.6.3. Dấu vân tay tài liệu (Document Fingerprint)

Trong khoa học máy tính, dấu vân tay nhận dạng duy nhất dữ liệu gốc cho tất cả các mục đích thực tiễn giống như là việc nhận dạng duy nhất dấu vân tay người trong thực tế. Dấu vân của tài liệu là tập hợp các mã được sinh ra tù các khóa nội dung của tài liệu đó. Mỗi mã đó được gọi là một giá trị băm.

Thông thường, ta chọn chuỗi con từ văn bản ban đầu sau đó áp dụng một hàm toán học cho mỗi chuỗi con đã chọn để tạo ra dấu vân tay tài liệu. Hàm này, giống như một hàm băm, tạo ra một giá trị băm. Giá trị băm này sau đó được lưu trữ trong một chỉ mục (index) để truy cập nhanh khi truy vấn . Khi một tài liệu truy vấn (query document) sẽ được so sánh với tập hợp các số nguyên đã được lưu trữ đó, dấu vân tài liệu cho các truy vấn đó sẽ được tạo ra. Đối với mỗi giá trị băm trong dấu vân tay tài liệu, chỉ mục của truy vấn và một danh sách các dấu vân tay đối sánh được lấy ra. Số lượng giá trị băm chung giữa dấu vân tay truy vấn và mỗi dấu vân tay trong tập hợp đã lưu trữ xác định tài liệu tương ứng đó.

Có một vài phương pháp để lấy dấu vân tay tài liệu dựa trên 4 sự biến đổi của các thông số thiết kết sau:

- Chiến lược lựa chọn (được sử dụng để chọn các chuỗi con từ tài liệu đã cho).
- Kích thước của các chuỗi con (được trích ra từ tài liệu).
- Số lượng giá trị băm (được sử dụng để xây dựng một tài liệu dấu vân tay).
- Hàm Fingerprint (được sử dụng để tạo ra một giá trị băm từ chuỗi con trong tài liệu, như là các checksum, hàm băm, hàm băm mật mã, và chữ kí số).

1.6.4. Thuật toán Rabin Fingerprint

Thuật toán Rabin Fingerprint là một trong nhiều thuật toán Fingerprint thực hiện khóa công khai sử dụng các đa thức trên một trường giới hạn.

Thuật toán Rabin Fingerprint điển hình tạo ra một giá trị băm từ chuỗi con trong các trang web (web pages), bởi vì đây là một thuật toán nhanh và dễ để thực thi, và nó cũng đi kèm với một phân tích chính xác toán học của xác suất đụng độ (hai tập tin có dấu vân tay giống nhau).

Thuật toán được sử dụng trong hệ thống như sau:

Đầu vào: Tài liệu (trang web công khai)

Đầu ra: Dấu vân tay tài liệu (các giá trị băm của tài liệu đó)

Bước 1: Bắt đầu.

Bước 2: Xử lý văn bản, xoá hết tất cả khoảng trắng và các kí tự đặc biệt (như: <, >, %, !, …) từ mã HTML (mã trang web) để thu được một khối văn bản thuần túy (pure text block).

Bước 3: Chia khối văn bản đã xử lý đó thành các chuỗi con có độ dài K.

// Số lượng chuỗi con có độ dài K và số lượng giá trị băm (mã băm) bằng (m-K+1), với m là kích thước của tài liệu.

Bước 4: Tính toán giá trị băm đối với mỗi chuỗi con bằng cách tính H(P) như sau:

// H(P) là một tuyến tính trong n (n là độ dài của P)

Khởi tạo:

Count=K

Tr = T[r..r+n-1]

$H(S) = S(n) + 2*S(n-1) + 4*S(n-2) + \ldots + 2^{n-1}*S(1)$

Do while Count > 0

Sử dụng $H_p(P) = H(P) \bmod p$ như là một giá trị băm (fingerprint) của P

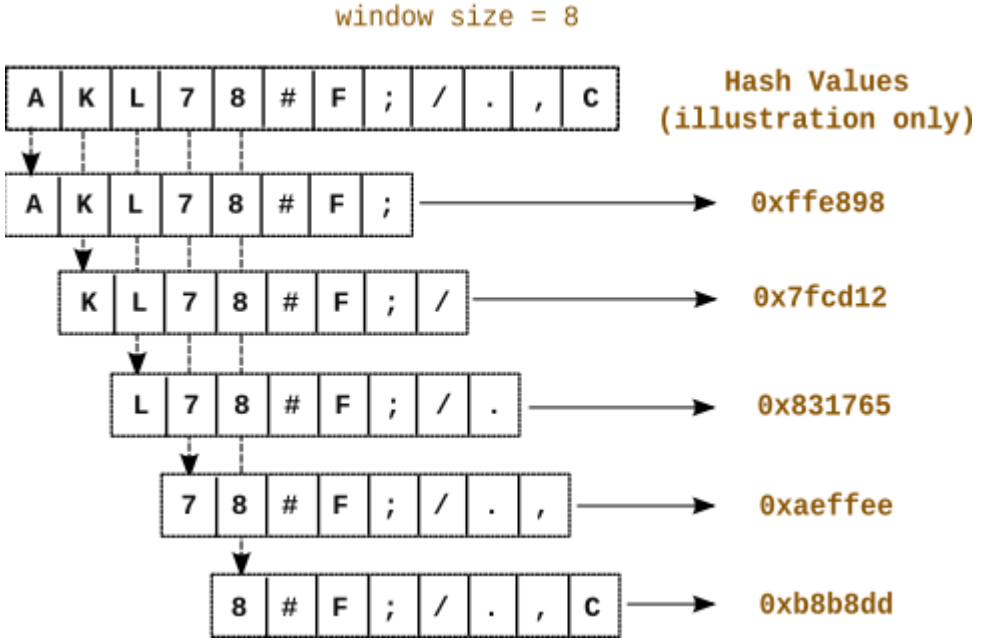
$H_p(T_r) = [2*H_p(T_{r-1}) - (2^n \bmod p) * T(r-1) + T(r+n-1)] \bmod p$

//Tính giá trị băm cho các chuỗi con tiếp theo.

Until Count = 1

Bước 5: Lưu lại các giá trị băm.

Bước 6: Kết thúc.



Hình 1.6 Mô tả thuật toán Rabin Fingerprint

1.6.5. Thuật toán Rabin Fingerprint cải tiến

Thuật toán Rabin Fingerprint cải tiến áp dụng xây dựng hệ thống giám sát website nhằm phát hiện kịp thời các cuộc tấn công để đảm bảo tính toàn vẹn của

trang web đồng thời tạo ra thông điệp cảnh báo có ý nghĩa khi trang web đã bị tấn công.

Thuật toán cải tiến được đề xuất trong hệ thống như sau:

Đầu vào: Tài liệu (trang web công khai)

Đầu ra: Dấu vân tay tài liệu (các giá trị băm của tài liệu đó)

Bước 1: Bắt đầu.

Bước 2: Xử lý văn bản, xoá hết tất cả khoảng trắng và các kí tự đặc biệt (như: <, >, %, !, …) từ mã HTML (mã trang web) để thu được một khối văn bản thuần túy (pure text block).

Bước 3: Từ văn ban M, chia ra thành K khối với kích thước n mỗi khối (n nguyên dương)

Bước 4: Tính mã băm cho các chuỗi con:

Khởi tạo:

Tr = T[r..r+n-1];

K=0;

$H(S) = S(n) + 2*S(n-1) + 4*S(n-2) + \ldots + 2^{n-1}*S(1)$;

While (K<m/n)

{

for (r=K*n; r<=K*n+n; r++)

{

$H_p(T_r)= (H_p(T_r) + T(r))$ mod p *//Tính giá trị băm cho các chuỗi con, p là số nguyên tố lớn.*

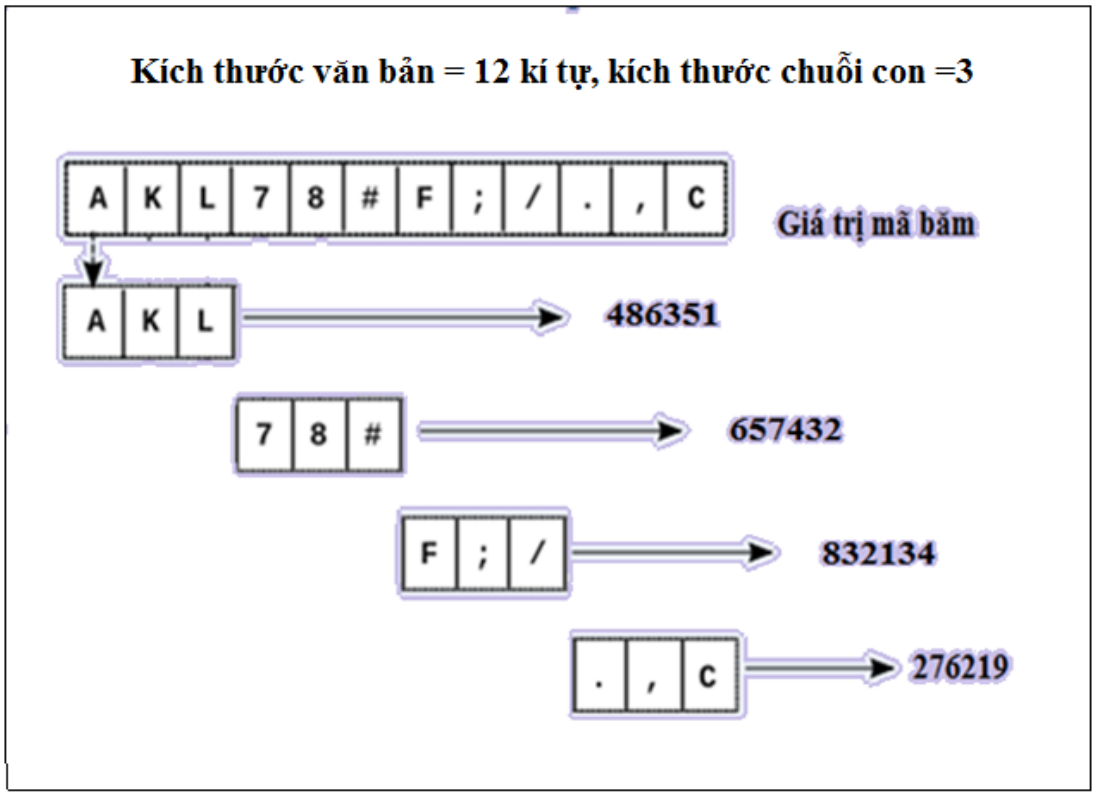
}

K++;

}

Bước 5: Lưu lại các giá trị băm.

Bước 6: Kết thúc.



Hình 1.7 Minh hoạ cải tiến giải thuật

1.6.6. Thuật toán tìm sự khác nhau của hai văn bản "An O(ND) Difference Algorithm"

Thuật toán này được xuất bản lần đầu tiên cách đây 30 năm dưới tựa đề "An O(ND) Difference Algorithm and its Variations" của Eugene Myers, trong cuốn Algorithmica Vol. 1 số 2, 1986, trang 251. Trong bài này, tác giả có đề cập đến định nghĩa của thuật toán theo cách đệ quy, và sử dụng một số mã của ngôn ngữ lập trình hiện có.

+ Cách hoạt động:

So sánh các ký tự của hai tệp văn bản lớn không dễ thực hiện và có xu hướng chậm. Để việc so sánh được dễ hơn, cần so sánh đưới dạng dấu vân, do đó, bước đầu tiên là tính dấu vân cho các dòng văn bản. Nếu các dòng văn bản giống hệt nhau thì sẽ cho các dấu vân giống nhau và ngược lại.

Có một số tùy chọn có thể lựa chọn trước khi tính dấu vân, tùy theo yêu cầu người dùng hay tùy theo một số loại văn bản như xóa những khoảng trống, xóa các từ khóa, id, thẻ...

Phần chính của thuật toán là cách so sánh hai mảng dấu vân và tìm sự khác nhau giữa chúng. Cốt lõi của thuật toán được xây dựng bằng hai phương pháp:

LCS (Longest common subsequence): Là việc thực hiện các thuật toán chia để trị để giải quyết bài toán tìm chuỗi con chung dài nhất.

SMS (Shortest Middle Snake): Phương pháp tìm đường ngắn nhất.

+ Cải tiến:

Thuật toán gốc thực hiện bằng cách sử dụng phương pháp tiếp cận đệ quy, so sánh các trình tự được lập chỉ mục, và truyền các phần của các chuỗi này thành các tham số, trích xuất các mảng con và nối lại chúng.

Thuật toán cải tiến bằng cách thêm một số mã vào chức năng LCS để nhận được kết quả ngay lập tức trên các mảng con giống nhau, đã bị xóa hoàn toàn hoặc chèn vào thay vì phân tích đệ quy chúng.

Kết quả được lưu trữ trong hai mảng chứa đối tượng mô tả sự khác biệt đã tìm thấy.

1.6.7. Thuật toán tìm sự khác nhau của hai hình ảnh

Việc tìm sự khác nhau của hai hình ảnh cơ bản là sự so sánh trực tiếp các điểm ảnh của hai ảnh.

+ Cải tiến:

Việc lấy thông số các điểm ảnh trong C# thường sử dụng 2 phương thức set và get, tuy nhiên khi bạn gọi 2 phương thức này hệ thống sẽ Lock ảnh lại đến khi kết thúc phương thức vừa gọi tự động sẽ UnLock ảnh đó cho việt truy cập lần sau. Chính việc Lock rồi Unlock liên tục đã làm đã làm cho việc xử lý ảnh chậm, nhất là với ảnh có kích thước lớn.

Vì vậy thuật toán có thể cải tiến bằng cách sử dụng kỹ thuật LockBits, lưu các thông tin của ảnh vào mảng để xử lý.

1.7. Kết luận chương

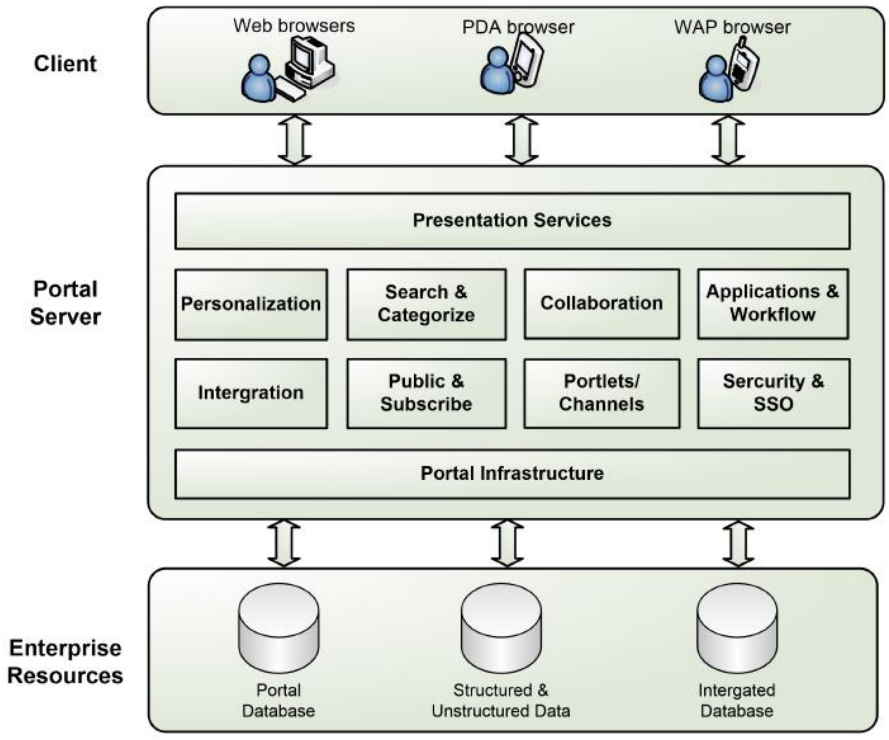
Trong chương 1 luận văn nêu khái niệm tấn công thay đổi nội dung, nguyên nhân và cách khắc phục cùng với một số thuật toán phát hiện sự thay đổi đó.

Việc bị hacker tấn công là điều không thể tránh khỏi vì ngay cả những ông lớn như Google lẫn Facebook cũng đã từng phải chao đảo vì "những vị khách không mời mà đến" này. Tuy nhiên với những kiến thức trên, chúng ta có thể hạn chế được tới 99% các cuộc xâm lăng ngoài ý muốn đó. Suy cho cùng thì tấn công giao diện Deface cũng không quá ghê gớm.

CHƯƠNG 2. NGHIÊN CỨU PHƯƠNG PHÁP KIỂM TRA PHÁT HIỆN THAY ĐỔI NỘI DUNG TRANG TIN XỔ SỐ

Đảm bảo tính toàn vẹn của thông tin, tức là thông tin chỉ được phép xóa hoặc sửa đổi bởi những đối tượng được phép và phải đảm bảo rằng thông tin vẫn còn chính xác khi được lưu trữ hay truyền đi. Ví dụ trường hợp tính toàn vẹn của thông tin bị phá vỡ: thay đổi kết quả xổ số trên trang xổ số kiến thiết từ một đối tượng không được phép dẫn đến nhiều hệ lụy. Chương này trình bày cụ thể về kiến trúc, cơ chế hoạt động của trang TTĐT cùng với mô hình tổng quát cho phương pháp kiểm tra phát hiện giả mạo nội dung trang tin. Bằng cách trình bày cụ thể phương pháp thu thập thông tin, chọn lọc nội dung thông tin cần kiểm tra đối với trang TTĐT, phân tích các công cụ thu thập thông tin sẽ đưa ra phương pháp kiểm tra phát hiện giả mạo nội dung trang kết quả xổ số.

2.1. Khái quát về kiến trúc chung, cơ chế hoạt động của các trang TTĐT.



Hình 2.1 Mô hình kiến trúc Portal

Được xây dựng dựa trên mô hình Web ba tầng (Web Application 3-tier): tầng trình diễn (Client), tầng ứng dụng (Portal Server) và tầng cơ sở dữ liệu (Enterprise Resources).

[1] ***Tầng trình diễn (Client):***

Người dùng có nhiều lựa chọn về nền trình diễn (Internet, Mobile, PDA,...). Hệ thống sẽ tự động gọi các tệp cấu hình sẵn cho tầng nền thông qua lớp Presentation Services.Tầng trình diễn chịu trách nhiệm về cung cấp giao diện cho nhiều loại người dùng khác nhau, có nhiệm vụ lấy các yêu cầu, dữ liệu từ người dùng, có thể định dạng nó theo những qui tắc đơn giản (dùng các ngôn ngữ Script) và gọi các thành phần thích hợp từ tầng Business Logic để xử lý các yêu cầu. Kết quả sau xử lý được trả lại cho người dùng.

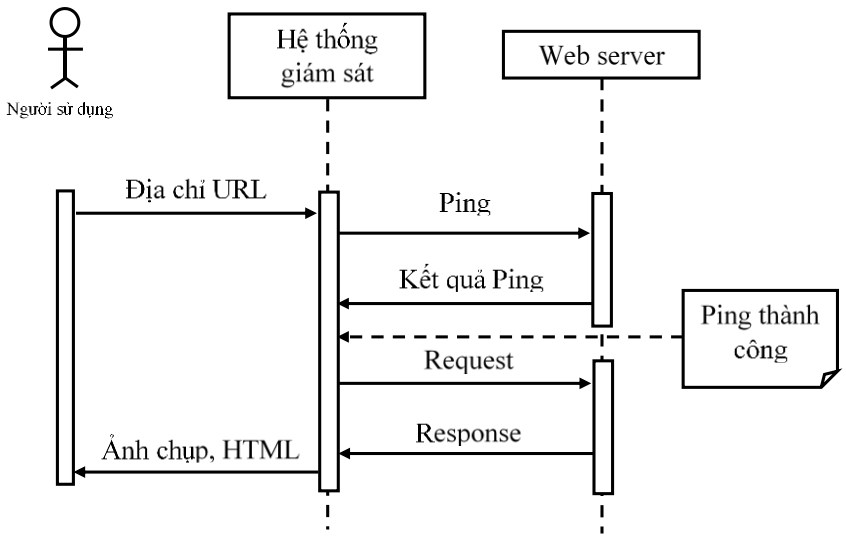
[2] ***Tầng ứng dụng (Portal Server):***

Là môi trường hoạt động và là nơi chứa các ứng dụng của Cổng giao tiếp điện tử. Là đầu mối tiếp nhận và xử lý yêu cầu của người dùng đầu cuối, phân tích, tiền xử lý yêu cầu và chuyển yêu cầu đã xử lý cho phần ứng dụng tương ứng xử lý. Tầng này bao gồm 3 thành phần chính: dịch vụ phục vụ trình diễn, phần ứng dụng, kiến trúc hạ tầng

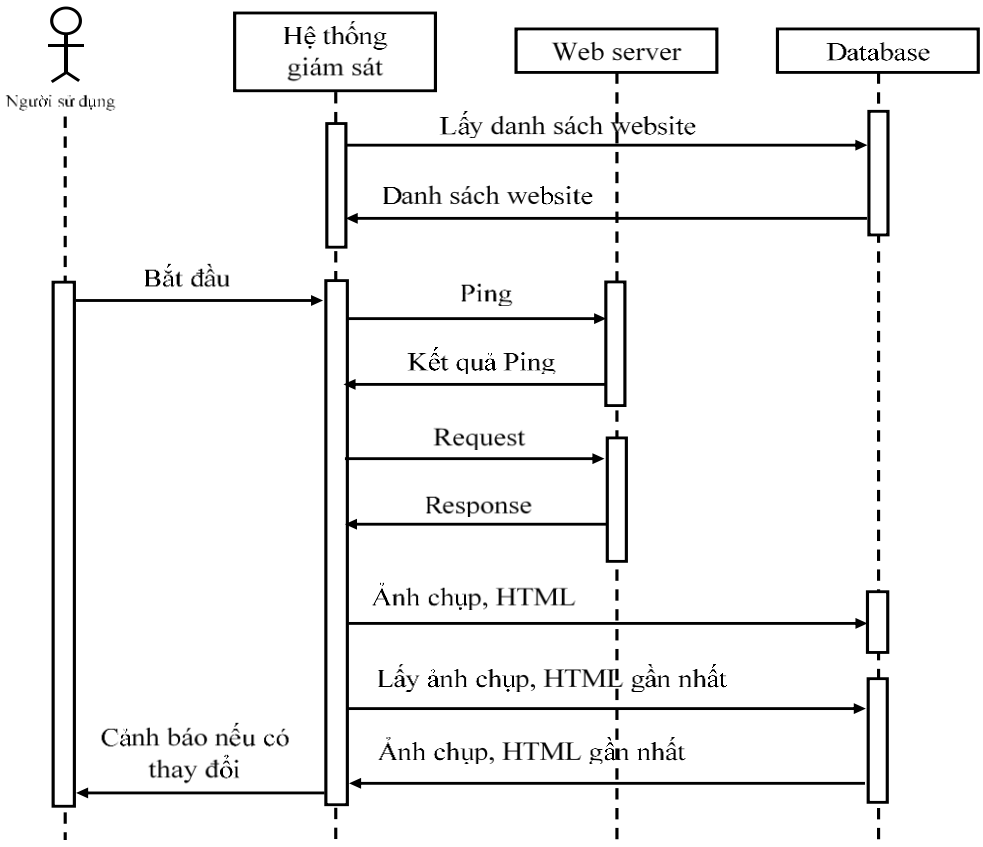
- Dịch vụ phục vụ trình diễn (Persentation Services): Đảm nhận nhiệm vụ đón các yêu cầu từ tầng trình diễn (yêu cầu phía client) và trả về kết quả cho phía client. Đồng thời có nhiệm vụ thực thi các thành phần điều khiển trình diễn của ứng dụng chủ cũng như thực thi các modules giao tiếp với các Server khác (Email, LDAP server).
- Phần ứng dụng (Bussiness Logic): Thực hiện các quy trình xử lý nghiệp vụ và điều khiển. Phần này bao gồm tập các API để thực hiện các luồng công việc, tập các API dùng để tạo ra các dữ liệu và sau đó thông qua Presentation Services xuất ra XHTML, HTML, WML, … tùy theo nền trình diễn mà phía client yêu cầu. Phần này bao gồm các khối chức năng chính sau:

Khối bảo mật (Sercurity & SSO – Single Sign-On): Khối này bao gồm các chức năng cơ bản liên quan đến việc đăng ký, quản lý tài khoản (tạo mới, sửa đổi, xóa, ...) của người sử dụng hoặc nhóm người sử dụng, phân quyền cho người dùng hoặc nhóm người dùng truy cập tới tài nguyên và dịch vụ của hệ thống. Với quan điểm thông tin và dịch vụ chỉ được truy nhập bởi người dùng hợp lệ, Portal cần thiết duy trì hệ thống kiểm tra và xác thực người dùng truy cập. Thêm nữa để tránh cho người dùng phải nhớ quá nhiều tên và mật khẩu khi truy nhập tài nguyên của mình, Portal cũng được cài đặt khả năng xác thực một cửa theo đó người sử dụng (đã nhập duy nhất đáp ứng yêu cầu sử dụng của một cộng đồng người dùng muốn chia sẻ, trao đổi các thông tin trực tuyến trên Internet).

2.2. Mô hình tổng quát cho phương pháp kiểm tra phát hiện thay đổi nội dung bảng kết quả của trang tin xổ số.



Hình 2.2 Biểu đồ trình tự kiểm tra trang TTĐT



Hình 2.3 Biểu đồ trình tự so sánh nội dung

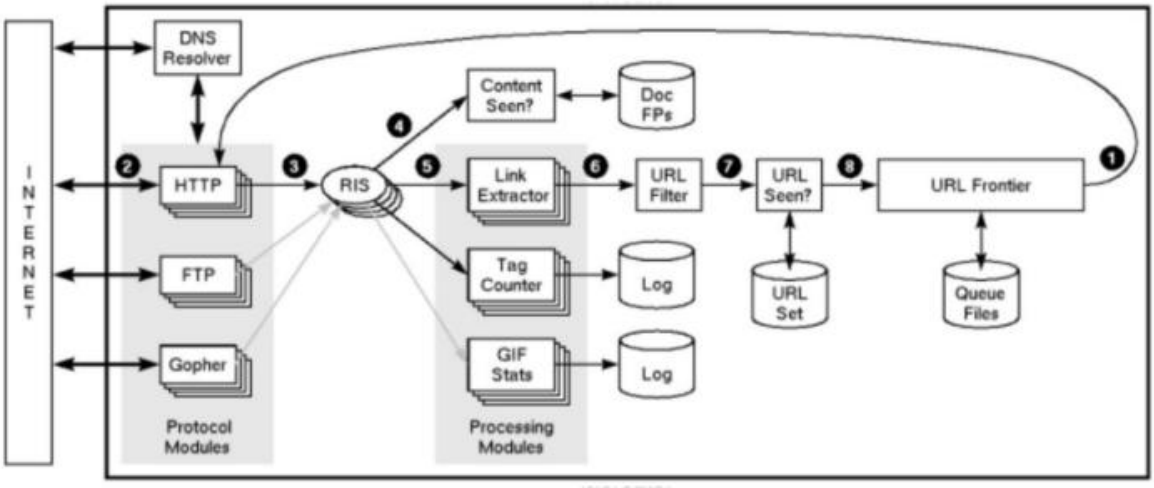
- Quản lý danh sách trang TTĐT cần kiểm tra: gồm các thông tin về địa chỉ và tên trang TTĐT, thời gian kiểm tra, khu vực kiểm tra với các chức năng:
  - Thêm trang TTĐT
  - Xóa trang TTĐT
  - Sửa thông tin trang TTĐT
  - Chuyển trang TTĐT thành hình ảnh
  - Tìm mã dấu vân của đoạn đã chọn trong mã nguồn trang TTĐT
- Giám sát:
  - Quản lý khoảng thời gian giữa mỗi lần kiểm tra
  - Các chức năng bắt đầu, tạm dừng giám sát
  - So sánh dấu vân
  - So sánh hình ảnh
  - Chức năng cài đặt các tùy chỉnh cho quá trình giám sát
- Hiển thị kết quả:
  - Hiển thị thông tin trang TTĐT
  - Hiển thị hình ảnh trang TTĐT
  - So sánh 2 hình ảnh khác thời gian của trang TTĐT, khoanh vùng thay đổi
  - So sánh 2 mã nguồn khác thời gian của trang TTĐT, chỉ ra thay đổi
- Cảnh báo
  - Cảnh báo bằng màu sắc
  - Cảnh báo bằng âm thanh
  - Cảnh báo bằng tin nhắn
- Quản lý dữ liệu hệ thống:
  - Lưu dữ liệu trang TTĐT
  - Load dữ liệu trang TTĐT
  - Lọc dữ liệu theo ngày

2.3. Phân tích, đánh giá một số công cụ thu thập thông tin. Chọn một công cụ thu thập thông tin (dự kiến dùng bộ công cụ Scrapy).

2.3.1. Hệ thống thu thập dữ liệu Mercator

Vào năm 1999 hai tác giả Allan Heydon và Marc najork đã đề xuất một kiến trúc cho hệ thống thu thập dữ liệu Web có khả năng mở rộng trong bài báo có tên "Mervatỏ- A Scalable, Extensible Web Crawler". Trong bài báo này, hai tác giả đưa ra một số thành phần yêu cầu cần có đối với hệ thống thu thập dữ liệu Web có khả năng mở rộng, đó là:

- Thành phần được gọi là URL Frontier cho việc lưu trữ danh sách các URL cần download.
- Thành phần cho việc giải quyết tên miền và địa chỉ IP.
- Thành phần cho việc download trang sử dụng giao thức HTTP.
- Thành phần cho việc trích rút liên kết từ tài liệu HTML.
- Thành phần cho việc kiểm tra URL nào đó đã được viếng thăm hay chưa.



Hình 2.4 Các thành phần chính của Mercator.

(1) Lấy 1 URL đầy đủ (absolute URL) ra khỏi Frontier cho việc download. Một URL đầy đủ bắt đầu với định dạng cái mà chỉ rõ giao thức mạng, ví dụ: http. Trong Mercator, giao thức mạng được cài đặt trong "Protocol Modules". Các module trong "Protocol Modules" được cấu hình trong file được định nghĩa bởi người dùng, và được load động khi hệ thống bắt đầu tiến trình duyệt dữ liệu. Cấu hình mặc định bao gồm các giao thức: HTTP, FTP, Gopher. Dựa vào cấu trúc URL, hệ thống sẽ lựa chọn giao thức thích hợp cho việc download. Sau đó nó sử dụng chức năng fetch

trong “Protocol Modules“ để download trang từ Internet (2) và ghi dữ liệu download được vào trong RIS (3) sử dụng “Content Seen“ để kiểm tra nội dung trang (gắn với URL khác) đã được viếng thăm trước đó hay chưa (4).

Nếu đã được viếng thăm, hệ thống sẽ không xử lý thêm nữa, đồng thời xóa URL khỏi frontier. Dựa vào MIME type trong trang download được, các module thích hợp sẽ được chọn để xử lý tiếp theo (5).

Ví dụ, “Link Extractở“ và “Tag Counter“ sử dụng cho MIME text/html, và GIF Stats xử lý các trang có MIME là image/ gif. Với MIME text/html hệ thống sẽ trích rút tất cả các liên kết bên trong nó, mỗi liên kết được chuyển đổi thành URL đầy đủ, qua “URL Filter“ để kiểm tra (6).

Các URL sau khi qua “URL Filter“ sẽ được “URL Seen“ (7) kiểm tra xem URL đã được viếng thăm hay chưa, hoặc URL có nằm trong Frontier hay không. Nếu không sẽ bổ sung URL mới vào Frontier (8).

2.3.2. Hệ thống thu thập dữ liệu từ Twitter- TwitterEcho

Các dịch vụ truyền thông đa phương tiện xã hội (social media) đã nổi lên trong vài thập kỷ gần đây, thay đổi cách mà chúng ta thông tin với nhau. Do đó, chúng trở thành đối tượng nghiên cứu trong một vài lĩnh vực bao gồm thu thập thông tin, phân tích mạng xã hội, việc thu thập dữ liệu cho dịch vụ này thường là vấn đề phức tạp vì các dịch vụ thường không kết nối trực tiếp tới nơi mà dữ liệu đó được sinh ra, thậm chí cho mục đích nghiên cứu. Do đó những nhà nghiên cứu cần xây dựng hệ thống cho việc thu thập dữ liệu đó hoặc là sử dụng các API được cung cấp bởi mạng xã hội, hoặc là thu thập dữ liệu thông qua Web Crawler.

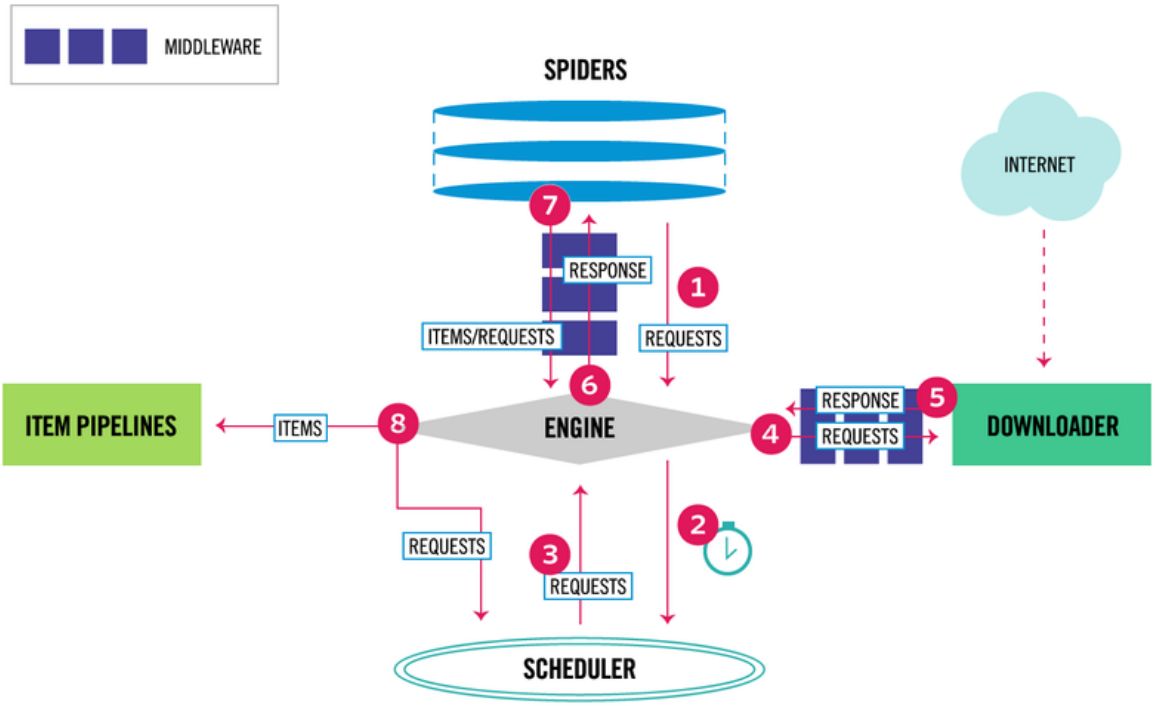
Đặc biệt mạng xã hội Twitter chứa đựng nguồn thông tin cho việc nghiên cứu, từ việc phân tích tương tác của người sử dụng, phân tích việc sử dụng hashtag, và trích dẫn URL, phân tích nội dung cụ thể nào đó (Ví dụ: phân tích sự lan truyền của dịch cúm, điều tra số lượng người nước ngoài nói tiếng Pháp).

Yêu cầu kỹ thuật trong TwitterEcho: Tôn trọng giới hạn, vận hành liên tục, khả năng mở rộng thời gian thức, đầy đủ dữ liệu, tha thứ lỗi, module hóa.

2.3.3. Công cụ HTTrack

HTTrack là công cụ miễn phí cho phép download WWW từ Internet tới thư mục nằm trên máy tính. HTTrack sắp xếp cấu trúc liên kết của site gốc.

2.3.4. Công cụ Scrapy:



Hình 2.5 Các thành phần của công cụ Scrapy

2.3.4.1. Thành phần

- Scrapy Engine: Scrapy Engine có trách nhiệm kiểm soát luồng dữ liệu giữa tất cả các thành phần của hệ thống và kích hoạt các sự kiện khi một số hành động xảy ra
- Scheduler: Giống như một hàng đợi (queue), scheduler sắp xếp thứ tự các URL cần download
- Dowloader: Thực hiện dowload trang web và cung cấp cho engine
- Spiders: Spiders là class được viết bởi người dùng, chúng có trách nhiệm bóc tách dữ liệu cần thiết và tạo các url mới để nạp lại cho scheduler qua engine.
- Item Pipeline: Những dữ liệu được bóc tách từ spiders sẽ đưa tới đây, Item pipeline có nhiệm vụ xử lý chúng và lưu vào cơ sở dữ liệu
- Các Middlewares: Là các thành phần nằm giữa Engine với các thành phần khác, chúng đều có mục đích là giúp người dùng có thể tùy biến, mở rộng khả năng xử lý cho các thành phần. Ví dụ: sau khi dowload xong url, bạn muốn tracking, gửi thông tin ngay lúc đó thì bạn có thể viết phần mở rộng và

sửa lại cấu hình để sau khi Dowloader tải xong trang thì sẽ thực hiện việc tracking.

- Spider middlewares: Là thành phần nằm giữa Eninge và Spiders, chúng xử lý các response đầu vào của Spiders và đầu ra (item và các url mới).
- Dowloader middlewares: Nằm giữa Engine và Dowloader, chúng xử lý các request được đẩy vào từ Engine và các response được tạo ra từ Dowloader
- Scheduler middlewares: Nằm giữa Engine và Scheduler để xử lý những requests giữa hai thành phần

2.3.4.2. Luồng dữ liệu

- Khi bắt đầu crawl một website, Engine sẽ xác định tên miền và tìm vị trí của spider đó và yêu cầu spider đó tìm các URLs đầu tiên để crawl
- Engine nhận danh sách các URLs đầu tiên từ spider, gửi cho Scheduler để sắp xếp
- Engine yêu cầu danh sách cách URLs tiếp theo từ Scheduler
- Engine nhận danh sách các url tiếp theo từ Scheduler vào gửi đến Donwloader (requests)
- Downloader nhận request và thực hiện việc tải trang, sau khi tải xong sẽ tạo một response và gửi lại Engine
- Respone từ Dowloader sẽ được Engine đẩy qua Spiders để xử lý
- Tại Spiders, khi nhận được response, chúng bóc tách thông tin từ response (tilte, content, author, date publish...) và những url có khả năng để crawl và đẩy lại cho Engine (requests)
- Ở bước này, Engine nhận được kết quả từ Spiders sẽ thực hiện 2 công việc: đẩy những dữ liệu đã được bóc tách tới Item Pipeline để xử lý và lưu vào Database, đẩy những URL mới (requests) về Scheduler và quay về bước 3.

Thông tin trên cũng là cơ sở lý thuyết cho việc thu thập dữ liệu trang TTĐT kết quả xổ số của học viên được trình bày trong luận văn này.

Bằng cách sử dụng các thuật toán đã nghiên cứu, tôi đã xây dựng các hàm để thực hiện các công việc cụ thể:

2.4. So sánh thay đổi nội dung mã nguồn web

Việc so sánh thay đổi nội dung mã nguồn, có thể so sánh toàn bộ mã nguồn hoặc chỉ so sánh một phần nội dung (ví dụ: những nội dung xuất hiện trên giao diện,

bỏ qua các thẻ...). Hai phần này đều có chung quy trình, chỉ khác so sánh một phần nội dung cần có thêm bước tiền xử lý.

Sau khi có phần văn bản cần so sánh sử dụng thuật toán Rabin Fingerprint cải tiến để lấy giá trị băm của văn bản để so sánh chúng với nhau, nếu giá trị băm khác nhau thì hai văn bản khác và đã có sự thay đổi.

Rabin Fingerprint cải tiến:

```
class CaitienRB
    {
        long ht = 0;
        long htt = 0;
        long Q = 10000007;
        long D = 2;
        long dM = 1;

        public ArrayList mangluu_giatriban = new ArrayList();

        public void KR(String text, int K)
        {
            int m;
            m = text.Length;
            dM = (dM * D) % Q;

            int khoi = 0;
            while (khoi < m / K)
            {

                for (int i = khoi * K; i < khoi * K + K; i++)
                {
                    htt = (htt * D + text[i]) % Q;

                }
                khoi++;

                mangluu_giatriban.Add(htt);
            }
        }
}
```

Nếu phát hiện sử thay đổi, sử dụng Thuật toán tìm sự khác nhau của hai văn bản để tìm những chỗ thay đổi:

```
        public static Item[] DiffText(string TextA, string TextB, bool trimSpace,
bool ignoreSpace, bool ignoreCase)
                    {
                        Hashtable h = new Hashtable(TextA.Length + TextB.Length);
                        DiffData DataA = new DiffData(DiffCodes(TextA, h,
trimSpace, ignoreSpace, ignoreCase));
                        DiffData DataB = new DiffData(DiffCodes(TextB, h,
trimSpace, ignoreSpace, ignoreCase));

                        h = null;
                        int MAX = DataA.Length + DataB.Length + 1;
```

```
                    int[] DownVector = new int[2 * MAX + 2];
                    int[] UpVector = new int[2 * MAX + 2];
                    LCS(DataA, 0, DataA.Length, DataB, 0, DataB.Length,
DownVector, UpVector);

                    Optimize(DataA);
                    Optimize(DataB);
                    return CreateDiffs(DataA, DataB);
                }
```

2.5. Chuyển đổi Trang web thành hình ảnh

Trong C# có hỗ trợ công cụ giúp chuyển đổi Trang web thành hình ảnh

```
        protected void Capture(object sender, EventArgs e)
        {
            string url = txtUrl.Text.Trim();
            Thread thread = new Thread(delegate()
            {
                using (WebBrowser browser = new WebBrowser())
                {
                    browser.ScrollBarsEnabled = false;
                    browser.AllowNavigation = true;
                    browser.Navigate(url);
                    browser.Width = 1024;
                    browser.Height = 768;
                    browser.DocumentCompleted += new
WebBrowserDocumentCompletedEventHandler(DocumentCompleted);
                    while (browser.ReadyState != WebBrowserReadyState.Complete)
                    {
                        System.Windows.Forms.Application.DoEvents();
                    }
                }
            });
            thread.SetApartmentState(ApartmentState.STA);
            thread.Start();
            thread.Join();
        }

        private void DocumentCompleted(object sender,
WebBrowserDocumentCompletedEventArgs e)
        {
            WebBrowser browser = sender as WebBrowser;
            using (Bitmap bitmap = new Bitmap(browser.Width, browser.Height))

            {
                browser.DrawToBitmap(bitmap, new Rectangle(0, 0, browser.Width,
browser.Height));
                using (MemoryStream stream = new MemoryStream())
                {
                    bitmap.Save(stream, System.Drawing.Imaging.ImageFormat.Png);
                    byte[] bytes = stream.ToArray();
                    imgScreenShot.Visible = true;
                    imgScreenShot.ImageUrl = link + Convert.ToBase64String(bytes);
                }
            }
        }
```

2.6. So sánh thay đổi nội dung hình ảnh trang web

Sau khi đã có hình ảnh trang web, sử dung Thuật toán tìm sự khác nhau của hai hình ảnh đã cải tiến để tìm sự khác nhau giữa hai ảnh, giá tri trả về là một ảnh được bôi đỏ những chỗ thay đổi

```
        public Bitmap SoSanh(Image bm1, Image bm2)
                {
                    Bitmap bmp1 = new Bitmap(bm1);
                    Bitmap bmp2 = new Bitmap(bm2);
                    int w1 = bmp1.Width;
                    int h1 = bmp1.Height;
                    int w2 = bmp2.Width;
                    int h2 = bmp2.Height;
                    int wkq, hkq, x, y;
                    Color b1, b2;

                    if (w1 < w2) wkq = w1; else wkq = w2;
                    if (h1 < h2) hkq = h1; else hkq = h2;
                    Bitmap bitmap = new Bitmap(wkq, hkq);
                    bool[,] k = new bool[wkq, hkq];
                    bool[,] l = new bool[wkq, hkq];
                    bool[,] m = new bool[wkq, hkq];

                    for (y = 0; y < hkq - 1; y++)
                        for (x = 0; x < wkq - 1; x++)
                            k[x, y] = true;

                    for (y = 0; y < hkq - 1; y++)
                    {
                        for (x = 0; x < wkq - 1; x++)
                        {
                            b1 = bmp1.GetPixel(x, y);
                            b2 = bmp2.GetPixel(x, y);
                            bitmap.SetPixel(x, y, Color.FromArgb(b1.R, b1.G,
b1.B));

                            byte g1 = Convert.ToByte(b1.R + b1.G + b1.B);
                            byte g2 = Convert.ToByte(b2.R + b2.G + b2.B);
                            if (Math.Abs(g1 - g2) > 15) k[x, y] = false;
                        }
                    }

                   for (y = 0; y < hkq - 1; y++)
                        for (x = 0; x < wkq - 1; x++)
                            if (k[x, y] == false)
                            {
                                b1 = bmp1.GetPixel(x, y);
                                bitmap.SetPixel(x, y, Color.FromArgb(255, 0, 0));
                            }
                     return bitmap;
        }
```

2.7. Quản lý thời gian thực

Sử dụng công cụ Timer trong C# để liên tục kiểm tra sự thay đổi

```
                private void timer1_Tick(object sender, EventArgs e)
                {
```

```
                KiemTra();
            }
```

2.8. Lưu dữ liệu

```
        public void ThemWeb(string Ten, string DiaChi, string ThongTin)
            {
                conn.Open();
                SQLiteCommand command = conn.CreateCommand();
                command.CommandText = SQLInsertWeb;
                command.Parameters.AddWithValue("TenWeb", Ten);
                command.Parameters.AddWithValue("DiaChi", DiaChi);
                command.Parameters.AddWithValue("ThongTin", ThongTin);
                command.ExecuteNonQuery();
                conn.Close();
            }


        public void XoaWeb(string Id)
            {
                conn.Open();
                SQLiteCommand command = conn.CreateCommand();
                command.CommandText = SQLDelete;
                command.Parameters.AddWithValue("WebId", Id);
                command.ExecuteNonQuery();
                conn.Close();
            }

        public void ThemLichSu(string WebId, DateTime ThoiGian, string Anh, string
Html, string Ping)
            {
                conn.Open();
                SQLiteCommand command = conn.CreateCommand();
                command.CommandText = "INSERT INTO LichSu(WebId, ThoiGian, Anh,
Html, Ping) VALUES(?, ?, ?, ?, ?)";
                command.Parameters.AddWithValue("WebId", WebId);
                command.Parameters.AddWithValue("ThoiGian", ThoiGian);
                command.Parameters.AddWithValue("Anh", Anh);
                command.Parameters.AddWithValue("Html", Html);
                command.Parameters.AddWithValue("Ping", Ping);
                command.ExecuteNonQuery();
                conn.Close();
            }
```

2.9. Kết luận chương

Như đã giới thiệu ở Chương 1, những cuộc tấn công thay đổi trang TTĐT được thực hiện để xâm phạm tính toàn vẹn của nó bằng nhiều hình thức.

Có nhiều biện pháp để giữ cho trang TTĐT được an toàn hơn, nhưng không có biện pháp nào hoàn toàn tối ưu, bởi vì các cuộc tấn công như vậy không thể được ngăn chặn ở các lớp (layer) mạng cao hơn, do đó những cơ chế an ninh tốt hơn cần được cung cấp.

Chương 2 đã đề xuất nghiên cứu phương pháp kiểm tra phát hiện thay đổi nội dung trang tin xổ số nhằm phát hiện kịp thời các cuộc tấn công (như đã nêu) bằng phương pháp đa kiểm tra dựa trên nhiều thuật toán nhằm phát hiện thay đổi để đảm bảo tính toàn vẹn của trang TTĐT.

CHƯƠNG 3. CÀI ĐẶT VÀ THỬ NGHIỆM

Chương này nhằm hiện thực hóa các kết quả đã nghiên cứu, sẽ tiến hành triển khai thử nghiệm thu thập nội dung thông tin, ghi thông tin, kiểm tra phát hiện thay đổi nội dung trang tin kết quả xổ số.

3.1. Cơ sở chọn trang tin kết quả xổ số?

Trong 5 tháng đầu năm 2020 đã ghi nhận 1200 cuộc tấn công Deface, website của doanh nghiệp thường xuyên là mục tiêu tấn công của tin tặc để khai thác đánh cắp các thông tin liên quan bên trong. Trang web xổ số kiến thiết cũng không là một ngoại lệ. Với số lượng người truy cập lớn, thông tin kết quả xổ số đáng giá hàng tỷ đồng, thì nguy cơ bị tấn công thay đổi kết quả xổ số là rất lớn. Chính vì vậy, học viên lựa chọn hệ thống xổ số kiến thiết cho luận văn của mình.

3.2. Cài đặt công cụ thu thập thông tin.

Vì scrapy là một công cụ tạo web spider cực mạnh. Rất nhiều dự án và ứng dụng sử dụng scrapy, ví dụ như lấy toàn bộ hình ảnh của một website, các bài viết theo danh mục và theo chủ đề, tạo bot lấy dữ liệu người dùng như số điện thoại và email trên facebook.. hoặc đơn giản hơn là lấy kết quả sổ xố kiến thiết ... Nên học viên đã lựa chọn công cụ này để thu thập nội dung trang TTĐT trong khuôn khổ luận văn này.

Để chuẩn bị cho scrapy chúng ta cần cài đặt những package sau

Cài đặt

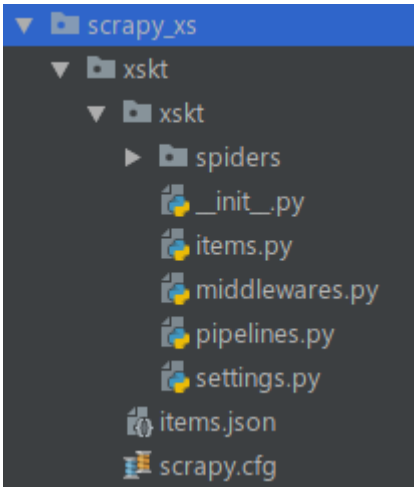
```
pip install scrapy
```

Bắt đầu project

Trong luận văn này sẽ sử dụng scrapy để lấy dữ liệu kết quả sổ xố từ trang TTĐT http://xskt.com.vn, nếu muốn tạo trang web mở đại lý có thể sử dụng source này. Trên virtual environment command line, chạy dòng sau:

```
1. scrapy startproject xskt
```

Scrapy sẽ tạo một folder và các file như sau:



- Scrapy.cfg: file config project scrapy
- Settings.py: file chứa settings cho spiders
- Middlewares.py: file chứa spider middlewares, là những framework được hook vào scrapy processing.
- Items.py: file chứa cấu trúc của item mà bạn sẽ tách dữ liệu. Nói đơn giản thì nó là một cái khung, sau khi lấy được dữ liệu từ spiders, sẽ đặt dữ liệu vào đây và xử lý
- Pipelines.py: sau khi đặt dữ liệu vào cho items.py, sẽ xử lý ở function process_items trong pipeline. Có thể save vào database, hoặc chỉ trả về item đó
- Folder spiders: là nơi tạo file spider để chạy dữ liệu.

3.3. Phương pháp thu thập thông tin từ trang TTĐT về kết quả xổ số.

Khi muốn lấy dữ liệu của một site nào đó, thì điều quan trọng là phân tích cấu trúc trang TTĐT. Những dữ liệu nào trên site đó ta cần lấy, có những link nào trên site đó hỗ trợ lấy dữ liệu dễ dàng... Bây giờ sẽ phân tích site http://xskt.com.vn

Kết quả xổ số được chia làm 3 miền, sẽ lấy miền nam làm ví dụ, làm tương tự cho các miền còn lại. Khi vào site, kéo xuống sẽ thấy "Kết quả sổ xố toàn quốc". Tại đây sẽ click vào miền nam

Hình 3.1 Màn hình trang chủ trang xổ số kiến thiết

Browser sẽ đưa xuống phần "KQXS Miền Nam". Ở mục này sẽ có 3 link dẫn đến mục kết quả sổ xố: "KQXS Miền Nam" sẽ đưa đến trang chứa kết quả mới nhất toàn miền nam, "Ngày 05/02" đưa đến trang chứa kết quả theo ngày, "Thứ Tư" sẽ đưa đến kết quả sổ xố ngày thứ tư. Để dễ dàng lấy dữ liệu từ tất cả các ngày trong ngày, tháng năm sẽ chọn link từ "Ngày 05/02" và lấy tiếp link từ "Ngày 05-02-2020"

KQXS MIỀN NAM NGÀY 05/02 (THỨ TƯ) xskt.com.vn

Thứ 4 05/02	Cần Thơ	Đồng Nai	Sóc Trăng
G.8	12	85	50
G.7	049	776	829
G.6	1630 9983 5171	2674 3319 3899	2775 4297 0225
G.5	7909	8155	6926
G.4	22945 57619 28931 56336 32282 17320 17466	87999 64235 67567 99722 06623 50102 66159	33501 89613 68512 49367 39770 01447 71795
G.3	17289 53667	05527 63053	95737 00567
G.2	22040	39121	73100
G.1	73234	40917	06593
ĐB	369663	624832	360834

Đầy đủ 2 số 3 số Lịch

XSMN 5-2 XSMN 30 ngày

Hình 3.2 Kết quả xổ số miền Nam ngày 5/02/2020

Link sẽ có dạng là https://xskt.com.vn/ket-qua-xo-so-theo-ngay/mien-nam-xsmn/5-2-2020.html.

Coding

Bây giờ quay lại phần coding, mở file items.py lên và chỉnh sửa như sau:

```
# -*- coding: utf-8 -*-

# Define here the models for your scraped items
#
# See documentation in:
# http://doc.scrapy.org/en/latest/topics/items.html

import scrapy

class XsktItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    xs_info = scrapy.Field()
    xs_data = scrapy.Field()
```

Tạo 2 item cho xskt là xs_info và xs_data. xs_info dùng lưu trữ thứ ngày tháng của bảng xổ số, xs_data dùng lưu trữ kết quả xổ số các tỉnh miền nam

Mở file settings.py lên và bạn hãy thêm FEED_FORMAT ở cuối file

```
FEED_FORMAT = 'json'
```

FEED_FORMAT cho phép định dạng dữ liệu xuất ra theo dạng nào, xuất ra ở dạng json trong ví dụ này

Tiếp theo tạo file xosokienthiet.py trong folder xskt/xskt/spiders

```
import scrapy
import calendar
import datetime

from scrapy.spiders import CrawlSpider
from ..items import XsktItem #1

def get_total_date_month(year, month): #2
    now = datetime.datetime.now()
```

```
    total_date = calendar.monthrange(year, month)[1]

    if year == now.year and month == now.month and now.day < total_date:
        return now.day

    return total_date
```

1. Import XsktItem vào spider để chuẩn bị đổ dữ liệu lấy được vào item
2. Function get_total_date_month sẽ tính toán xem có bao nhiêu ngày trong tháng đã đưa vào. Nếu cùng năm và cùng tháng thì chỉ lấy tới ngày hiện tại

Tiếp tục ta sẽ tạo class SoxokienthietSpider để lấy dữ liệu từ site xskt.com.vn

```
class SoxokienthietSpider(CrawlSpider):
    name = 'xosokienthiet'
    allowed_domains = ['xskt.com.vn'] #1

    start_urls = []

    month_to_scrap = 1
    year_to_scrap = 2020
    total_date = get_total_date_month(year_to_scrap, month_to_scrap)

    for i in range(1, total_date):
        start_urls.append('http://xskt.com.vn/ket-qua-xo-so-theo-ngay/mien-nam-xsmn/'
        '{0}-{1}-{2}.html'.format(i, month_to_scrap, year_to_scrap)) #2

    def parse(self, response): #3
        xs_item = XsktItem()
        tmp_data = {}
        data_resp = scrapy.Selector(response)

        xs_item['xs_info'] = [
            # Thứ
```

```
data_resp.xpath("//table[@id='MN0']/tr/th[1]/a/text()").extract_first(),
            # Ngày tháng
            data_resp.xpath("//table[@id='MN0']/tr/th[1]/text()").extract_first()
        ] #4

        for i in range(2, 5):
            # Các tỉnh trong bảng xổ số
            tmp_location = data_resp.xpath("//table[@id='MN0']/tr/th[{0}]/a/text()".format(i)).extract_first()
            tmp_data[tmp_location] = {}

            for j in range(2, 11):
                # Cột các giải từ giải 8 đến giải đặc biệt
                tmp_giai = data_resp.xpath("//table[@id='MN0']/tr[{0}]/td[1]/text()".format(j)).extract_first()
                # Các số trúng thưởng trong cột theo tỉnh
                tmp_number = data_resp.xpath("//table[@id='MN0']/tr[{0}]/td[{1}]//text()".format(j, i)).extract()
                tmp_data[tmp_location][tmp_giai] = ", ".join(tmp_number)

        xs_item['xs_data'] = tmp_data #5

        yield xs_item #6
```

1. Scrapy cho phép quản lý link theo domain để lấy dữ liệu, những domain ngoài allowed_domain scrapy sẽ bỏ qua.
2. Lấy số ngày từ function get_total_date_month sau đó đưa vào vòng lặp for. Đưa các giá trị như ngày, tháng, năm vào để tạo một list cho scrapy chạy lấy dữ liệu vào link (http://xskt.com.vn/ket-qua-xo-so-theo-ngay/mien-nam-xsmn/01-6-2018.html) đã phân tích lúc trước.
3. Parse là function kế thừa từ class CrawlSpider. Function này sẽ parse respone lấy được của từng link trong danh sách trên
4. Đưa thứ ngày tháng được phân tích từ xpath vào xs_info. extract_first() sẽ trả về trực tiếp giá trị là string.

5. Tương tự cho xs_data, đưa số các giải vào.
6. Yield tương tự như return, nhưng điểm khác biệt là nó sẽ lưu lại dữ liệu được yield, sau đó khi kết thúc nó sẽ trả về toàn bộ dữ liệu mà chương trình đã chạy. Cần lưu ý là yield không lưu ở memory, mà được tạo ra và dùng trực tiếp.

Code file xosokienthiet.py sẽ như sau:

```
import scrapy
import calendar
import datetime

from scrapy.spiders import CrawlSpider
from ..items import XsktItem

def get_total_date_month(year, month):
    now = datetime.datetime.now()
    total_date = calendar.monthrange(year, month)[1]

    if year == now.year and month == now.month and now.day < total_date:
        return now.day

    return total_date

class SoxokienthietSpider(CrawlSpider):
    name = 'xosokienthiet'
    allowed_domains = ['xskt.com.vn']

    start_urls = []

    month_to_scrap = 1
    year_to_scrap = 2020
    total_date = get_total_date_month(year_to_scrap, month_to_scrap)

    for i in range(1, total_date):
        start_urls.append('http://xskt.com.vn/ket-qua-xo-so-theo-ngay/mien-nam-xsmn/'
```

```
            '{0}-{1}-{2}.html'.format(i, month_to_scrap, year_to_scrap))

        def parse(self, response):
            xs_item = XsktItem()
            tmp_data = {}
            data_resp = scrapy.Selector(response)

            xs_item['xs_info'] = [
                # Thứ
data_resp.xpath("//table[@id='MN0']/tr/th[1]/a/text()").extract_first(),
                # Ngày tháng
                data_resp.xpath("//table[@id='MN0']/tr/th[1]/text()").extract_first(),
                self.year_to_scrap
            ]

            for i in range(2, 5):
                # Các tỉnh trong bảng xổ số
                tmp_location =
data_resp.xpath("//table[@id='MN0']/tr/th[{0}]/a/text()".format(i)).extract_first()
                tmp_data[tmp_location] = {}

                for j in range(2, 11):
                    # Cột các giải từ giải 8 đến giải đặc biệt
                    tmp_giai =
data_resp.xpath("//table[@id='MN0']/tr[{0}]/td[1]/text()".format(j)).extract_first()
                    # Các số trúng thưởng trong cột theo tỉnh
                    tmp_number =
data_resp.xpath("//table[@id='MN0']/tr[{0}]/td[{1}]//text()".format(j, i)).extract()
                    tmp_data[tmp_location][tmp_giai] = ", ".join(tmp_number)

            xs_item['xs_data'] = tmp_data

            yield xs_item
```

Đoạn code trong bài viết này sẽ lấy toàn bộ kết quả xổ số miền nam trong tháng 1/2020. Có thể chạy thử và export ra file json như sau/

```
scrapy crawl xosokienthiet -o items.json
```

Nếu muốn lưu kết quả vào database, phải kích hoạt pipeline. Sẽ dùng sqlite3 trong ví dụ này. Mở file settings.py vào tìm ITEM_PIPELINE ra và uncomment

```
ITEM_PIPELINES = {
   'xskt.pipelines.XsktPipeline': 300,
 }
```

Bây giờ hãy mở file pipelines.py và chỉnh sửa như sau:

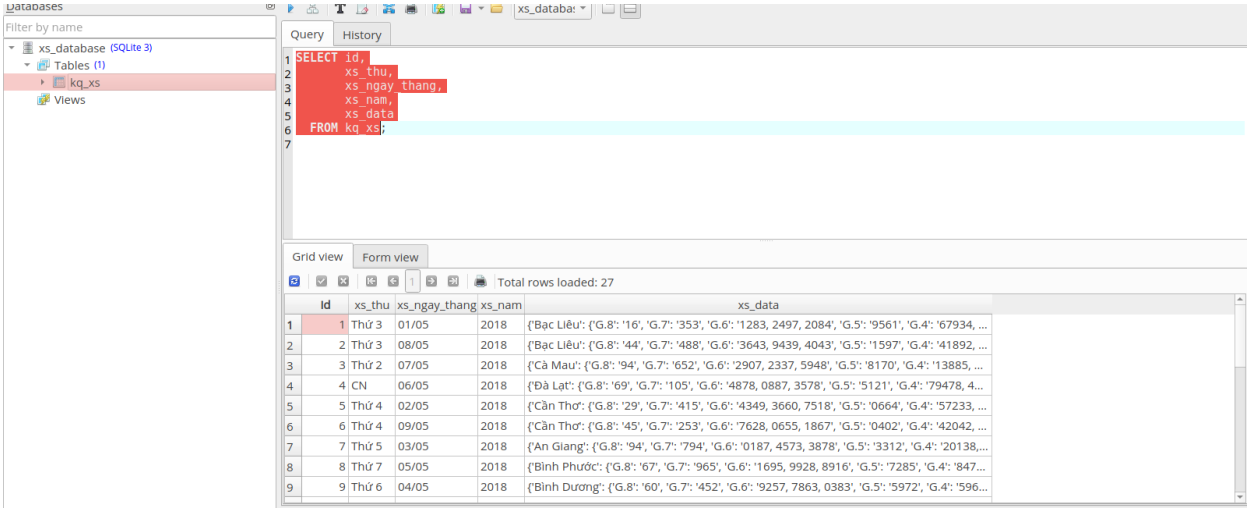
```
# -*- coding: utf-8 -*-

# Define your item pipelines here
#
# Don't forget to add your pipeline to the ITEM_PIPELINES setting
# See: http://doc.scrapy.org/en/latest/topics/item-pipeline.html
from sqlite3 import dbapi2 as sqlite

class XsktPipeline(object):
   def __init__(self):
      self.connection = sqlite.connect('./xs_database.db')
      self.cursor = self.connection.cursor()
      self.cursor.execute('CREATE TABLE IF NOT EXISTS kq_xs '
                          '(id INTEGER PRIMARY KEY, xs_thu VARCHAR(80),'
                          'xs_ngay_thang VARCHAR(80), xs_nam
VARCHAR(80), xs_data TEXT)')

   def process_item(self, item, spider):
      self.cursor.execute("select * from kq_xs where xs_thu=? and
xs_ngay_thang=? and xs_nam=?", (item['xs_info'][0], item['xs_info'][1],
item['xs_info'][2]))

      result = self.cursor.fetchone()
      if not result:
         self.cursor.execute(
```

```
            "insert into kq_xs (xs_thu, xs_ngay_thang, xs_nam, xs_data) values (?, ?, ?, ?)",
            (item['xs_info'][0], item['xs_info'][1], item['xs_info'][2], str(item['xs_data'])))

        self.connection.commit()

        return item
```

Sau đó chạy command:

scrapy crawl xosokienthiet

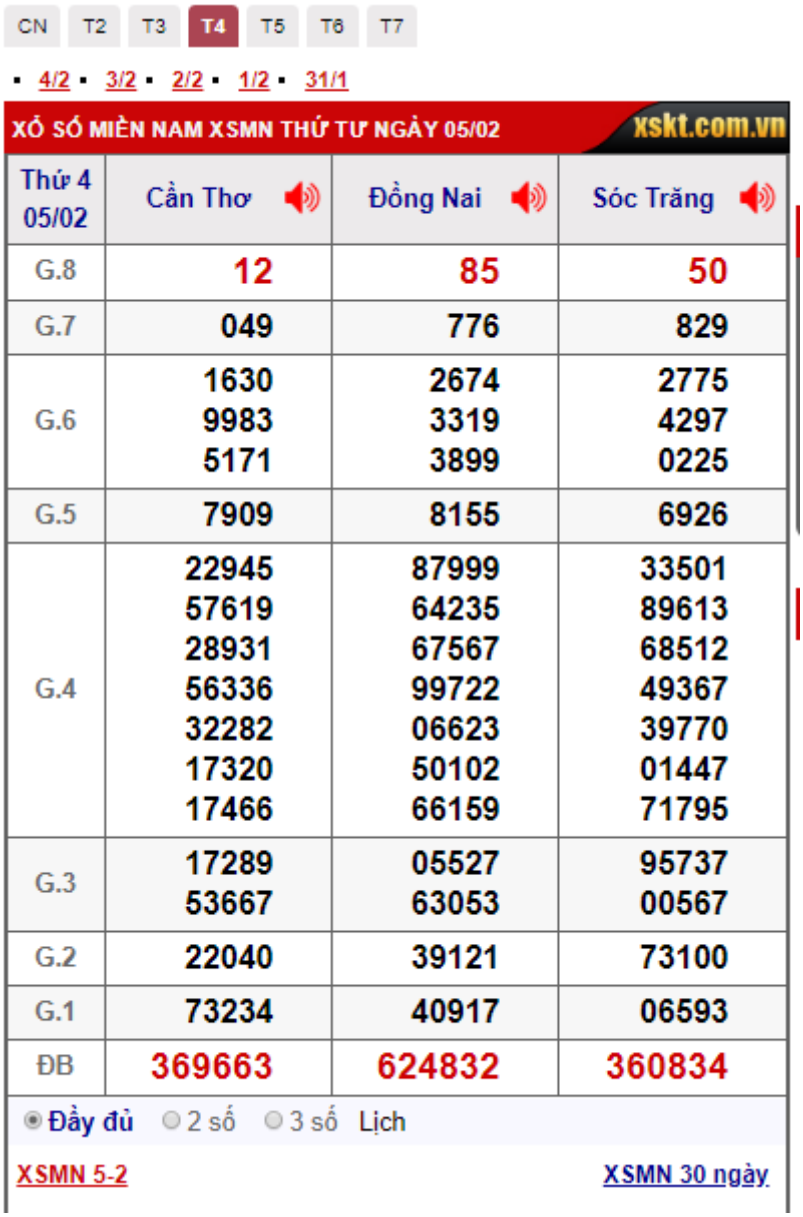
Kết quả sau khi chạy scrapy:



Hình 3.3 Kết quả sau khi chạy Scrapy

❖ **Phân tích site để lấy xpath**

Đầu tiên cần vào site lấy dữ liệu https://xskt.com.vn/ket-qua-xo-so-theo-ngay/mien-nam-xsmn/5-2-2020.html



CN T2 T3 T4 T5 T6 T7

▪ 4/2 ▪ 3/2 ▪ 2/2 ▪ 1/2 ▪ 31/1

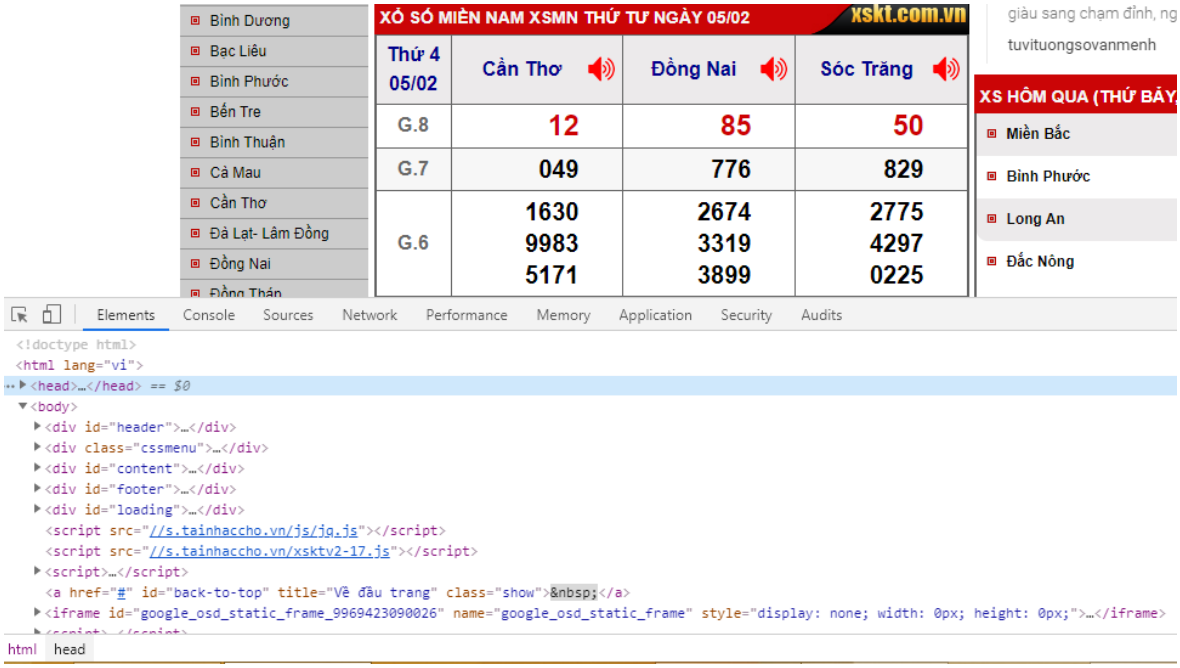
XỔ SỐ MIỀN NAM XSMN THỨ TƯ NGÀY 05/02 xskt.com.vn

Thứ 4 05/02	Cần Thơ	Đồng Nai	Sóc Trăng
G.8	12	85	50
G.7	049	776	829
G.6	1630 9983 5171	2674 3319 3899	2775 4297 0225
G.5	7909	8155	6926
G.4	22945 57619 28931 56336 32282 17320 17466	87999 64235 67567 99722 06623 50102 66159	33501 89613 68512 49367 39770 01447 71795
G.3	17289 53667	05527 63053	95737 00567
G.2	22040	39121	73100
G.1	73234	40917	06593
ĐB	369663	624832	360834

◉ Đầy đủ ○ 2 số ○ 3 số Lịch

XSMN 5-2 XSMN 30 ngày

Để lấy thứ ta cần biết là table chứa bảng số này có class hay id gì khác biệt không, dùng Inspector (nhấn F12 hoặc click phải lên Thứ 4) để xem



Theo hình trên table này có cả Id và class để chúng ta khai thác. Nhưng vì class sẽ có thể dễ bị trùng với các element khác, nên chúng ta sẽ sử dụng id. Hãy ghi lại MN0.

Tiếp theo ta sẽ xem tới "Thứ 4", ta thấy "Thứ 4" nằm trong table (id = MN0) > tr > th[1] > a (text). th[1] chỉ định th đầu tiên trong table data_resp.xpath("//table[@id='MN0']/tr/th[1]/a/text()").extract_first()

Tương tự cho ngày tháng sẽ là table (id = MN0) > tr > th[1] (text) data_resp.xpath("//table[@id='MN0']/tr/th[1]/text()").extract_first()

Các element khác cũng làm tương tự như vậy.

3.4. Xây dựng một kịch bản thử nghiệm.

Kịch bản: Chạy chương trình 1 giờ, 5 giờ, 1 ngày với thời gian kiểm tra là 10 phút/lần, vùng kiểm tra là vùng chứa các nội dung chính, có lưu dữ liệu mã nguồn, ảnh chụp trang TTĐT với trang TTĐT: http://xskt.com.vn

3.5. Kết quả thử nghiệm thu thập nội dung thông tin, ghi thông tin, kiểm tra phát hiện thay đổi nội dung trang tin kết quả xổ số.

Lần 1, chạy 1 giờ từ 19h00 đến 20h05 ngày 15/2/2020.

Bảng 3-1. Kết quả thử nghiệm lần 1

STT	Website	Số lần KT	Số lần phát	Tình trạng

			hiện thay đổi	bất thường
1	xskt.com.vn	7	0	Không

Lần 2, chạy 5 giờ từ 14h00 đến 19h05 ngày 16/02/2018

Bảng 3-2. Kết quả thử nghiệm lần 2

STT	Website	Số lần KT	Số lần phát hiện thay đổi	Tình trạng bất thường
1	xskt.com.vn	31	1	Không

Lần 3, chạy 1 ngày từ 19h00 ngày 17/02/2020 đến 19h05 ngày 18/02/2020

Bảng 3-3. Kết quả thử nghiệm lần 3

STT	Website	Số lần KT	Số lần phát hiện thay đổi	Tình trạng bất thường
1	xskt.com.vn	145	1	Không

3.6. Phân tích, đánh giá kết quả thử nghiệm.

Hệ thống chạy ổn định, không bị lỗi, cảnh báo chính xác bằng âm thanh khi phát hiện thay đổi, gửi cảnh báo về email cho học viên mỗi khi có sự thay đổi, mức độ chiếm bộ nhớ RAM ổn định, không tăng khi hệ thống chạy lâu dài, dung lượng lưu trữ dữ liệu kiểm tra trang TTĐT trên ổ cứng trung bình 200KB/lần kiểm tra

(gồm ảnh chụp, dữ liệu lưu trong database). Nếu tiến độ kiểm tra 10 phút/lần thì 1 ngày 1 trang TTĐT lưu dữ liệu tốn 30MB dung lượng.

3.7. Kết luận chương

Sau khi hoàn thành demo đã đạt được kết quả như sau:

- Phát hiện được tất cả các thay đổi xảy ra của website
- Gửi cảnh báo về email cho quản trị viên mỗi khi có sự thay đổi.
- Giao diện ứng dụng khá thuận tiện
- Dễ dàng cho quản trị viên kiểm tra và phát hiện vị trí cần khắc phục khi có sự cố.
- Tốc độ chương trình tương đối ổn định .

KẾT LUẬN

❖ Các kết quả đạt được:

Nghiên cứu về các giải thuật chính được sử dụng để phát hiện sự thay đổi về nội dung của website, giúp tăng cường khả năng giám sát, phát hiện và cảnh báo, nhằm hỗ trợ cho người quản trị có thể phản ứng nhanh hơn trong các trường hợp trang TTĐT của mình bị tấn công.

Nắm rõ các nguy cơ mất ATTT đối với các trang TTĐT, đặc biệt là thay đổi nội dung. Từ đó nghiên cứu các phương pháp thu thập thông tin, các phương pháp kiểm tra tính toàn vẹn của thông tin để phân tích, thử nghiệm, kiểm tra phát hiện thay đổi nội dung trang TTĐT về kết quả xổ số.

❖ Hướng phát triển:

Tìm hiểu thêm về các tấn công hiện đại, có nguy cơ gây tổn thương trang TTĐT, và tìm cách khắc phục nhằm đảm bảo tính an toàn của trang TTĐT.

Tiếp tục nghiên cứu và phân tích bộ công cụ Scrapy và những công cụ thu thập thông tin khác nhằm phát hiện các điểm yếu khác để khắc phục

TÀI LIỆU THAM KHẢO

Tiếng việt

[1] Phan Đình Diệu (2002)- Lý thuyết mật mã và an toàn thông tin — NXB Đại Học Quốc Gia Hà Nội 2002.

[2] Phạm Huy Điển, Hà Huy Khoái (2003)- Mã hóa thông tin - Cơ sở Toán học & ứng dụng, Nhà xuất bản Đại học Quốc gia Hà nội.

[3] Hà Huy Khoái (1997), Nhập môn số học thuật toán, Nhà xuất bản Khoa học.

[4] Nguyễn Ngọc Tuấn, Hồng Phúc (2005)- Công nghệ bảo mật World Wide Web- Nhà xuất bản Thống kê.

[5] Nguyễn Đình Vinh (2005)- Những vấn đề cơ bản của an toàn thông tin (Tập 1, tập 2)- Học viện kỹ thuật mật mã.

[6] Pfleege (2004)- An toàn tính toán (bản dịch)- Học viện kỹ thuật mật mã.

Tiếng Anh

[7] A. Menezes, P. van Oorschot và S. Vanstone. (1996) - Handbook of Applied Cryptography, Fifth Edition- CRC Press.

[8] Douglas Stinson (2007)- Cryptography: Theory and Practice. Boca Raton. FL- CRC Press.

[9] Nik Goots, Boris Izotov, Alex Moldovyan and Nik Moldovyan (2003)- "Modern CryptographyProtect Your Data with Fast Block Ciphers"- A-LIST Publishing.

[10] William Stallings (2003)- Cryptography and Network Security: Principles and Practice. Third Edition- Pearson Education.