Phân loại tên miền sử dụng các đặc trưng ngữ nghĩa trong hệ thống phát hiện DGA Botnet

Tống Văn Vạn, Nguyễn Linh Giang, Trần Quang Đức

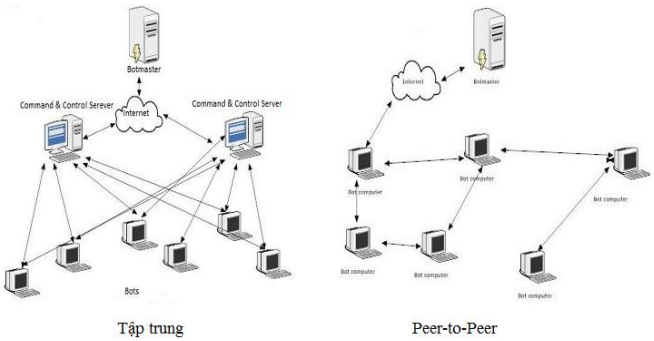
***Tóm tắt*—**Các mạng *botnet* giữ vai trò then chốt trong một số đe doạ vào mạng máy tính như tấn công từ chối dịch vụ *DDoS*, tấn công bằng thư rác, ... Nhiều dạng *botnet* sử dụng cơ chế điều khiển tập trung và che dấu các máy chủ *C&C* trong một danh sách ngẫu nhiên được sinh bằng thuật toán tạo tên miền (*DGA*). Việc phát hiện và theo dõi các *DGA botnet* trở nên cấp thiết do tính đơn giản của mô hình, khả năng che giấu của các máy chủ *C&C*.

Báo cáo trình bày phương pháp phát hiện *DGA botnet* dựa vào phân tích lưu lượng *DNS*. Phương pháp sử dụng các đặc trưng ngữ nghĩa của tên miền, kết hợp với độ đo *entropy* để đánh giá khả năng xuất hiện của tên miền. Quá trình hân loại tên miền được thực hiện qua các bước: lọc các tên miền, phân tích các đặc trưng ngữ nghĩa và phân cụm dựa trên biến thể của khoảng cách *Mahalanobis*. Quá trình này là cải tiến của phương pháp *Phoenix* [6], trong đó sử dụng biến thể thuật toán *K-means* để cải thiện hiệu năng của bước huấn luyện hệ thống. Hệ thống phát hiện *DGA botnet* đã được triển khai và thử nghiệm trên bộ dữ liệu thu thập từ trang *Alexa* và các tên miền giả mạo.

Từ khóa - Bi-gram, DGA Botnet, Mahalanobis distance, NXDomain.

1. GIỚI THIỆU

Botnet là mạng các máy tính, thiết bị tính toán bị lây nhiễm (*bot*) và bị điều khiển bằng các máy chủ *C&C* qua những kênh liên lạc đặc thù. Những kiến trúc mạng *botnet* chủ yếu bao gồm: kiến trúc tập trung, kiến trúc mạng ngang hàng và kiến trúc lai.



Hình 1. Kiến trúc tập trung và mạng ngang hàng *P2P*.

Kiến trúc tập trung

Kiến trúc tập trung bao gồm một trạm điều khiển trung tâm, hệ thống các máy chủ điều khiển *C&C* và nhóm các máy trạm chịu điều khiển của máy chủ *C&C*. Trạm này sẽ gửi các thông điệp điều khiển tới toàn mạng. Mô hình này dễ thực thi và tùy chỉnh. Tuy nhiên, mô hình này rất dễ bị phát hiện bởi các *IDS*. Một số botnet sử dụng mô hình này là AgoBot[15], SDBot[16], and Zotob.

Kiến trúc ngang hàng P2P

Để khắc phục những hạn chế của mô hình trung tâm, các *botnet* chuyển sang mô hình *P2P*. So với mô hình trung tâm, mô hình ngang hàng *P2P* khó bị phát hiện và phá hủy, và có thể gửi thông tin từ bất kì điểm nào trong mạng đến các điểm khác. Tuy nhiên việc thiết kế các mô hình này khá phức tạp. Một số *botnet* sử dụng mô hình này là: Phatbot[17] và Peacomm [18].

Mô hình lai

Trong mô hình lai, bot không trực tiếp liên hệ tới *botmaster* mà chỉ lắng nghe và chờ các kết nối và các lệnh từ *botmaster*. *Botmaster* sẽ quét ngẫu nhiên trên mạng *Internet* và sau đó gửi thông điệp khi đã phát hiện được *bot*.

DGA (Domain Generation Algorithm) Botnet

Hiện nay, phần lớn các *botnet* vẫn đang sử dụng kiến trúc tập trung do dễ xây dựng và phát triển. Các *DGA botnet* được thiết kế để khắc phục những nhược điểm của kiến trúc tập trung. Với các *botnet* trước đây, *bot* sẽ kết nối định kì đến máy chủ *C&C* và chờ lệnh. Do đó nếu máy chủ *C&C* bị phát hiện thì mạng *botnet* sẽ bị phá hủy.

Trong các *DGA botnet*, tên miền của máy chủ *C&C* sẽ được sinh ngẫu nhiên. Khi *bot* muốn kết nối đến máy chủ *C&C*, chúng sẽ chạy thuật toán và sinh ra tập tên miền sau đó *bot* sẽ kết nối lần lượt với từng tên miền trong tập này. Tại từng thời điểm *botmaster* biết tập tên miền do *bot* tạo ra để đăng kí địa chỉ cho máy chủ *C&C*. Những tên miền của các máy chủ *C&C* trong tập hợp tên miền do *botmaster* sinh ra phần lớn không được đăng ký và tương ứng với địa chỉ *IP*, và là những tên miền không tồn tại *NXDomain* (*Non-eXistent Domain*).

Ưu điểm của *DGA Botnet* là nếu địa chỉ của máy chủ *C&C* bị phát hiện và chặn tất cả kết nối đến những địa chỉ này, mạng *botnet* vẫn không bị loại bỏ hoàn toàn. Vấn đề là ở chỗ tại mỗi thời điểm, tập tên miền được sinh ra sẽ khác nhau. Do đó tại những lần kết nối sau đó, tập tên miền sau sẽ khác so với lúc trước. *Botmaster* chỉ cần đăng kí một địa chỉ mới và *bot* vẫn sẽ hoạt động như bình thường.

Hiện nay có rất nhiều công trình nghiên cứu về *botnet* đã được công bố. Cơ chế phát hiện *botnet* do Etienne[1] đề xuất sử dụng hành vi của *botnet* thông qua những đặc trưng lưu lượng *DNS*. Cơ chế này cho phép không cần duy trì danh sách đen hay hay cập nhật các dấu hiệu của *bot*. Phương pháp sử dụng các đặc trưng của lưu lượng *DNS* như bản ghi *Name Server*, địa chỉ *IP*, thời gian tồn tại của tên miền và các chữ cái xuất hiện trong tên miền. Quá trình phân loại được thực hiện bằng thuật toán *Naive Bayes*. Hiệu quả của cách tiếp cận này không cao do việc sử dụng các đặc trưng và thuật toán nêu trên không đủ để phát hiện chính xác .

Yadav và Reddy[2] đề xuất phương pháp phát hiện *botnet* dựa vào phân bố xác xuất của các kí tự trong lưu lượng *DNS*, và ánh xạ chúng với tập địa chỉ *IP*. Phương pháp này sử dụng khoảng cách *K-L* (*Kullback-Leibler*), khoảng cách biến đổi và chỉ số *Jaccard* để phát hiện *NXDomain*.

Nhauo Davuth and Sung-Ryul Kim[3] đưa ra phương pháp phân loại tên miền dựa vào *SVM* và phân phối *bi-gram* của tập dữ liệu. Các đặc trưng *bi-gram* của tên miền được sử dụng và trích rút và được lọc theo ngưỡng. Sau đó, bộ phân loại *SVM Light* [4] được sử dụng để phân lớp các tên miền bình thường và *NXDomain*. Phương pháp này có hiệu quả khá cao tuy vậy chỉ cho phép phân biệt các dạng *botnet* đã biết, khi có những dạng *botnet* chưa biết (chưa được huấn luyện) thì hiệu quả phân loại giảm xuống.

Zhou, Li, Miao, and Yim[5] đề xuất cơ chế phát hiện *DGA Botnet* dựa vào phân tích lưu lượng *DNS* do các *NXDomain* sinh ra. Phương pháp sử dụng những đặc trưng: khoảng thời gian hoạt động, thời gian tồn tại của mỗi tên miền. Những đặc trưng này được xác định dựa trên phân tích lưu lượng *DNS*. Hệ thống đề xuất kết hợp với địa chỉ *IP* với các đặc trưng nêu trên phân cụm những tên miền này. Danh sách các *DGA Botnet* được xác định dựa trên tính toán độ tương tự của mỗi nhóm tên miền.

Schiavoni, Schiavoni, Maggi, Zanero[6] đã đề xuất cơ chế *Phoenix* dựa vào thông tin ngữ nghĩa của tên miền và các đặc trưng dựa vào địa chỉ *IP* để phát hiện các tên miền được sinh bởi *DGA*. Ban đầu *Phoenix* lọc bớt những *NXDomain* dựa vào danh sách các tên miền đen (*Blacklist*). Sau đó hệ thống phân loại các tên miền thành hai tập: tên miền bình thường và tên miền không tồn tại *NXDomain*. Trong quá trình này, hệ thống sử dụng hàm khoảng cách *Mahalanobis* để đánh giá độ tương hợp và dùng địa chỉ *IP* để phân cụm các *NXDomain*. Những *NXDomain* trong một cụm có khả năng lớn sẽ do cùng một thuật toán *DGA* tạo sinh. Sau quá trình phân cụm, chúng ta có các cụm tên miền tương ứng với những thuật toán *DGA* khác nhau sinh ra. Đặc trưng của mỗi cụm *NXDomain* sẽ được xác định để phục vụ quá trình phân loại và phát hiện những tên miền chưa biết.

Chúng tôi cải tiến một số giai đoạn của *Phoenix* để nâng cao hiệu năng của hệ thống, trong đó tập trung vào độ chính xác của quá trình phát hiện tên miền, giảm thời gian tính toán các đặc trưng, khoảng cách trong quá trình huấn luyện và phát hiện. Những cải tiến đó bao gồm:

- Sử dụng nhiều đặc trưng ngữ nghĩa hơn so với *Phoenix*, bao gồm trọng số *bi-gram*, tần suất của *bi-gram*, *entropy* và mức độ ý nghĩa của tên miền.
- Khảo sát dữ liệu tên miền và sử dụng hàm khoảng cách *Mahalanobis* giản lược để giảm thời gian tính toán trong thực tế.
- Sử dụng thuật toán *K-means* thay vì sử dụng *DBSCAN* như trong *Phoenix* và thực hiện một số cải tiến nhỏ để đạt hiệu quả phân cụm tốt hơn.

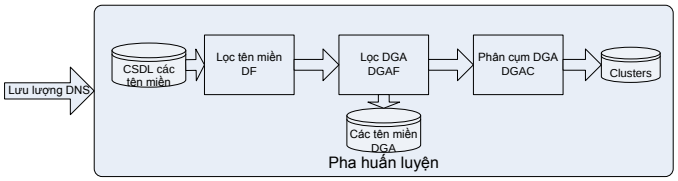
Phần tiếp theo của bài báo trình bày các giai đoạn của phương pháp đề xuất phát hiện *DGA botnet*. Phần ba mô tả các kết quả thực nghiệm và bình luận đánh giá.

2. PHÂN LOẠI TÊN MIỀN TRONG PHÁT HIỆN *DGA BOTNET*

Hệ thống phát hiện *DGA botnet* tập trung vào phân tích lưu lượng *DNS*, để từ đó phát hiện những tên miền thuộc tập tên miền do một thuật toán *DGA* đặc trưng của một *botnet* tạo sinh ra. Kết quả của quá trình phân tích cho phép cảnh báo những tấn công sử dụng dạng *botnet* do hệ thống phát hiện được. Hệ thống phát hiện *DGA botnet* sẽ bao gồm hai thành phần chính: khối huấn luyện và khối phát hiện *botnet*.

- Khối huấn luyện thực hiện phân tích các dữ liệu liên quan tới tên miền của những dạng *botnet* đã biết, xác định các dạng tên miền liên quan tới từng *botnet* xác định, do những thuật toán *DGA* khác nhau tạo sinh.
- Khối phát hiện sẽ phân tích lưu lượng *DNS*, xác định những tên miền thu nhận được thuộc nhóm tên miền bình thường hay tên miền thuộc một dạng *botnet* nào đó.

Bài báo này tập trung vào quá trình huấn luyện với mục tiêu cơ bản là phát hiện các đặc trưng của tên miền do một thuật toán *DGA* tạo sinh. Kết quả của quá trình huấn luyện là các phân lớp tên miền *NXDomain* đại diện cho các thuật toán *DGA*.



Hình 2. Các pha huấn luyện hệ thống.

Hệ thống huấn luyện phân loại tên miền (hình 2) gồm có các thành phần sau:

- Cơ sở dữ liệu các tên miền: Tên miền sau khi được thu thập sẽ được lưu vào cơ sở dữ liệu.
- Khối lọc tên miền *DF*: Trong quá trình thu thập và khảo sát tên miền trong lưu lượng *DNS*, nhiều tên miền bình thường như: facebook.com, youtube.com, yahoo.com… bị lẫn trong tập dữ liệu thu được. Vấn đề là cần thu thập và phân tích các tên miền *NXDomain* nghi ngờ do *DGA botnet* sinh ra, do đó tập dữ liệu tên miền sau khi thu thập sẽ được lọc và tinh chỉnh bằng bước lọc trong khối *DF*. Sau khi được lọc, tập các tên miền còn lại có khả năng cao là *NXDomain* do thuật toán *DGA* của các *botnet* tạo ra.
- Khối lọc *DGA*: Tập dữ liệu còn lại sau bước lọc tên miền sẽ được phân tách bằng khối *DGA Filtering*. Tại đây, các tên miền sẽ được phân loại thành hai nhóm: nhóm tên miền bình thường và nhóm *NXDomain* do các thuật toán sinh tên miền của *botnet* tạo ra.
- Khối phân cụm *DGA DGA Clustering*: Sau khi lọc *DGA*, các tên miền trong phân lớp *NXDomain* sẽ được phân cụm bằng khối *DGA Clustering* dựa vào thuật toán phân cụm *K-means*. Trong đó, *NXDomain* sẽ được phân nhóm thành các cụm khác nhau. Mỗi cụm tương ứng với một dạng *DGA botnet* và do cùng một thuật toán *DGA* tạo sinh. Quá trình phân cụm sử dụng đặc trưng địa chỉ *IP* của tên miền để phân loại. Thông thường, mỗi tên miền sẽ có một tập người dùng kết nối đến và địa chỉ *IP*

của những người dùng này sẽ được sử dụng để xác định khoảng cách của hai tên miền.

Chi tiết từng pha trong phương pháp phát hiện *NXDomain* sẽ được trình bày dưới đây.

A) Lọc tên miền DF

Trong quá trình tên miền được thu thập, có rất nhiều tên miền bình thường bị lẫn vào trong tập dữ liệu như: amazon.com, wikipedia.org, instagram.com… Các tên miền này có thể làm nhiễu tập dữ liệu thu được và sẽ làm ảnh hưởng đến kết quả phân loại *NXDomain*. Do đó ở khối *DF*, chúng tôi xây dựng một bộ lọc để lọc bớt những tên miền bình thường. Trong đó chúng tôi sử dụng hai kĩ thuật là lọc dựa vào những tên miền bình thường đã biết và dựa vào các đặc điểm ngữ nghĩa của tên miền.

i. Lọc sử dụng danh sách White-list

Chúng tôi sử dụng 1.000.000 tên miền được truy nhập nhiều nhất của Alexa. Sau giai đoạn này, một số tên miền bình thường như: dantri.com.vn, vnexpress.net… sẽ bị lọc bỏ. Tập dữ liệu sau khi đi qua khối này sẽ giảm đi đáng kể.

ii. Trích xuất đặc trưng

Pha trích xuất đặc trưng sử dụng những đặc trưng ngữ pháp sau:

- Trọng số *bi-gram*:

  Đặc trưng này cho biết mức độ xuất hiện của các *bi-gram* trong tên miền. Nếu một tên miền được sinh ra bởi *DGA Botnet* thì đặc trưng này sẽ nhỏ. Chúng tôi tách ra *p bi-gram* của một tên miền *d*, sau đó dựa vào tần suất xuất hiện và thứ hạng của tên miền trong từ điển tiếng Anh để đánh giá trọng số.

$$S(d) = \frac{\sum_{t \in p} count(t) * n(t)}{|p|} \quad (1)$$

  Trong đó: *n(t)* là tần suất xuất hiện *bi-gram t* trong tên miền d.

  $|p|$ là số phần tử *bi-gram* của tên miền *d*.

- Tần suất của *bi-gram*:

  Đặc trưng này cho biết được tần suất xuất hiện của các *bi-gram* thường hay xuất hiện và *bi-gram* ít xuất hiện. Chúng tôi đã khảo sát 100.000 tên miền có thứ hạng cao của Alexa thấy xuất hiện 1459 cụm *bi-gram* khác nhau. Quá trình phân tích tần suất được thực hiện dựa trên ngưỡng θ, phân biệt các *bi-gram* xuất hiện nhiều và các *bi-gram* xuất hiện ít. Sau khi khảo sát tập dữ liệu trên, chúng tôi chọn $\theta = 520$. Độ đo tần suất *F* bằng tỉ lệ giữa số *bi-gram* xuất hiện nhiều và xuất hiện ít của một tên miền để đánh giá mức độ xuất hiện của một tên miền đó.

- *Entropy*:

  Entropy đặc trưng cho độ bất định của một tên miền. Ví dụ tần số xuất hiện của chữ *x* sẽ khác chữ phổ biến hơn *t*. Những tên miền nào được sinh ra bởi thuật toán *DGA* sẽ có đặc trưng này cao hơn so với tên miền bình thường.

$$E(d) = -\sum_{t \in p} \frac{count(t)}{N} * \log(\frac{count(t)}{N}) \quad (2)$$

  Trong đó:

  t là một *bi-gram* trong *p bi-gram* của tên miền *d*.

  count(t) là chỉ số của *bi-gram t* trong tập từ điển.

  N là số phần tử trong tập từ điển.

- Mức độ ý nghĩa tên miền

  Đặc trưng này cho thấy mức độ ý nghĩa của tên miền. Những tên miền do *DGA botnet* sinh ra sẽ có giá trị này thấp.

  Tên miền được chia thành những từ $w(i) \geq 3$. Khi đó với tên miền *d* ta có:

$$R(d) = \frac{\sum_{i \in n} w(i)}{p} \quad (3)$$

  Trong đó: *p* là số kí tự của tên miền *d*.

  n là số từ có ý nghĩa trong tên miền.

  Nếu mức độ ý nghĩa của tên miền *R(d)* được tính như trong công thức (3) thì thời gian tính toán của thuật toán khá lớn. Do đó, chúng tôi sử dụng ba đặc trưng gồm tần suất của *bi-gram, trọng số bi-gram, entropy* để ước lượng mức độ ý nghĩa của tên miền.

  Giá trị *R(d)* được ước lượng như sau:

  If $S(d) \geq 325$ *and* $E(d) \leq 0.28$ *and* $F(d) \leq 0.32$ *then*

  $R(d) = 0.5$ *Else* $R(d) = 0$

  Các giá trị ngưỡng 300, 0.25, 0.3 được đưa ra dựa vào quá trình khảo sát và đánh giá tập tên miền ở trên.

iii. Lọc tên miền dựa trên đặc trưng ngữ pháp

Sau khi tập dữ liệu tên miền được lọc bởi danh sách Whitelist của Alexa, các tên miền trong tập dữ liệu sẽ được rút ra các đặc trưng ngữ nghĩa để phục vụ cho các giai đoạn phía sau. Chúng tôi sử dụng 4 đặc trưng ngữ nghĩa ở trên và đưa ra một ngưỡng để lọc một số tên miền ra khỏi tập dữ liệu. Hơn 100.000 tên miền bình thường có thứ hạng cao nhất của Alexa và hơn 300.000 NXDomain của DGA Botnet được khảo sát để đưa ra giá trị ngưỡng trong quá trình lọc sao cho số lượng *NXDomain* bị lọc theo tên miền bình thường là không đáng kể.

Thuật toán của quá trình lọc được mô tả như sau:

If $S(d) \geq 300$ *and* $E(d) \leq 0.25$ *and* $F(d) \leq 0.3$ *and* $R(d) = 0.5$

then domain d is discarded

B) Lọc DGA

Sau khi qua khối lọc tên miền *DF*, tập dữ liệu sẽ được lọc tiếp bằng khối lọc *DGA*. Các đặc trưng đề xuất trong mục trước sẽ tham gia vào vector đặc trưng

$$v(d) = [R(d), S[d], E[d]]^T.$$

Tập dữ liệu để huấn luyện và trích xuất đặc trưng gồm 10000 tên miền có thứ hạng cao của Alexa, rút ra các đặc trưng trên và tính ra giá trị trung bình $\mu = [\overline{R}, \overline{S}, \overline{E}]$ của mỗi giá trị và ma trận tương quan *C* cho biết giá trị trung bình của các đặc trưng và tương quan giữa các đặc trưng đó.

Các tên miền sẽ được phân loại bằng sử dụng khoảng cách *Mahalanobis*. Với một tên miền *d'* chúng tôi sẽ ước lượng khoảng cách giữa $x = v(d')$ và trọng tâm μ. Khoảng cách này được tính như sau:

$$d_{mah}(x) = \sqrt{(x - \mu)^T C^{-1} (x - \mu)} \quad (4)$$

Những tên miền bình thường sẽ gần trọng tâm hơn so với *NXDomain*. Ngưỡng khoảng cách được xác định dựa trên quan sát trên tập hợp dữ liệu thử nghiệm. Nếu $d_{mah}(x) > \theta_d$ thì x được phân loại là *NXDomain* và x sẽ được gán nhãn là tên miền bình thường nếu ngược lại. Ngưỡng $\theta_d = 1.9$, được chúng tôi chọn dựa vào quá trình khảo sát tập dữ liệu tên miền ở trên.

C) Phân cụm DGA

Đây là bước phân cụm các *NXDomain* có khả năng cao cùng do một thuật toán *DGA* tạo ra. Thuật toán *K-means* được áp dụng và điều chỉnh phù hợp với việc xác định số cụm động.

Đối với *K-means*, ta phải xác định trước số cụm cần đưa ra. Ban đầu chúng tôi để mặc định là hai. Sau khi phân cụm xong, nếu khoảng cách giữa một phần tử trong cụm Ω_i với tâm cụm μ_i lớn hơn một ngưỡng θ_i xác định thì phần tử đó sẽ được tách ra thành một cụm mới. Như vậy, trong cụm chỉ còn lại những phần tử đủ gần so với tâm cụm, tức là có khoảng cách tới μ_i nhỏ hơn ngưỡng xác định θ_i. Khoảng cách giữa hai tên miền trong một cụm được xác định bằng danh sách các người dùng kết nối tới tên miền.

Gọi U_1 và U_2 lần lượt là tập hợp các người dùng kết nối đến hai tên miền d_1 và d_2, khoảng cách giữa hai tên miền:

$$Distance(d1, d2) = |U_1 \cap U_2|$$

Chi tiết giải thuật như sau:

Khởi tạo ngẫu nhiên trọng tâm của các phân cụm $\Omega_1, \Omega_2, ..., \Omega_n$ *là* $\mu_1, \mu_2, ..., \mu_n$;

Repeat until convergence:

Gán mỗi điểm x_i *vào cụm* Ω_k *với đại diện là* μ_k *theo khoảng cách nhỏ nhất*

Tính lại giá trị trọng tâm μ_k *cho mỗi cụm* Ω_k

End Until

For k from 1 to n

Foreach tên miền *d in cluster* Ω_k

If distance(d, μ_k*)* $> \theta_k$ *then*

$num_k = num_k + 1$

End If

End For

If $num_k > \theta_c$ *then*

Tạo cụm mới từ các thành viên có tên miền cách trọng tâm μ_k *lớn hơn ngưỡng* $\mu_k > \theta_{ck}$

$n = n + 1$

End If

End For

Với α_k là trung bình khoảng cách các phần tử trong cụm Ω_k tới tâm cụm μ_k và n_k là số phần tử của cụm Ω_k. Giá trị ngưỡng được lựa chọn $\theta_k = 1.5 * \alpha_k$ *và* $\theta_{ck} = 0.3 * n_k$.

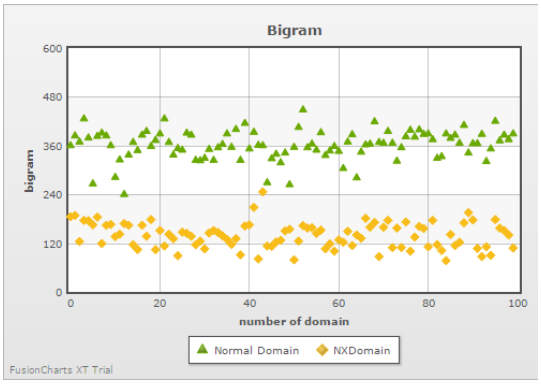
Với dữ liệu thu được là các tên miền và danh sách người dùng truy nhập đến tên miền, thuật toán sẽ duyệt lần lượt từng tên miền. Mỗi tên miền sẽ tìm được danh sách các tên miền lân cận để thêm vào cụm sau đó sẽ tiếp tục mở rộng cụm và xét với các tên miền khác. Quá trình lặp lại cho đến khi không mở rộng được cụm và không có cụm mới tạo ra.

3. KẾT QUẢ THỰC NGHIỆM

Để đánh giá hệ thống trên, chúng tôi đã sử dụng bộ dữ liệu hơn 300.000 tên miền được sinh ra bởi *DGA Botnet* của *DNS-BH – Malware Domain Blocklist* [7] gồm các tên miền của *Conficker, Kraken, Tinba, Bebloh, Tovar_GOZ* và 100.000 tên miền bình thường có thứ hạng cao của Alexa.

Chúng tôi tiến hành khảo sát, phân tích và trích chọn các đặc trưng của các bộ dữ liệu trên.

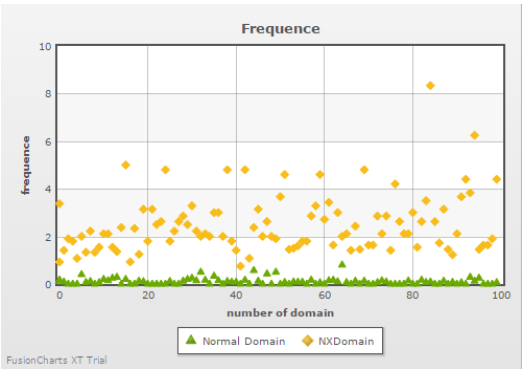
Đầu tiên chúng tôi khảo sát các *n-gram* của các tập tên miền, chúng tôi nhận thấy *bi-gram* là một đặc trưng tốt vì có sự khác nhau giữa tập tên miền bình thường và *NXDomain*, do đó chúng tôi sử dụng *bi-gram* trong quá trình tính toán. Dưới đây là biểu đồ thống kê cho thấy sự khác nhau về trọng số *bi-gram* giữa tên miền bình thường và *NXDomain*:



Hình 3. Trọng số *Bi-gram* của tên miền bình thường và *NXDomain*.

Sau khi chúng tôi thu thập được một số tập tên miền bình thường và *NXDomain*, đặc trưng trọng số *bi-gram* của những tập tên miền này được rút ra. Hình 3 cho thấy tập giá trị trọng số *bi-gram* của tên miền bình thường và *NXDomain* khá tách biệt. Trọng số *bi-gram* trong tập tên miền bình thường nằm trong khoảng từ 240 đến 470, còn *bi-gram* của *NXDomain* thì thấp hơn chỉ từ 100 đến 210.

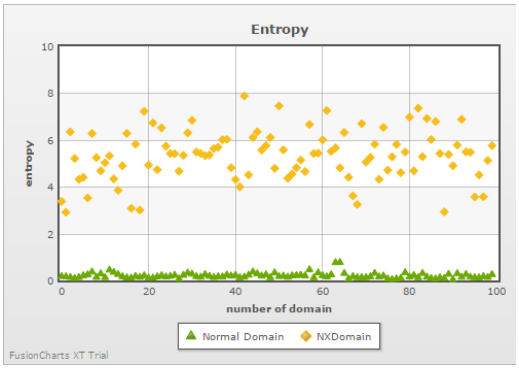
Tiếp theo chúng tôi khảo sát đặc trưng tần suất của *bi-gram* của tập tên miền. Hình 4 dưới đây là biểu đồ cho thấy sự khác nhau về trọng số này giữa tên miền bình thường và *NXDomain*.



Hình 4. Tần suất của Bi-gram của tên miền bình thường và NXDomain.

Từ hình 4, ta có thể thấy tần suất xuất hiện của *bi-gram* trong tập tên miền bình thường và *NXDomain* là khá khác biệt. Giá trị đối với tập tên miền bình thường nằm trong khoảng từ 0 đến 0.4, còn giá trị đối với *NXDomain* thì phân bố dải rác trong khoảng từ 0.75 đến 5.

Cuối cùng chúng tôi khảo sát và trích chọn đặc trưng về *entropy* giữa tên miền bình thường và *NXDomain*.

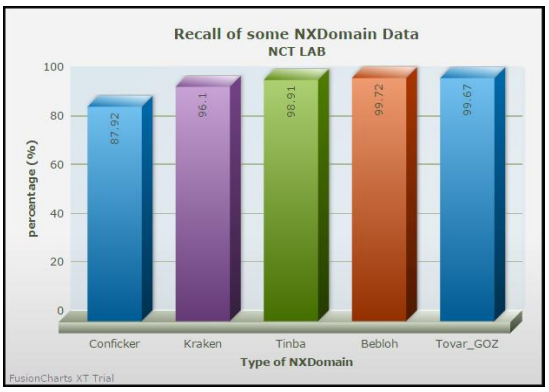


Hình 5. *Entroy* của tên miền bình thường và *NXDomain*.

Thông qua hình vẽ trên chúng ta có thể thấy được giá trị *entropy* của tập tên miền bình khá thấp và phân bố tập trung trong khoảng từ 0 đến 0.77. Tuy nhiên giá trị này đối với *NXDomain* thì cao hơn và phân bố rải rác từ 2.91 đến 7.88.

Trong giai đoạn lọc tên miền dựa vào các đặc trưng ngữ pháp trên, có khoảng 60 đến 70 % tên miền bình thường bị lọc. Do số lượng tên miền bình thường trong thực tế rất lớn nên số lượng tên miền bị lọc trong khối lọc tên miền dựa vào các đặc trưng ngữ pháp rất nhiều.

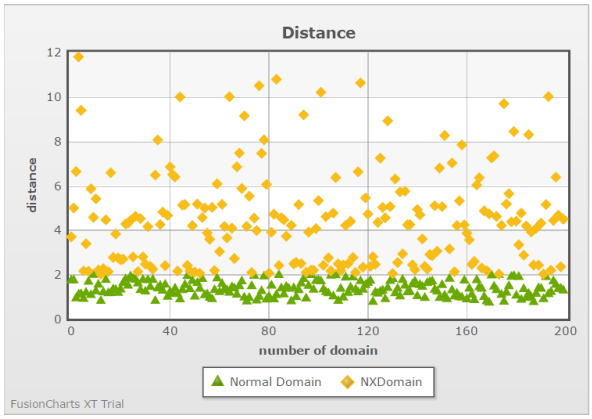
Sau lọc tên miền dựa vào các đặc trưng ngữ pháp, tên miền sẽ đi qua pha phân loại để chia tên miền thành 2 tập là tên miền bình thường và *NXDomain*.



Hình 6. Hệ số triệu hồi (*Recall*) của các cụm tên miền.

Kết quả trên cho thấy hệ thống phát hiện những tên miền do *Conficker* sinh ra không tốt lắm khoảng *88%* trong khi đó đối với những tên miền của *Kraken, Tinba, Bebloh* thì hệ thống cho kết quả khá tốt, hiệu quả phát hiện lên tới *99%* tên miền trong tập tên miền của *DGA Botnet*.

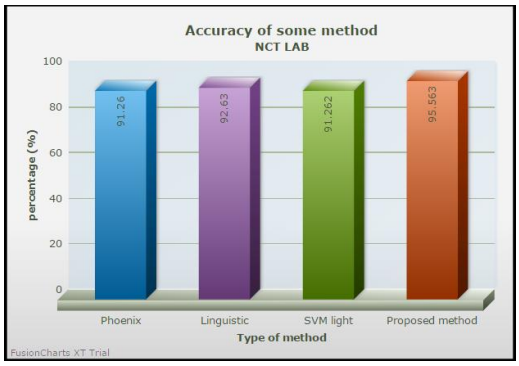
Biểu đồ trong hình 7 cho thấy khoảng cách *Mahalanobis* của tập tên miền bình thường và *NXDomain* :



Hình 7. Khoảng cách *Mahalanobis* của tên miền bình thường và *NXDomain*.

Từ hình vẽ trên chúng ta có thể thấy sau khi rút ra một số đặc trưng ngữ nghĩa và sử dụng để tính khoảng cách, khoảng cách của tập tên miền bình thường tới trung tâm nhỏ hơn so với *NXDomain*. Do đó khi phân loại, những tên miền nào mà có khoảng cách *Mahalanobis* càng lớn thì tỉ lệ là *NXDomain* càng cao.

Kết quả dưới đây so sánh độ chính xác của phương pháp của chúng tôi với các kết quả của các công trình đã công bố trước đó *Phoenix*[6], *Lingustic*[2], *SVM light*[3]:



Hình 8. So sánh độ chính xác của *Phoenix, Linguistic* và *SVM Light* với phương pháp đề xuất.

Trong đó độ chính xác được tính theo công thức:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} * 100 \quad . \qquad (5)$$

Với: *TP* và *FP* lần lượt là tỉ lệ *NXDomain* được nhận diện đúng và sai.

TN và *FN* lần lượt là tỉ lệ tên miền bình thường được nhận diện đúng và sai.

Hệ thống đề xuất này đã cải tiến *Phoenix*[6] ở các bước lọc tên miền *DF*, lọc và phân cụm *DGA*. Những hiệu chỉnh này làm tăng thêm hiệu quả cho các thuật toán như trong khối lọc các tên miền bình thường để giảm nhiễu trong quá trình phân loại tên miền. Dựa vào hình 8 cho thấy, độ chính xác của phương pháp đề xuất dường như tốt hơn so với *Phoenix*, và một số phương pháp khác.

4. KẾT LUẬN

Bài báo đã đề cập tới một số cải tiến quá trình huấn luyện và phát hiện tên miền không tồn tại [6], một bước quan trọng trong

phát hiện *botnet*. Kết quả thử nghiệm thu được khá khả quan. Khoảng *88%* đến *99.72%* tên miền trong tập dữ liệu *NXDomain* bị phát hiện. Ngoài ra tỉ lệ tên miền bình thường bị phát hiện sai là khá nhỏ, chỉ khoảng 2 đến 3 % . Điều này đạt được do hệ thống đã lọc bớt những tên miền bình thường dựa vào các đặc trưng ngữ nghĩa và danh sách tên miền thực (whitelist). Những kết quả thu được còn nhiều hạn chế và cần có những cải tiến phù hợp để tăng khả năng phát hiện đúng tên miền. Hơn nữa, phương pháp đề xuất chỉ cho phép phát hiện được những tên miền do thuật toán *DGA* sinh ra, nhưng chưa thể phát hiện được những tên miền được đặt tên giống tên miền bình thường. Do đó để có thể phát hiện những tên miền này, cần phải dựa vào những thông tin bổ sung, như địa chỉ IP, số luồng truy nhập trong một đơn vị thời gian, số gói tin trên một luồng, kích thước mỗi gói tin, tần suất gửi gói tin, khoảng thời gian gửi.

5. LỜI CẢM ƠN

Các nghiên cứu trong báo cáo này được sự hỗ trợ của đề tài cấp Bộ năm 2016-2017 "Nghiên cứu xây dựng hệ thống xử lý tấn công từ chối dịch vụ phân tán và phát hiện Botnet" mã số B2016-BKA-06

6. TÀI LIỆU THAM KHẢO

[1] E. Stalmans, "A Framework for DNS Based Detection and Mitigation of Malware Infections on a Network", Information Security South Africa Conference, 2011.

[2] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names", Proceedings of the 10th annual Conference on Internet Measurement, IMC '10, pages 48–61, New York, NY, USA, 2010, ACM.

[3] Nhauo Davuth, Sung-Ryul Kim, "Classification of Malicious Domain Names using Support Vector Machine and Bi-gram Method", International Journal of Security and Its Applications, Vol. 7, No. 1, January, 2013.

[4] T. Joachims," SVM light, Making large-Scale SVM Learning Practical", Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (eds.), MIT-Press, 1999.

[5] Zhou, Li, Miao, and Yim, "DGA-Based Botnet Detection Using DNS Traffic", Journal of Internet Services and Information Security (JISIS), volume: 3, number: 3/4, pages 116-123.

[6] Stefano Schiavoni, Federico Maggi, Lorenzo Cavallaro, Stefano Zanero, "Phoenix: DGA-Based Botnet Tracking and Intelligence", Chapter Detection of Intrusions and Malware, and Vulnerability Assessment Volume 8550 of the series Lecture Notes in Computer Science pages 192-211, Springer, 2014.

[7] http://www.malwaredomains.com/?cat=111

[8] http://www.alexa.com/

[9] G. Eason, B. Noble, I. N. Sneddon, "On certain integrals of Eggdrop: Open source IRC bot, http://www.eggheads.org/,1993.

[10] C. Associates. GTBot1, http://www3.ca.com/securityadvisor/pest/pest.aspx?id=453073312, 1998.

[11] Chao Li, Wei Jiang, Xin Zou, "Botnet: Survey and Case Study", Fourth International Conference on Innovative Computing, Information and Control , 2009.

[12] Rajab MA, Zarfoss J, Monrose F, Terzis A, "A multifaceted approach to understanding the botnet phenomenon", Almeida JM, AlmeidaVAF, Barford P, eds. Proc. of the 6th ACM Internet MeasurementConf. (IMC 2006). Rio de Janeriro: ACM Press, pages 41-52, 2006.

[13] Abebe Tesfahun, D.Lalitha Bhaskari, "Botnet Detection and Countermeasures-A Survey", International Journal of Emerging Trends & Technology in Computer Science, Volume 2, Issue 4, July – August, 2013.

[14] Sophos, Troj/Agobot-A, http://www.sophos.com/ virusinfo/analyses/ trojagobota.html, 2002.

[15] Sophos, Troj/SDBot, http://www.sophos.com/ virusinfo/analyses/trojsd bot.html, 2002.

[16] Phatbot Trojan Analysis, http://www.secureworks. com/ research/threats/phatbot.

[17] M. Suenaga, M. Ciubotariu, "Symantec: Trojan. peacomm."http://www.symantec.com/security response/writeup.jsp?docid=2007011917-1403-99, February 2007.

[18] Ying Zhang, Yongzheng Zhang, Jun Xiao, "Detecting the DGA-Based Malicious Domain Names", ISCTCS 2013, CCIS 426, pages 130–137, 2014.