PHÁT HIỆN VÀ CẢNH BÁO SỰ THAY ĐỔI CỦA WEBSITE DỰA TRÊN THAY ĐỔI NỘI DUNG VÀ CẤU TRÚC HTML

***Trần Đắc Tốt*[1]**
***Vũ Văn Vinh*[1]**

TÓM TẮT

Thực tế cho thấy hậu quả của việc tấn công làm thay đổi giao diện, nội dung website của hacker là đặc biệt nghiêm trọng. Vì vậy cần phải có những phương pháp cho phép kịp thời phát hiện những hình thức tấn công này, nhằm hạn chế tối những thiệt hại mà hacker gây ra. Trong bài báo này chúng tôi trình bày một phương pháp mới cho phép phát hiện sự thay đổi giao diện, nội dung của website. Phương pháp này được phát triển dựa trên thuật toán HTML Diff kết hợp với hàm băm MD5, và nó đã được xây dựng thành một ứng dụng với giao diện hài hòa, dễ sử dụng. Các thay đổi như chèn thêm nội dung mới, xóa hay sửa nội dung cũ, thay đổi các định dạng về màu sắc, kích thước, kiểu chữ của nội dung sẽ được ứng dụng ngay lập tức ghi nhận và thông báo tới người quản trị website. Ứng dụng cũng sẽ làm nổi bật nhưng vị trí đã thay đổi và gửi thư cảnh báo và khuyến nghị cho người quản trị trang web để xử lý.

Từ khóa: *Giám sát sự thay đổi, tấn công defacement, tính toàn vẹn trang web, phát hiện thay đổi trang web*

1. Mở đầu

Một trong những kiểu tấn công được biết rộng rãi nhất là tấn công thay đổi nội dung, giao diện của website [1]. Hình thức tấn công này thường sử dụng các mã độc (virus, worm, trojan, và các loại mã độc khác), để xóa bỏ, sửa đổi, hoặc thay thế nội dung các trang web trên host (web server) [2].

Lỗ hổng website là mục tiêu tiềm tàng của việc tấn công (hack) vì các mục đích khác nhau. Các hacker có các công cụ để tìm kiếm các lỗ hổng website một cách sâu rộng và nhanh chóng, tiếp theo là chúng sẽ tiến hành khai thác những điểm yếu đó [3-4].

Những cuộc tấn công thay đổi website đã được thực hiện để xâm phạm tính toàn vẹn của web bằng một trong những hình thức sau [1]:

- Thay đổi nội dung của trang web.
- Thay đổi bất kỳ phần nào của nội dung trang web.
- Thay thế toàn bộ trang web.
- Chuyển hướng trang web.
- Phá hủy hoặc xóa bỏ trang web.

Các hệ thống kiểm soát an ninh mạng như Firewall, VPN (Virtual Private Network), PKI (Public Key Infrastructure)… là những công cụ quan trọng để giữ cho web được an toàn hơn,

[1]Trường Đại học Công nghiệp Thực phẩm TP. Hồ Chí Minh
Email: tottd@cntp.edu.vn

nhưng chúng không đủ để đảm bảo an ninh website, do đó cần những cơ chế an ninh tốt hơn [1].

Có nhiều phương pháp được đề xuất để bảo vệ trang web chống lại các cuộc tấn công như; Integrit [5], Veracity [6], Aide [7], L5 [8], Tripwire [9]. Tuy nhiên các phương pháp này cũng có nhiều nhược điểm cụ thể như sau:

- Integrit, Veracity, Aide và L5 không có phương án tự bảo vệ mình khi bản thân bị tấn công.

- Tripwire thiếu sự cảnh báo nếu quá trình kiểm tra của nó bị thất bại bởi kẻ tấn công.

- Các hệ thống nêu trên đều bị dừng lại và không có giá trị bảo mật nếu bộ phận kiểm tra bị thất bại vì bất kỳ lý do nào.

Những hạn chế của các hệ thống nêu trên là động lực thúc đẩy chúng tôi nghiên cứu phương pháp mới kết hợp sử dụng hàm băm và thuật toán HTML Diff để tìm sự thay đổi nội dung dựa trên sự khác biệt giữa hai trang HTML của cùng một trang web tại hai thời điểm khác nhau. Từ đó 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. Đặc biệt, hệ thống này đã khắc phục được tối đa những hạn chế đã được đề cập ở trên.

Phần còn lại của bài báo được tổ chức như sau: phần 2 trình bày các kiến thức cơ sở, phần 3 đề xuất phương pháp phát hiện sự thay đổi trong tập tin HTML, phần 4 trình bày các kết quả thực nghiệm khi triển khai hệ thống. Phần 5 là kết luận và hướng nghiên cứu tiếp theo.

2. Kiến thức cơ sở

2.1. Phân loại sự thay đổi

Hiện nay, với sự phát triển mạnh mẽ của công nghệ gần như tất cả các tổ chức, doanh nghiệp, các nhân đều sử dụng Website để quảng bá thông tin, sản phẩm của mình. Tuy nhiên vấn đề an toàn cũng trở nên hết sức cấp thiết "các cuộc tấn công vào website Việt Nam trong ba tháng đầu năm 2017 gồm 2.853 trang bị tấn công Deface (thay đổi giao diện), 3.783 trang bị cài Malware (mã độc) và 1.050 website bị đặt mã Phishing (lừa đảo)" theo VNCERT[1].

Vì vậy để giám sát và phát hiện các đấu hiệu bất thường trên website để cảnh báo kịp thời thì việc cần làm là tìm ra các dấu hiệu để nhận biết được các thay đổi này theo nhóm tác giả phân tích thì sự thay đổi của một trang web có thể chia làm 4 loại phổ biến như sau: Thay đổi về cấu trúc, thay đổi về nội

[1] http://nhipsongso.tuoitre.vn/bao-mat/quy-12017-gan-7700-cuoc-tan-cong-mang-viet-nam-1284710.htm (Truy cập ngày 8/8/2017).

dung, thay đổi về hình thức, định dạng và thay đổi về hành vi.

Thay đổi về cấu trúc: Các hành động thêm, xóa, hoặc chỉnh sửa một thẻ trong văn bản HTML chính là đang thay đổi cấu trúc của một trang web. Việc phát hiện tự động những thay đổi về cấu trúc này rất quan trọng vì các cấu trúc của trang web khi thay đổi rất khó để có thể phát hiện một cách trực quan.

```
<html>
<head><title> Trang chủ khoa CNTT
 </title></head>
<body>
<div> …. <div>
</body>
</htm>
```

```
<html>
<head><title> Trang chủ khoa CNTT
 </title></head>
<body>
<div><font> …. </font><div>
<div> <B>…</B></div>
</body>
```

Hình 1: *a) HTML ban đầu* *b) HTML đã chỉnh sửa*

Thay đổi về nội dung và ngữ nghĩa: là những thay đổi từ cách nhìn của người sử dụng. Ví dụ, việc thay đổi về giá vàng và ngoại tệ trên các trang sàn giao dịch hay trên các trang của ngân hàng sẽ rất thu hút sự quan tâm của những nhà đầu tư và kinh doanh. Một ví dụ khác là sự thay đổi của các trang cập nhật tỷ số bóng đá online như livescore.com, người dùng rất quan tâm tới tỷ số hiện tại và sự thay đổi tỷ số giữa các trận đấu.

```
<html>
<head><title> Livescores.com
 </title></head>
<body>
<table>
<tr> <td> Doi A</td><td> 2 </td></tr>
<tr> <td> Doi A</td><td> 2 </td></tr>
</body>
</htm>
```

```
<html>
<head><title> Livescores.com
 </title></head>
<body>
<table>
<tr> <td> Doi A</td><td> 3 </td></tr>
<tr> <td> Doi A</td><td> 2 </td></tr>
</body>
</htm>
```

Hình 2: *a) HTML ban đầu* *b) HTML đã chỉnh sửa*

Thay đổi về hình thức và định dạng: là thay đổi về cách thức thể hiện nhưng vẫn giữ nguyên nội dung của trang web. Ví dụ một trang web có thể thay đổi về tính chất các thẻ định dạng nhưng không có sự thay đổi nào về nội dung.

```
<html>
<head><title> Livescores.com
 </title></head>
<body >
<table border=1>
<tr> <td> Doi A</td><td> 2 </td></tr>
<tr> <td> Doi A</td><td> 2 </td></tr>
</body>
</htm>
```

```
<html>
<head><title> Livescores.com
 </title></head>
<body backgroundcolor= ”red”>
<table>
<tr> <td> Doi A</td><td> 3 </td></tr>
<tr> <td> Doi A</td><td> 2 </td></tr>
</body>
</htm>
```

Hình 3: *a) HTML ban đầu* *b) HTML đã chỉnh sửa*

Thay đổi về hành vi: Một trang web có thể chứa nhiều đoạn kịch bản (scripts), applet là các thành phần hoạt động của trang web đó. Khi một trong các thành phần đang được ẩn giấu bị thay đổi thì dẫn đến hành vi của trang web đó cũng thay đổi theo. Tuy nhiên những thay đổi này rất khó phát hiện, đặc biệt là các thành phần hoạt động lại nằm trong một file khác.

2.2. Thuận toán HTML Diff

Thuật toán HTML Diff là thuật toán dùng để so sánh 2 tập tin HTML và xác định sự thay đổi của tập tin theo từng từ.

- **Input**: 02 tập tin HTML, Text01 và Text02

- **Output**: 01 tập tin được tạo thành từ tập tin 01 và chỉ rõ sự thay đổi của tập tin so với tập tin 2:

Các bước thực hiện của thuật toán:

- B1: Tách file thành danh sách các từ

OneWords=TachTu(Text01)

TwoWords=TachTu(Text02)

- B2: Đánh chỉ số cho các từ trong TwoWords lưu trong **wordIndices**
- B3: Với mỗi từ word trong OneWord
  - Tìm kiếm và xác định vị trí trong **wordIndices**
  - Xác định loại thay đổi
  - Chèn vào trong danh sách thay đổi
- B4: Hiển thị các thay đổi.

Giả sử ta có 2 file HTML, Text01 và Text02 có nội dung như sau:

```
<p><i>Đây là </i> ví dụ minh họa <strong> mô tả</strong> kiểm tra sự thay đổi nội dung bằng <strong>thuật toán Diff</strong>.</p>
<p>Ngôn ngữ sử dụng <b>C Sharp</b> trên hệ điều hành windows  của khoa CNTT
<a href='http://fit.hufi.edu.vn'>tại đây</a></p>
<table cellpadding='1' cellspacing='1' border=’1’>
<tr><td>Nội dung minh họa</td><td>Giá trị minh họa</td></tr>
<tr><td>Dữ liệu thử (this row will be removed)</td><td>Dữ liệu thật</td></tr>
</table>
Số lượng giảng viên trong khoa là 35"
```

Hình 4: *Nội dụng HTML của Text01*

Có hiển thị trên website như sau:

Đây là ví dụ minh họa **mô tả** kiểm tra sự thay đổi nội dung bằng **thuật toán Diff**.

Ngôn ngữ sử dụng **C Sharp** trên hệ điều hành windows của khoa CNTT tại đây

Nội dung minh họa	Giá trị minh họa
Dữ liệu thử (this row will be removed)	Dữ liệu thật

Số lượng giảng viên trong khoa là 35"

Hình 5: *Nội dung hiển thị của Text01*

```
<p><i>Đây là </i> ví dụ minh họa <strong> mô tả</strong> đánh giá kiểm tra sự thay đổi nội dung bằng <strong>thuật toán Diff</strong>.</p>
<p> Đây là dòng dữ liệu thêm mới</p>
<p>Ngôn ngữ sử dụng <b>C Sharp</b> trên hệ điều hành windows  của khoa CNTT
<a href='http://fit.hufi.edu.vn'>tại đây</a></p>
<table cellpadding='1' cellspacing='1' border='1'>
<tr><td>Nội dung minh họa <b>mới</b></td><td>Giá trị minh họa</td></tr>
</table>
Số lượng giảng viên trong khoa là 35"
```

Hình 6: *Nội dụng HTML của Text02*

Có hiển thị trên website như sau:

Đây là ví dụ minh họa **mô tả** đánh giá kiểm tra sự thay đổi nội dung bằng **thuật toán Diff**.

Đây là dòng dữ liệu thêm mới

Ngôn ngữ sử dụng **C Sharp** trên hệ điều hành windows của khoa CNTT tại đây

Nội dung minh họa **mới**	Giá trị minh họa

Số lượng giảng viên trong khoa là 35"

Hình 7: *Nội dung hiển thị của Text02*

Áp dụng thuật toán với đầu vào là hai tập tin Text01 và Text02 như hình 2 và 3

Bước 1: Thuật toán sẽ tách từ các văn bản trên thành danh sách các từ. Với Text01 ta có danh sách các từ **oneWords** như sau

i	1	2	3	4	5	6	7	8	…
Từ	<p>		Đây	Là		ví	dụ	minh	…

Tương tự vậy ta có danh sách các từ của Text02 là **twoWords**

Bước 2: Chương trình sẽ đánh chỉ mục cho các cho các từ đã của Text02 mà đã được tách trong B1 như sau và lưu trong **wordIndices**

i	1	2	3	4	5	6	7	…
Từ	<p>		Đây	là		ví	dụ	…
Vị trí	0, 53,70	1	2, 55	4, 57, 149	6	8	10	

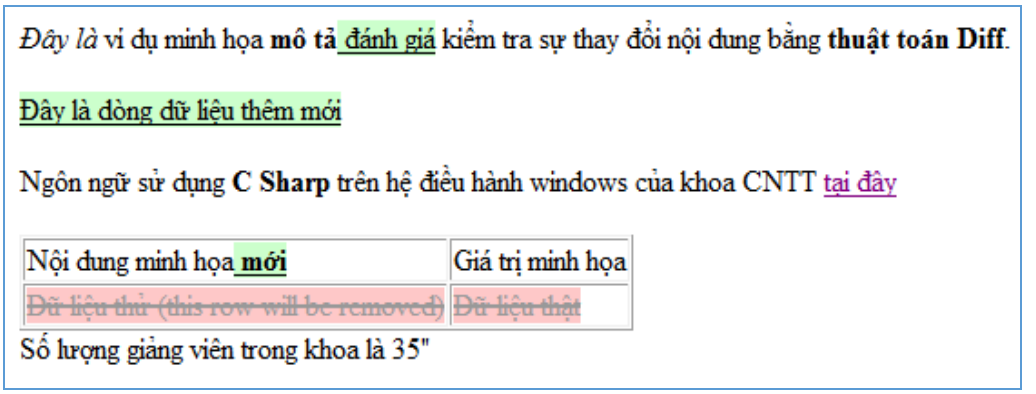
Bước 3: Thuật toán tiến hành so khớp. Trong khi so khớp thuật toán chia làm 3 thao tác là so sánh bằng, thêm và xóa. Và với mỗi ký tự so khớp thuật toán chia làm 3 loại cần so khớp là khoảng trắng, ký tự đóng mở thẻ và ký tự khác (whitespace, tag, character).

- Khởi tạo danh sách lưu kết quả so sánh content=null
- Với mỗi từ item trong oneWords đã xác định trong bước 1, thuật toán dựa vào **wordIndices** xác định xem item đó xuất hiện ở vị trí nào trong twoWord

Kết quả thực hiện:

- Nếu tìm thấy và đúng vị trí thì là gán nhãn bằng: có nghĩa là không thay đổi
- Nếu không bằng: Gán nhãn xóa bằng cách thêm vào từ kiểm tra cặp thẻ <del> </del> rồi kiểm tra xem trong tài liệu 02 vị trí đó được thay thế bằng từ khác hoặc ký tự khác thì gán nhãn cho từ đó là thêm mới và chèn thêm cặp thẻ <ins> </ins>
- Cập nhật kết quả so sánh vào content

Bước 4: Hiển thị nội dung trong content cho người sử dụng



Hình 8: *Kết quả thực hiện*

2.3. Các hàm băm thông dụng

Các hàm băm dòng MD (MD2, MD4, MD5) do Giáo sư Ronald L. Rivest đề xuất. Giá trị băm theo các thuật toán này có độ dài cố định là 128bit.

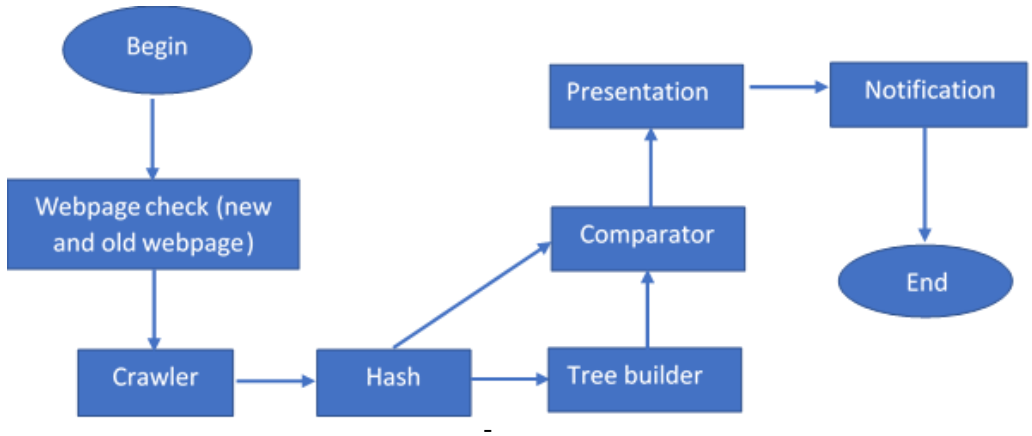
Phương pháp Secure Hash Standard (SHS) gồm tập hợp các thuật toán băm mật mã an toàn (Secure Hash Algorithm – SHA) như SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 do NIST và NSA xây dựng. Hàm băm an toàn SHA phức tạp hơn nhiều cũng dựa trên các phương pháp tương tự, được công bố trong Hồ sơ Liên bang năm 1992 và được chấp nhận làm tiêu chuẩn năm 1993. Giá trị băm theo thuật toán này có độ dài cố định là 160 bit. Ngoài ra còn có một số thuật toán khác như: RIPEMD, HAVAL, Whirlpool, Tiger.

Mỗi hàm băm đều có những ưu điểm và nhược điểm riêng của mình. Tất cả các hàm băm trên đều có tính bảo mật cao, trong đó họ hàm băm SHA được coi là có tính bảo mật cao nhất. Tuy nhiên xét về tốc độ mã hóa, MD5 là hàm băm có tốc độ mã hóa cao nhất trong các hàm băm trên [13] do đó trong phương pháp đề nghị ở bài báo này chúng tôi quyết định chọn MD5.

3. Phương pháp đề xuất

Sau khi nghiên cứu chúng tôi đề xuất một phương pháp mới cho phép phát hiện sự thay đổi giao diện, nội dung của website. Phương pháp này được phát triển dựa trên thuật toán HTML Diff kết hợp với hàm băm MD5. Các thay đổi như chèn thêm nội dung mới, xóa hay sửa nội dung cũ, thay đổi các định dạng về màu sắc, kích thước, kiểu chữ của nội dung sẽ được ứng dụng ngay lập tức ghi nhận và thông báo tới người quản trị website.

3.1. Phương pháp phát hiện thay đổi nội dung



Hình 9: *Mô hình giải pháp*

Phương pháp giải quyết bài toán được mô hình hóa trong hình 9. Chi tiết được trình bày như sau:

Input: url: địa chỉ webpage cần kiểm tra

Output: cảnh báo nếu phát hiện vấn đề bất thường.

Các bước thực hiện

Bước 1: Hệ thống nhận dữ liệu đầu vào là webpage cần được so khớp (Webpage check (new and old webpage)).

Bước 2: Từ đó module **Crawler** sẽ thu thập dữ liệu của webpage liên tục theo định kỳ do người dùng xác lập sẵn.

Bước 3: Từ kết quả Crawler lấy về, thay vì sử dụng trực tiếp thuật toán HTML Diff để so sánh ngay và tìm ra sự thay đổi của webpage, phương pháp được đề xuất sẽ sử dụng thuật toán MD5 (**Hash**) để băm kết quả thành một chuỗi để lưu trữ và so sánh với kết quả băm trước đó. Nếu kết quả băm ở thời điểm hiện tại có sự thay đổi so với kết quả băm đã lưu trữ trước đó sẽ đồng nghĩa với việc webpage đã thay đổi về nội dung.

Lúc này hệ thống mới tiến hành áp dụng thuật toán HTML Diff để so khớp (**Comparator**) cụ thể trên toàn bộ webpage.

Để làm tăng tốc độ của phương pháp đề xuất, hệ thống bổ sung thêm module **Tree builder** nhằm chia nhỏ webpage thành nhiều phần theo cấu trúc HTML của cây trước khi so sánh. Vì vậy, khi phát hiện webpage bị thay đổi ở phần cấu trúc nào, hệ thống sẽ chỉ so khớp phần nội dung của cấu trúc đó bằng HTML Diff thay vì so khớp toàn bộ webpage.

Bước 4: Sau khi có kết quả so sánh, kết quả sẽ được lưu trữ, hiển thị (**Presentation**) cho người sử dụng để quan sát và gửi cảnh báo (**Notification**) thông qua email hay SMS.

Như vậy, trong phương pháp này hệ thống sẽ giảm được thời gian vì đã không cần phải luôn so khớp bằng HTML Diff ở mọi giai đoạn của

phương pháp mà chỉ áp dụng khi chắc chắn webpage có sự thay đổi. Thêm vào đó, cấu trúc cây sẽ giới hạn phạm vi thay đổi của webpage nên khi áp dụng HTML Diff sẽ hiệu quả hơn.

3.2. Phương pháp phát hiện thay đổi về hình ảnh

Kiểm tra sự thay đổi về hình ảnh bằng cách lấy dữ liệu HTML Document về. Để biết được sự thay đổi về hình ảnh trong web so với lần lấy hình ảnh trước đó, ta chỉ cần kiểm tra tập hình ảnh mới lấy A_{IMG_new} so với tập hình ảnh đã lấy trước đó A_{IMG_old}. Nếu A_{IMG_new} có tổng số lượng hình ảnh $T_{new} = 0$, T_{new} giảm hoặc tăng đột ngột so với tổng số lượng hình ảnh của A_{IMG_old}, có sự thay đổi về nội dung hình ảnh (nghĩa là cùng một đường dẫn tới tài nguyên hình ảnh nhưng lại có hàm băm MD5 khác nhau, trong đó hàm băm MD5 chính là sự thể hiện của nội dung hình ảnh) thì đó là một sự thay đổi bất thường và được cho là cần cảnh báo cho người quản trị website.

Thuật toán kiểm tra sự thay đổi hình ảnh được trình bày như sau:

Input:

- url: địa chỉ webpage chứa nội dung hình ảnh cần thu thập
- listIMGOld: danh sách các hình ảnh từ lần lấy hình ảnh trước đó

Output: Thực hiện lưu lại danh sách hình ảnh, đồng thời trả về giá trị ngưỡng. Nếu ngưỡng = 0 thực hiện cảnh báo ngay và thoát ra. Nếu ngưỡng là một giá trị > 0 nghĩa là các tiêu chí phát hiện không nghiêm trọng.

Các bước thực hiện chính:

CheckImage(url, listIMGOld)

Bước 1: Lấy tài liệu HTML của webpage

HtmlDocument ← GetHtml(url);

Bước 2: Lấy tất cả đường dẫn tới tài nguyên hình ảnh và so sánh

```
Foreach(src in GetAllSrcImg (htmlDocument))
{
   md5 ← DownloadIMG(src)
   total++;
   if(src != listIMGOld.src)
         listIMGNew.add(src)
   else
         if(md5 != listIMGOld.md5)
               listIMGChange.add(src)
}
```

Trong đó các hàm và thuộc tính được mô tả như sau:

- **GetHtml(url)**: có chức năng gửi một request, nhận và trả về tài liệu HTML tương ứng với địa chỉ url.
- **GetAllSrcImg(htmlDocument)**: có chức năng lấy tất cả các thuộc tính src của tất cả các thẻ <img> trong tài liệu HTML đã tải về.
- **DownloadIMG(src)**: tải về hình ảnh với đường dẫn src và chuyển hình ảnh sang md5
- **total**: tổng số lượng hình ảnh đếm được
- **listIMGNew**: danh sách các hình ảnh mới thêm vào trong lần kiểm tra này
- **listIMGChange**: danh sách các hình ảnh đã bị thay đổi nội dung hình ảnh

Bước 3: Xem xét sự thay đổi nằm trong ngưỡng nào

```
if(total == 0)
        return 0;
if(listIMGChange.count > 0)
        return 0;
if(totalOdl/3 > total)
totalValue ← value;
if(listIMGNew.count > total/3)
        totalValue ← value;
```

Có 2 mức độ nguy hiểm đó là tất cả các hình ảnh bị mất hết (total == 0) và danh sách những hình ảnh bị thay đổi nội dung > 0. Còn những mức độ còn lại sẽ cộng dồn ngưỡng giá trị.

Nếu trả về giá trị 0 nghĩa là đang ở mức độ nguy hiểm, cần cảnh báo ngay. Nếu không trả về không thì sẽ lấy giá trị totalValue làm giá trị ngưỡng cho lần kiểm tra này.

Bước 4: Sau đó lưu lại tất cả các hình ảnh xuống cơ sở dữ liệu dưới dạng file xml gồm 2 thuộc tính quan trọng là src và MD5 của hình ảnh.

Với phương pháp phát hiện thay đổi về hình ảnh được đề xuất nó giúp gia tăng độ chính xác của cảnh báo, cụ thể đối với một webpage thì sẽ có rất nhiều hình ảnh đính kèm khi bị tấn công có thể các hình ảnh này sẽ bị thay đổi hoặc xóa hết, và thông thường các hình ảnh này chỉ có tăng chứ không giảm, nên khi bị giảm hoặc thay đổi là đã bất thường cần phải cảnh báo ngay.

Tóm lại với các đề xuất trên đã giúp hệ thống cảnh báo có thể làm việc hiệu quả và giúp quản trị viên website ứng cứu kịp thời khi có các sự cố không mong muốn.

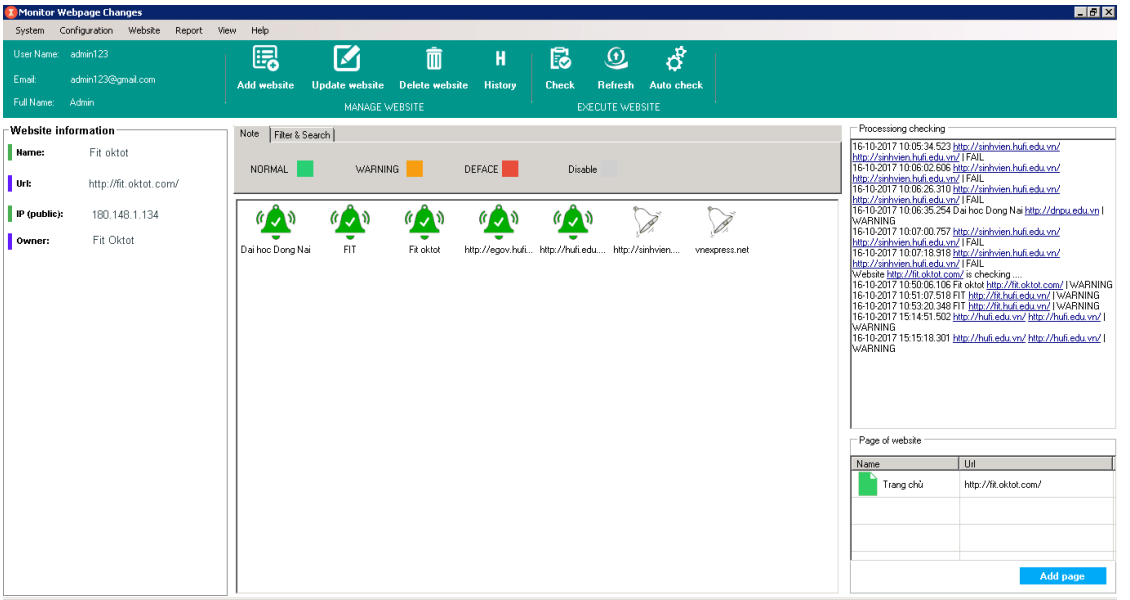
4. Kết quả thực nghiệm và thảo luận

Ứng dụng thực nghiệm "Monitor Webpage Changes" được phát triển bằng ngôn ngữ C# (Microsoft Visual Studio 2010). Với cấu hình máy sử dụng là:

- Bộ xử lý: Intel(R) Core(TM) i5 CPU M450 @ 2.40GHz
- Bộ nhớ Ram: 8.00 GB.
- Loại hệ thống: hệ điều hành 64-bit.
- Hệ điều hành: Windows 10 Professional.

Giao diện chính của "Monitor Webpage Changes"; biểu tượng cái chuông sẽ thay đổi màu theo tình trạng của website được giám sát: màu đỏ là website đang bị nguy hiểm, màu xanh là

chưa phát hiện bất thường, màu vàng là có dấu hiệu nguy hiểm cần được quản trị viên website kiểm tra.



Hình 10: *Giao diện chính của ứng dụng "Monitor Webpage Changes"*

4.1. Danh sách các trường hợp thử nghiệm

STT	Ngày	Địa chỉ website thử nghiệm
1	16/7/2017 – 19/8/2017	http://hufi.edu.vn
2	16/7/2017 – 6/10/2017	http://fit.oktot.com
3	22/8/2017 – 15/9/2017	http://dnpu.edu.vn

Bảng tổng hợp kết quả:

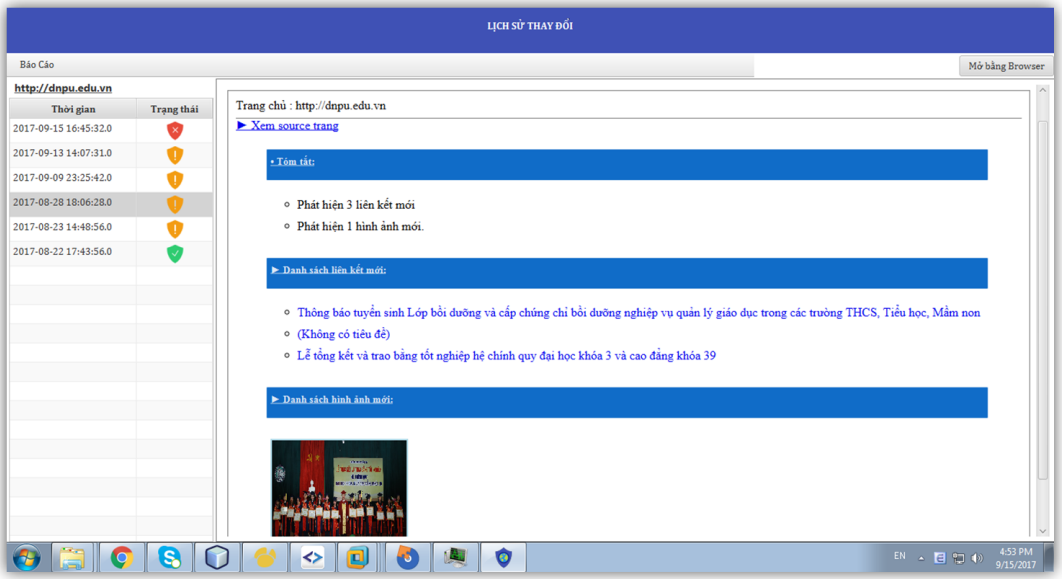
STT	Kịch bản thử nghiệm	Số lần thử nghiệm	Số lần đạt yêu cầu	Số lần không đạt yêu cầu	Tỷ lệ thành công
1	Thêm text trên home page	50	50	00	100%
2	Xóa text trên home page	50	50	00	100%
3	Thay đổi text trên home page	50	50	00	100%
4	Thêm hình ảnh trên home page	50	50	00	100%
5	Xóa hình ảnh trên home page	50	50	00	100%
6	Sửa hình ảnh trên home page	50	50	00	100%
7	Thêm link trên home page	50	50	00	100%

8	Xóa link trên home page	50	50	00	100%
9	Sửa link trên home page	50	50	00	100%

4.2. Một số hình ảnh chụp kết quả giám sát theo thời gian thực trên website http://dnpu.edu.vn

Ngày 22/8/2017 khởi tạo giám sát website http://dnpu.edu.vn

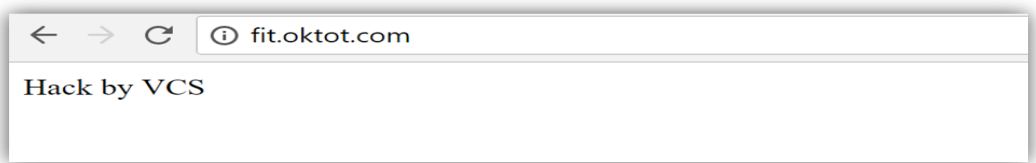
Ngày 23/8/2017, 28/8/2017 hệ thống phát hiện có liên kết mới được thêm vào, hệ thống đã ghi nhận lại kết quả và gởi mail cảnh báo.



Hình 11: *Ảnh chụp màn hình cảnh báo của ứng dụng "Monitor Webpage Changes"*

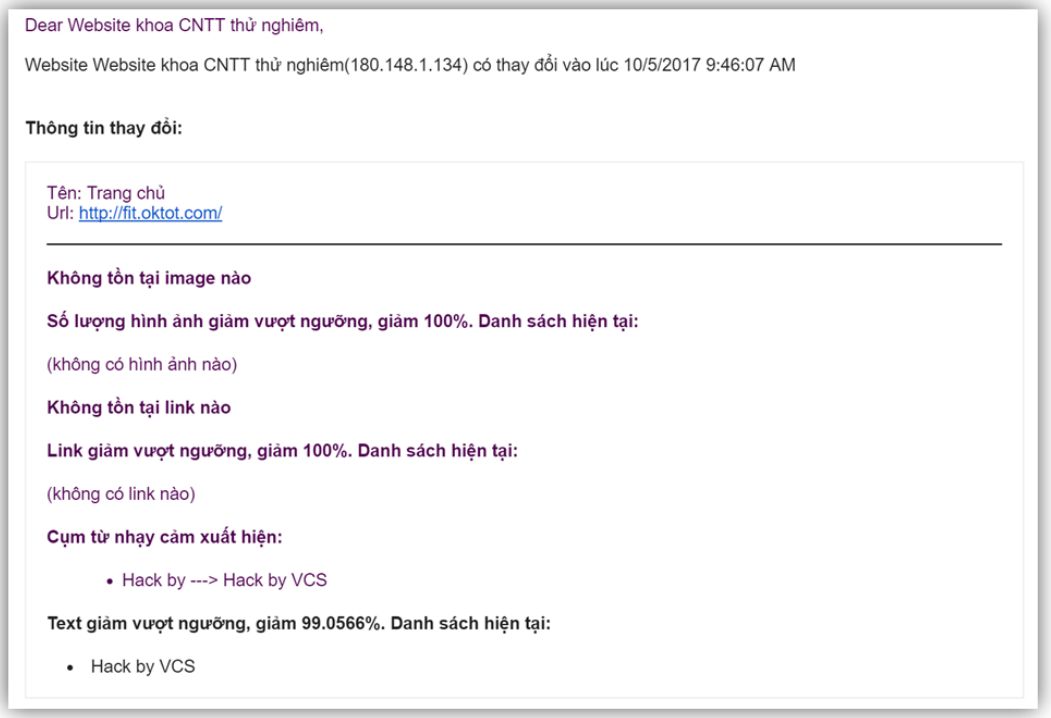
4.3. Một số hình ảnh chụp kết quả giám sát theo thời gian thực trên website http://fit.oktot.com (bản sao của website fit.hufi.edu.vn)

Website bị tấn công và để lại dòng chữ "Hack by VCS"



Hình 12: *Màn hình website bị tấn công để lại thông điệp của hacker*

Thông tin cảnh báo nhận được qua mail lúc 10/5/2017 9:46:07 AM



Hình 13: *Ảnh chụp email cảnh báo khi website bị tấn công*

Ứng dụng "Monitor Webpage Changes" đã cài đặt trên một số website và giám sát thời gian thực trên các website này. Kết quả thực nghiệm cho thấy ứng dụng hoạt động tốt và đã đạt được kết quả như sau:

- Phát hiện được tất cả các thay đổi của website (trừ một số thông tin không kiểm tra là số lượng online và số người truy cập) và gửi cảnh báo cho quản trị viên mỗi khi có sự thay đổi.
- Giao diện ứng dụng khá thuận tiện và 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.
- Ứng dụng "Monitor Webpage Changes" được cài đặt độc lập với website và giám sát thông qua môi trường internet và ứng dụng cài trực tiếp trong cùng hệ thống của website đều cho kết quả giống nhau.

5. Kết luận

Trong bài báo này, chúng tôi đã trình bày một hướng tiếp cận bài toán hoàn toàn khác so với các phương pháp cũ trước đây như Integrit, Veracity, Aide, L5 và Tripwire. Phương pháp đề xuất mới này dựa trên sự kết hợp hàm băm MD5 và thuật toán HTML Diff đã mang lại kết quả rất khả quan trong việc giám sát, có thể phát hiện sự thay đổi giao diện, nội dung của website một cách nhanh chóng, kịp thời theo thời

gian thực. Qua kết quả thực nghiệm cho thấy các thay đổi như chèn thêm nội dung mới, xóa hay sửa nội dung cũ, thay đổi các định dạng về màu sắc, kích thước, kiểu chữ của nội dung đã được ứng dụng "Monitor Webpage Changes" ngay lập tức ghi nhận và thông báo tới người quản trị website. Chương trình ứng dụng cũng làm nổi bật nhưng vị trí đã thay đổi và gửi thư cảnh báo và khuyến nghị cho người quản trị trang web để xử lý.

TÀI LIỆU THAM KHẢO

1. Charles P. Pfleeger and Shari Lawrence (2003), "Security in Computing", 3rd Edition, Prentice Hall (available at http://books.google.com/books?id=O3VB-zspJo4C&dq =%22web+site+defacement+attack+%22&source=gbs_navlinks_s)

2. William Stalling (1999), "Cryptography and Network Security", Prentice Hall

3. Shar, L.K. and Hee Beng Kuan Tan (2013), "Defeating SQL Injection", in IEEE Computer, Singapore, Vol. 46, Issue: 3, pp. 68-77

4. "Chinese websites 'defaced in Anonymous attack'" (2012), [Online], Available: http://www.bbc.co.uk/news/technology-17623939, April 5, 2012

5. E.L.Cashin (2000), "Integerit file Verification System", (available at http://integrit.sourceforge.net)

6. Rocksoft (2003), "Veracity- nothing can change without you knowing: Data integrity assurance", (available at http://www.rocksoft.com/veracity/)

7. R.Lehti (2005), "Advanced Intrusion Detection Environment", (available at http://www.cs.tut.fi/arammer/aide.html)

8. RSA Laboratories (1992), "The MD2 Message Digest Algorithm"

9. Gene Kim (2001), "Advanced Applications of Tripwire for Servers", Tripwire, Inc

10. E.Berk, "HtmlDiff: A Differencing Tool for HTML Documents", Student Project, Princeton University, http://www.htmldiff.com

11. S.Chawate, A.Rajaraman, H.Garcia-Molina and J.Widom (1996), "Change Detection in Hierarchical Structured Information", Proceedings of the ACM SIGMOD International Conference on Management of Data, Monteal, June 1996

12. H. P. Khandagale and P. P. Halkarnikar (2010), "Novel Approach for Web Page Change Detection System", International Journal of Computer Theory and Engineering, Vol. 2, No. 3, June, 2010, 1793-8201

13. TS. Dương Anh Đức - ThS. Trần Minh Triết (2005), *Mã hóa và ứng dụng, Khoa Công nghệ thông tin*, Trường Đại Học Khoa học Tự nhiên, Đại Học Quốc gia TP. Hồ Chí Minh

DETECTING AND GIVING WARNINGS OF CHANGED WEBSITES BASED ON CHANGED CONTENTS AND HTML STRUCTURES

ABSTRACT

Hackers' attacks, which change the interface and contents of webpages, bring about particularly serious consequences. Therefore, there should be methods to allow real-time detection of these changes to minimize the consequences. In this article, we present a new method to detect the changes in webpage interface and contents. This method is developed based on the HTML Diff algorithm combined with the MD5 hash function, and has been built into an application with a nice, easy-to-use interface. Changes such as new contents inserted, contents deleted or edited, and changes to the format of color, size, type of content will be immediately recorded and notified to the website administrator. The application will also highlight the changed locations and send a warning message and recommendations to the webmaster.

Keywords: *Supervise changes, attack defacement, entire of website, dectection of changed websites*

(Received: 20/9/2017, Revised: 5/10/2017, Accepted for publication: 12/12/2017)