BÁO CÁO KHÓA LUẬN TỐT NGHIỆP

Đề tài:

"Nghiên cứu, xây dựng mô hình phát hiện tấn công thay đổi giao diện website dựa trên học máy"

01

Tính cấp thiết

- Tổng quan về tấn công thay đổi giao diện

02

Mục tiêu nghiên cứu

- Mục tiêu khóa luận hướng đến

03

Bài toán

- Xác định đầu vào, đầu ra của bài toán
- Áp dụng học máy

04

Mô hình đề xuất

- Phát biểu về mô hình xây dựng áp dụng kỹ thuật học máy

05

Cài đặt thử nghiệm và kết quả

- Các cài đặt
- Kết quả các thử nghiệm

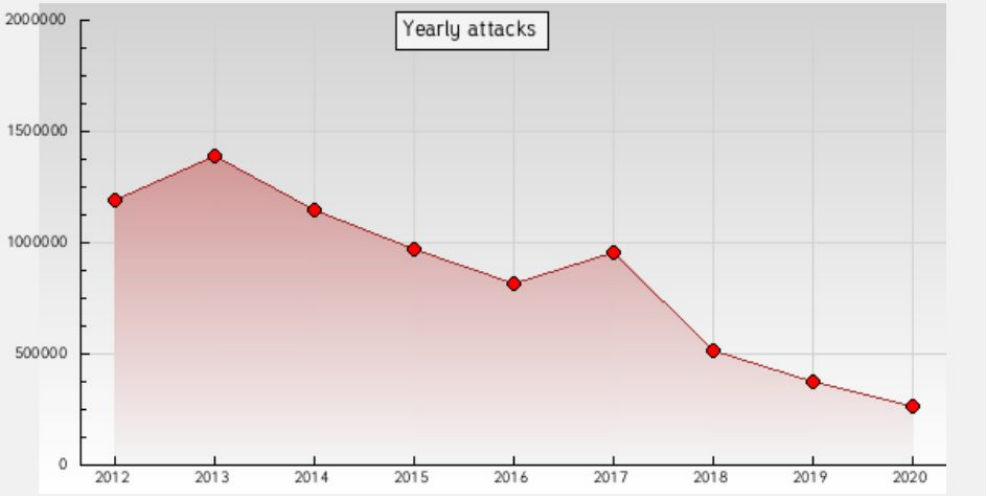
01 Tính cấp thiết

- Tấn công thay đổi giao diện: (hay Website defacement) là một loại tấn công làm thay đổi một cách trái phép nội dung của một website.

Hình ảnh các Website bị thay đổi giao diện

- Hậu quả:

- Tấn công thay đổi giao diện có trực tiếp làm cản trở hoạt động của Web và ứng dụng Web
- Bị lợi dụng để lan truyền các quan điểm về Chính trị, tôn giáo, …



Lượng tấn công thay đổi giao diện Zone-H thu thập

02 Mục tiêu nghiên cứu

1. Tìm hiểu và xây dựng mô hình phát hiện tấn công thay đổi giao diện
2. Áp dụng thuật toán học máy vào mô hình phát hiện
3. Kiểm thử và đánh giá kết quả

03 Bài toán

"**Phát hiện tấn công thay đổi giao diện** là việc kịp thời nhận ra một cuộc tấn công thay đổi giao diện đã diễn ra, kế theo đó là đưa ra những báo động và mức độ nghiêm trọng của nó." [1]

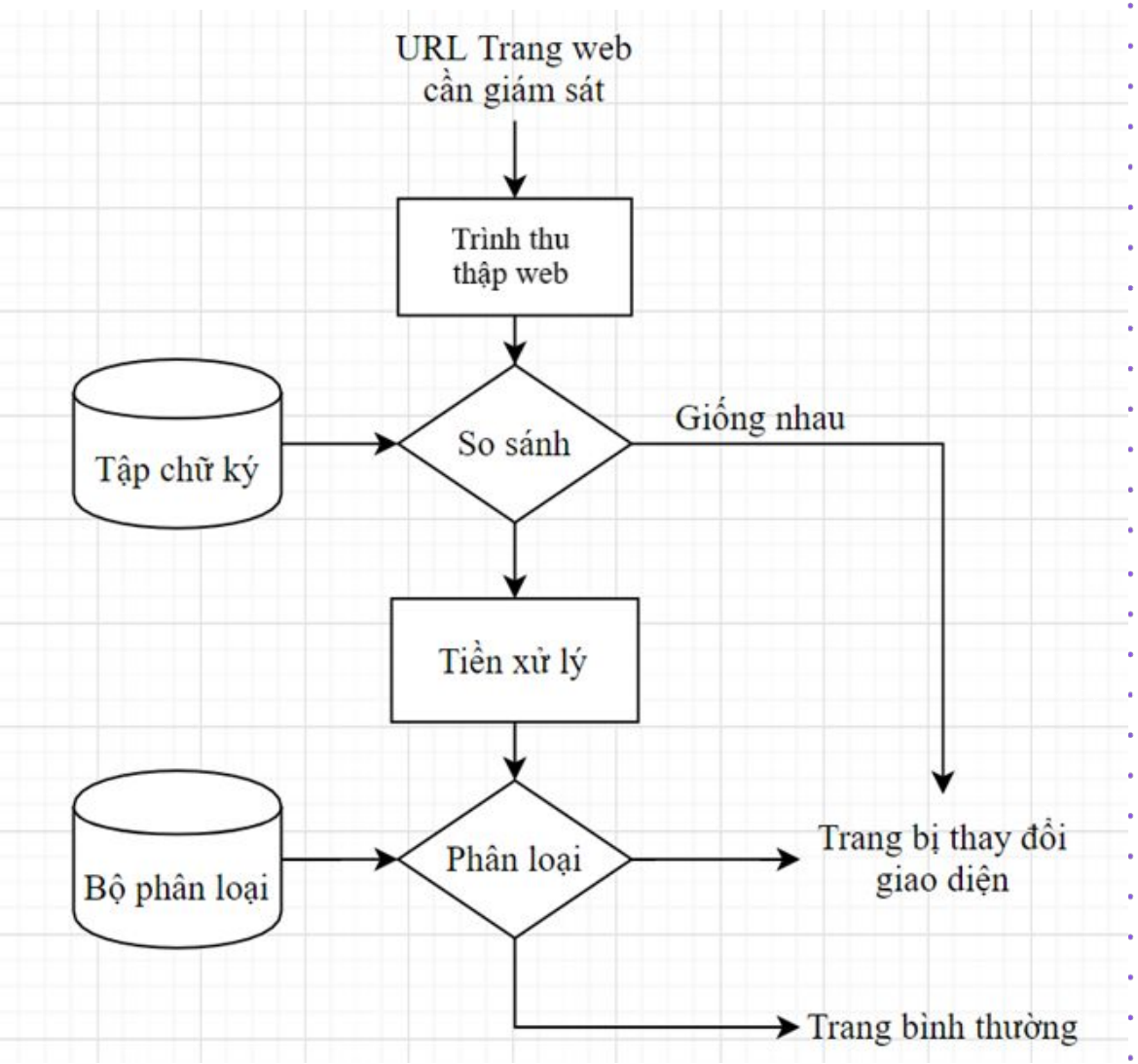
Đầu vào: Trang cần giám sát

Đầu ra: Bình thường hay đã bị tấn công

1. Mô hình áp dụng kỹ thuật?
   - Dựa trên chữ ký
   - Dựa trên bất thường
2. Một bộ phát hiện sử dụng học máy?
   - Ưu điểm
   - Dữ liệu, xử lý dữ liệu, thuật toán

04 Mô hình đề xuất

1. *Trình thu thập web:* Lấy các tài nguyên web
2. *Tập chữ ký:* Chữ ký của các cuộc tấn công
3. *Tiền xử lý:* Xử lý dữ liệu về dạng phù hợp
4. *Bộ phân loại:* Một bộ phát hiện bất thường sử dụng học máy



04 Mô hình đề xuất

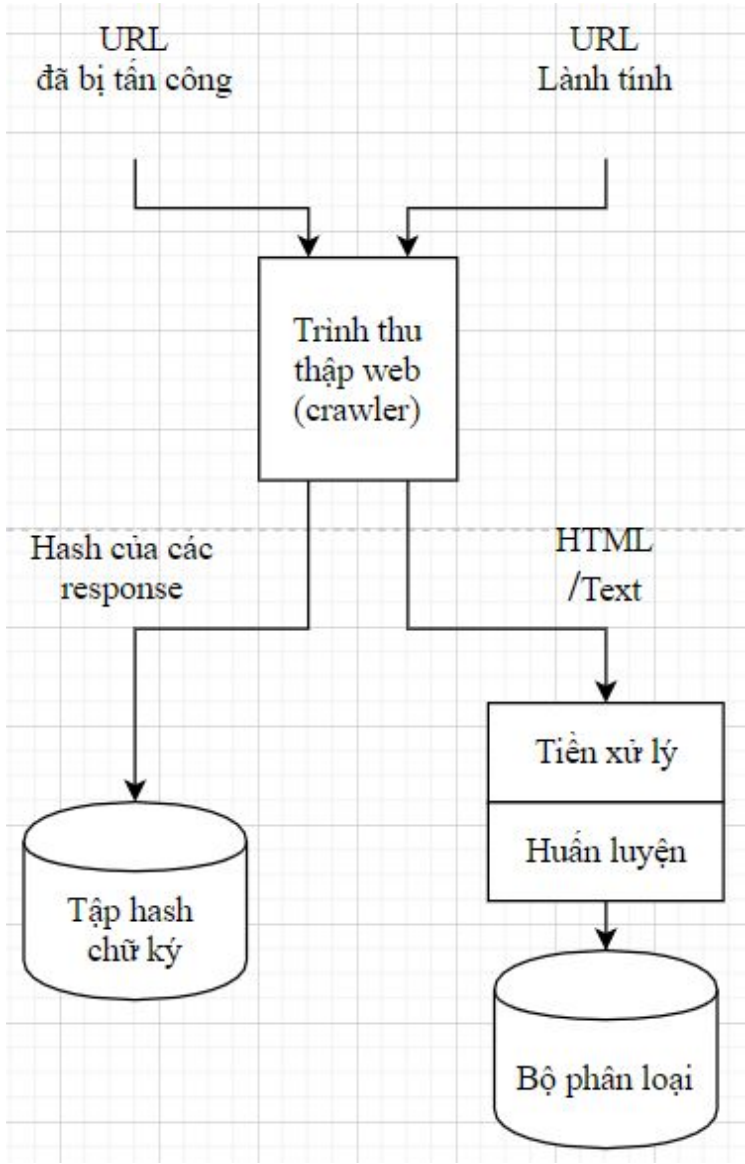
1. Xây dựng tập chữ ký:

Hash được dùng làm chữ ký
để đánh giá so sánh

2. Xây dựng bộ phân loại:

Dữ liệu mã HTML và văn bản được
sử dụng để đánh giá bất thường

- HTML gốc
- HTML xử lý động
- Dữ liệu chỉ văn bản

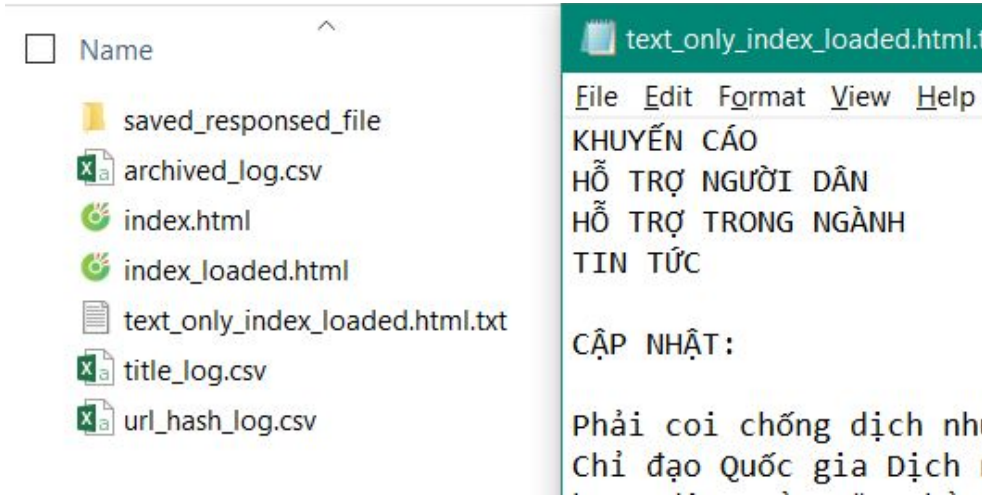


Quá trình huấn luyện

05.a Cài đặt

1. Trình thu thập Web

- Được viết trên ngôn ngữ Javascript, chạy trên môi trường Node.js và Pupeteer
- Thông qua thư viện Pupeteer, trang web được tự động xử lý trình duyệt web
- Lưu trữ các dữ liệu sử dụng trong mô hình phát hiện
  - HTML gốc *(index.html)*
  - HTML xử lý động *(index_loaded.html)*
  - Dữ liệu chỉ văn bản *(text_only…txt)*



Các dữ liệu được lưu trữ

hash	
ad8e90d81ebee0d44e4039f0738d98169f74b76f020478	text/html
f748821c6d70420563e26942ed0a4d996c7cbc9101ed40	applicatio
7dd728084ad83c867cc7678addc327b33e15e5981ee7b2	text/html
7d24bfed3ca9bcb135a97c224c861df039e6ce96de49bb	applicatio
787a75d60cbde93e64584708a0b589894fafc7a8245dc4	text/javas
e83e127ee1c4c1b3250f3148f8f48977834f97eafa5811	text/javas
7dd728084ad83c867cc7678addc327b33e15e5981ee7b2	text/html
ec4d7e5400a857d3c3c0406b3f90fa625175eb3e292b9e	applicatio
7dd728084ad83c867cc7678addc327b33e15e5981ee7b2	text/html

Tập các hash

05.a Cài đặt

2. Bộ phát hiện dựa trên bất thường

Tiền xử lý:

- N-gram: Phân tách văn bản thành các chuỗi n ký tự liên tục:
  - Khóa luận sử dụng 2-gram và 3-gram
- Tần suất xuất hiện tự (TF – Term Frequency): Sử dụng để vector hóa văn bản HTML thu được.
  - Các văn bản sẽ được biểu diễn bằng tần suất đại diện tf của 300 n-gram (Kim cùng cộng sự, 2016) xuất hiện nhiều nhất, với công thức:

$$tf(t, d) = \frac{f(t, d)}{\max\{f(w, d) : w \in d\}}$$

  - Với $f(t, d)$ là số lần xuất hiện của t trong d.
  - $tf(t, d)$ luôn nằm trong khoảng $[0,1]$

05.a Cài đặt

2. Bộ phát hiện dựa trên bất thường

- Mô hình thử nghiệm với hai thuật toán học máy là thuật toán Naive Bayes, và thuật toán Random Forest
- Quá trình tiền xử lý và huấn luyện được lập trình trên ngôn ngữ Python, các thuật toán học máy được hỗ trợ bởi thư viện scikit-learn.

05.b Thử nghiệm và kết quả

1. Dữ liệu và kịch bản thử nghiệm

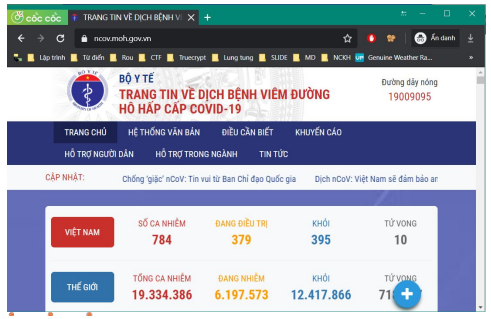
Zone-H:

- Nguồn lưu trữ công khai các trang web bị tấn công thay đổi giao diện
- Các trang sử dụng để đánh giá được lựa chọn ngẫu nhiên không trùng lặp (*)

Các trang web công khai:

- Bao gồm cả trang tĩnh và động
- Sử dụng các trang nổi tiếng làm tập bình thường

05.b Thử nghiệm và kết quả

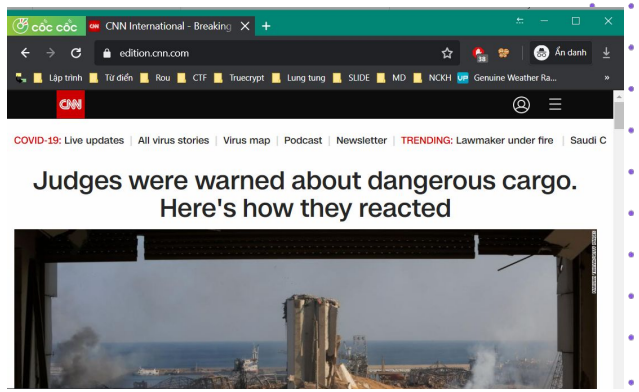


Các cổng thông tin chính phủ và các trang web phổ biến tại Việt Nam

Thử nghiệm 1

200 trang web tiếng Việt
200 trang web bị tấn công

Bổ sung thêm 200 bản ghi với nhiều ngôn ngữ



Thử nghiệm 2

400 trang bình thường
400 trang bị tấn công

Thử nghiệm 3

800 trang bình thường
800 trang bị tấn công

Thử nghiệm 4

Sử dụng bộ phát hiện từ thử nghiệm 3 trên bộ dữ liệu chuẩn (*)

(*) Lấy từ http://www.di.unito.it/~fpb/KeyedLearningData/DataSection5.2/PAGES.zip (F.Bergadano cùng cộng sự, 2019)

05.b Thử nghiệm và kết quả

2. Kết quả

algorithm	name	n-gram	percision	recall	accuracy	fail alarm	TP	FP	TN	FN
RandomForest	index.html	2	96.84%	100.00%	98.50%	2.78%	92	3	105	0
NaiveBayes	index.html	2	94.62%	95.65%	95.50%	4.63%	88	5	103	4
RandomForest	index.html	3	95.74%	100.00%	97.98%	3.70%	90	4	104	0
NaiveBayes	index.html	3	95.51%	94.44%	95.45%	3.70%	85	4	104	5
RandomForest	index_loaded.html	2	98.94%	100.00%	99.50%	0.93%	93	1	106	0
NaiveBayes	index_loaded.html	2	98.86%	93.55%	96.50%	0.93%	87	1	106	6
RandomForest	index_loaded.html	3	98.94%	100.00%	99.50%	0.93%	93	1	106	0
NaiveBayes	index_loaded.html	3	97.65%	89.25%	94.00%	1.87%	83	2	105	10
RandomForest	text_only_index_loaded.html.txt	2	97.87%	100.00%	99.00%	1.85%	92	2	106	0
NaiveBayes	text_only_index_loaded.html.txt	2	98.92%	100.00%	99.50%	0.93%	92	1	107	0
RandomForest	text_only_index_loaded.html.txt	3	97.73%	100.00%	98.96%	1.89%	86	2	104	0
NaiveBayes	text_only_index_loaded.html.txt	3	97.73%	100.00%	98.96%	1.89%	86	2	104	0

Kết quả kịch bản thử nghiệm 1

05.b Thử nghiệm và kết quả

2. Kết quả

algorithm	name	n-gram	percision	recall	accuracy	fail alarm	TP	FP	TN	FN
RandomForest	index.html	2	96.34%	98.40%	97.46%	3.38%	184	7	200	3
NaiveBayes	index.html	2	97.24%	94.12%	95.94%	2.42%	176	5	202	11
RandomForest	index.html	3	96.83%	97.86%	97.46%	2.90%	183	6	201	4
NaiveBayes	index.html	3	96.09%	91.98%	94.42%	3.38%	172	7	200	15
RandomForest	index_loaded.html	2	96.97%	98.97%	98.00%	2.91%	192	6	200	2
NaiveBayes	index_loaded.html	2	95.94%	97.42%	96.75%	3.88%	189	8	198	5
RandomForest	index_loaded.html	3	96.91%	96.91%	97.00%	2.91%	188	6	200	6
NaiveBayes	index_loaded.html	3	96.84%	78.87%	88.50%	2.43%	153	5	201	41
RandomForest	text_only_index_loaded.html.txt	2	97.47%	99.48%	98.50%	2.43%	193	5	201	1
NaiveBayes	text_only_index_loaded.html.txt	2	93.60%	97.94%	95.75%	6.31%	190	13	193	4
RandomForest	text_only_index_loaded.html.txt	3	96.04%	100.00%	98.00%	3.88%	194	8	198	0
NaiveBayes	text_only_index_loaded.html.txt	3	94.05%	89.69%	92.25%	5.34%	174	11	195	20

Kết quả kịch bản thử nghiệm 2

05.b Thử nghiệm và kết quả

2. Kết quả

algorithm	name	n-gram	percision	recall	accuracy	fail alarm	TP	FP	TN	FN
RandomForest	index.html	2	97.87%	97.87%	97.98%	1.92%	367	8	408	8
NaiveBayes	index.html	2	99.40%	88.27%	94.18%	0.48%	331	2	414	44
RandomForest	index.html	3	98.14%	98.40%	98.36%	1.68%	369	7	409	6
NaiveBayes	index.html	3	98.82%	89.60%	94.56%	0.96%	336	4	412	39
RandomForest	index_loaded.html	2	96.90%	97.66%	97.38%	2.88%	375	12	404	9
NaiveBayes	index_loaded.html	2	99.40%	86.98%	93.50%	0.48%	334	2	414	50
RandomForest	index_loaded.html	3	97.89%	96.88%	97.50%	1.92%	372	8	408	12
NaiveBayes	index_loaded.html	3	99.41%	87.24%	93.63%	0.48%	335	2	414	49
RandomForest	text_only_index_loaded.html.txt	2	96.42%	98.69%	97.62%	3.37%	377	14	401	5
NaiveBayes	text_only_index_loaded.html.txt	2	99.15%	91.88%	95.73%	0.72%	351	3	412	31
RandomForest	text_only_index_loaded.html.txt	3	97.42%	98.69%	98.12%	2.41%	377	10	405	5
NaiveBayes	text_only_index_loaded.html.txt	3	97.55%	93.98%	95.98%	2.17%	359	9	406	23

Kết quả kịch bản thử nghiệm 3

05.b Thử nghiệm và kết quả

2. Kết quả

algorithm	name	n-gram	percision	recall	accuracy	fail alarm	TP	FP	TN	FN
RandomForest	index.html	2	98.25%	84.85%	91.71%	1.50%	168	3	197	30
NaiveBayes	index.html	2	100.00%	77.78%	88.94%	0.00%	154	0	200	44
RandomForest	index.html	3	98.21%	83.33%	90.95%	1.50%	165	3	197	33
NaiveBayes	index.html	3	100.00%	76.26%	88.19%	0.00%	151	0	200	47
RandomForest	index_loaded.html	2	98.25%	84.85%	91.71%	1.50%	168	3	197	30
NaiveBayes	index_loaded.html	2	100.00%	76.26%	88.19%	0.00%	151	0	200	47
RandomForest	index_loaded.html	3	98.21%	83.33%	90.95%	1.50%	165	3	197	33
NaiveBayes	index_loaded.html	3	100.00%	72.73%	86.43%	0.00%	144	0	200	54

Kết quả kịch bản thử nghiệm 4

05.b Thử nghiệm và kết quả

3. Nhận xét

- Trong tất cả các kịch bản, các thuật toán phát hiện đều có thể cho ra kết quả dự đoán tốt với hầu hết các dự đoán đều có với độ chính xác vượt 90%.
- Thuật toán Random Forest cho kết quả với độ chính xác cao hơn so sánh với Naive Bayes trong các thử nghiệm.
- Việc chọn lựa các n-gram để thể hiện các trang web từ các bộ đặc trưng lệch không làm ảnh hưởng nhiều đến kết quả của bộ phát hiện.
- Việc bóc tách và huấn luyện đối theo các từng loại văn bản của trang web chưa cho được kết quả thực sự đáng chú ý.
- Bộ phát hiện xây dựng hoạt động tốt trên bộ dữ liệu chuẩn với tỉ lệ chính xác trên 88% cho thuật toán NaiveBayes, 90% cho thuật toán Random Forest

Kết luận

- Khóa luận đã đề xuất mô hình kết hợp để phát hiện tấn công thay đổi giao diện website áp dụng cùng lúc cả hai kỹ thuật phát hiện dựa trên chữ ký và phát hiện dựa trên bất thường.
- Đã áp dụng thuật toán học máy để cho bộ phát hiện, kết quả đạt được với các chỉ số cao, chứng tỏ thuật toán sử dụng phù hợp với mô hình.
- Cụ thể một số các kết quả khác như sau:
  - Hệ thống nghiên cứu phát hiện tấn công thay đổi giao diện trước đó
  - Đã thu thập và xử lý một lượng lớn các trang web bị tấn công thay đổi giao diện. Hoàn chỉnh công cụ thu thập dữ liệu.

Định hướng nghiên cứu tiếp theo

Đồng bộ

- Đưa quá trình thu thập và xử lý về cùng nền tảng
- Tăng độ thân thiện cho việc sử dụng

Hoàn thiện

- Đánh giá khả năng của phát hiện dựa trên chữ ký
- Xây dựng cơ sở dữ liệu

Kết hợp

- Sử dụng kết hợp nhiều loại dữ liệu khác cho phát hiện