BÁO CÁO ĐỒ ÁN

**Tên chủ đề: SELKS\_DGA**

*Ngày báo cáo: 22/08/2022*

1. **THÔNG TIN CHUNG:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Ngày vào thực tập** | **Người hướng dẫn** |
| 1 | Châu Thanh Tuấn | 11/07/2022 | Phan Văn Hảo  Hoàng Quốc Cường |
| 2 | Nguyễn Mỹ Quỳnh |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-2)**

|  |  |
| --- | --- |
| **STT** | **Công việc** |
| 1 | Tìm hiểu và triển khai SELKS |
| 2 | Đào tạo mô hình phân biệt DGA và triển khai các biến thể |
| 3 | Kết hợp DGA với SELKS, TheHive |
| 4 | Tìm hiểu và sử dụng Docker Compose |
| 5 | Đóng gói hệ thống thành OVF |

Nội dung

[I. PHẦN KỸ THUẬT 3](#_Toc111964096)

[1. DGA 3](#_Toc111964097)

[2. SELKS 3](#_Toc111964098)

[II. PHẦN TRIỂN KHAI 6](#_Toc111964099)

[1. DGA 6](#_Toc111964100)

[2. SELKS 6](#_Toc111964101)

[III. DEMO 10](#_Toc111964102)

[IV. TÀI LIỆU KHAM THẢO 12](#_Toc111964103)

BÁO CÁO CHI TIẾT

# PHẦN KỸ THUẬT

## DGA

Malware thực hiện đa dạng các hoạt động: từ đánh cắp thông tin cho đến hạn chế tài nguyên tính toán của mục tiêu. Để thực hiện được các việc đó, malware cần phải kết nối đến C&C center để có thể nhận lệnh và các cập nhật. Botmaster – C&C Controller – sử dụng các hardcode IP để giao tiếp với malware, dẫn đến kết nối bị ngắt khi nạn nhân phát hiện ra malware. Để giải quyết vấn đề này, các nhà thiết kế malware tạo ra domain generation algorithms (DGAs) – thuật toán tạo tên miền - cho phép duy trì kết nối C&C với malware ngay cả khi nó bị phát hiện

Được nhúng vào nhiều loại malware, DGAs hoạt động bằng cách tạo ngẫu nhiên hàng nghìn tên miền và cố gắng giải quyết tất cả chúng dựa trên máy chủ DNS của nó. Botmaster chỉ cần đăng ký một vài tên miền trong số này. Khi DGA được nhúng bên trong malware tạo ra một miền mà botmaster đã đăng ký, nó có thể thiết lập một địa chỉ IP hợp lệ và giao tiếp với C&C center. Việc phát hiện các DGA domain là 1 thách thức bất đối xứng đối với bên phòng thủ, vì họ vừa phải duy trì các doamin hợp lệ, vừa phải lọc các DGA domain. Trong khi bên phía tấn công chỉ cần 1 kết nối thành công là đã có thể thực hiện hành vi độc hại.

Mô hình sử dụng kiến trúc stacked convolutional-recurrent lấy cảm hứng từ các mô hình hiện đại để khai thác văn bản. Điều này cho phép mô hình học cách phân loại sự khác biệt ở cả short-distance và long-distance patterns trong tên miền. Chuỗi được đưa vào input là toàn bộ tên miền. Sau đó nó được chuyển đổi thành 1 chuỗi số nguyên thông qua khử viết hoa và đệm đến độ dài ký tự tối đa là 82. Sau đó, nó được chuyển qua một phép nhúng vectơ để chuyển đổi mỗi ký tự thành một vectơ 128 chiều mã hóa thông tin của nó. Điều này dẫn đến tên miền được chuyển đổi thành ma trận 128 x 82, trong đó mỗi cột đại diện cho một ký tự.

Tên miền sau khi được nhúng sẽ đi qua convolutional neural network (CNN). Sau đó nó sẽ đi qua long short-term memory layer (LSTM) – 1 loại recursive neural network (RNN). LSTM được biết đến với hiệu suất hiện đại về học tập liên quan đến trình tự. Do đó, nó là một lựa chọn tự nhiên để phân loại các mẫu trong chuỗi văn bản, chẳng hạn như tên miền. Mô hình sử dụng một lớp LSTM duy nhất, nhúng toàn bộ tên miền vào một vectơ 1 x 64.

Cuối cùng nó sẽ thực hiện phân loại nhị phân trên CNN-LSTM để dự đoán liệu tên miền đầu vào có phải là DGA hay không.

## SELKS

Được phát triển bởi Stamus Networks, SELKS là một hệ sinh thái IDS/IPS/NSM hoàn chỉnh dựa trên Suricata với trình quản lý quy tắc đồ họa riêng và khả năng săn tìm mối đe dọa mạng cơ bản. SELKS có thể là một giải pháp giám sát an ninh mạng cấp sản xuất (network security monitoring - NSM) và phát hiện xâm nhập (intrusion detection system - IDS) phù hợp.

SELKS là bản phân phối trực tiếp trên Debian được xây dựng từ 5 thành phần mã nguồn mở quan trọng tạo nên tên gọi của nó SELKS - Suricata, Elasticsearch, Logstash, Kibana và Stamus Scirius Community Edition (Suricata Management và Suricata Hunting). Ngoài ra, nó bao gồm các thành phần từ Arkime (trước đây là Moloch) và Evebox, được thêm vào sau khi từ viết tắt được thành lập.

* Các thành phần chính SELKS cụ thể như sau:
  + S: Suricata - <https://suricata.io/>

Vì bất kỳ và tất cả dữ liệu trong SELKS được tạo bởi Suricata, SELKS là nơi cho thấy những gì Suricata IDS/IPS/NSM có thể làm và nhật ký giám sát giao thức mạng và cảnh báo mà nó tạo ra.

* + E: Elasticsearch - <https://www.elastic.co/products/elasticsearch>

Công cụ phân tích và tìm kiếm, lưu trữ tập trung dữ liệu để tìm kiếm nhanh chóng, mức độ liên quan được tinh chỉnh và phân tích mạnh mẽ, có thể mở rộng quy mô một cách dễ dàng.

* + L: Logstash - <https://www.elastic.co/products/logstash>

Một công cụ sử dụng để thu thập, xử lý log, nhập dữ liệu từ nhiều nguồn, biến đổi nó, sau đó chuyển vào Elastichsearch. Mỗi dòng log của logstash được lưu trữ đưới dạng json.

* + K: Kibana - <https://www.elastic.co/products/kibana>

Kibana là một giao diện người dùng mở và miễn phí cho phép bạn trực quan hóa dữ liệu Elasticsearch của mình và điều hướng Elastic Stack. Làm bất cứ điều gì từ theo dõi truy vấn đến tìm hiểu cách các request đi qua ứng dụng của bạn.

* + S: Scirius - <https://github.com/StamusNetworks/scirius>:

Scirius CE là ứng dụng mã nguồn mở của Stamus Networks kết hợp tất cả các thành phần này lại với nhau. Scirius cung cấp giao diện web cho toàn bộ hệ thống, cho phép bạn:

* + - Quản lý nhiều bộ quy tắc Suricata và các nguồn thông tin tình báo về mối đe dọa
    - Tải lên và quản lý các quy tắc Suricata tùy chỉnh và tệp dữ liệu IoC
    - Tìm kiếm các mối đe dọa bằng cách sử dụng bộ lọc xác định trước và chế độ xem theo ngữ cảnh nâng cao
    - Áp dụng ngưỡng và ngăn chặn để hạn chế mức độ chi tiết của cảnh báo nhiễu
    - Xem thống kê hiệu suất Suricata và thông tin về hoạt động của quy tắc Suricata
    - Áp dụng Kibana, EveBox và Cyberchef cho Suricata NSM và dữ liệu cảnh báo
  + EveBox - <https://evebox.org/>
  + Arkime - <https://arkime.com/>
  + CyberChef - <https://github.com/gchq/CyberChef>

SELKS là một đóng góp của Stamus Networks cho cộng đồng nguồn mở. Nó có sẵn miễn phí và được phát hành theo giấy phép GNU GPLv3, bản phân phối trực tiếp có sẵn dưới dạng file cài đặt ISO Debian hoặc thông qua Docker compose trên bất kỳ hệ điều hành Linux nào.

Chúng ta có thể download file image của nó ở đây: <https://www.stamus-networks.com/open-source/#selks>

Hoặc source, README, và wiki có thể được tìm thấy trên GitHub: <https://github.com/StamusNetworks/SELKS>

Trong nghiên cứu này, chúng em sẽ phân tích, tìm hiểu, xây dựng và triển khai hệ thống hoàn chỉnh phiên bản **SELKS\_DGA** – một hệ thống SELKS được dựa trên IDS/IPS platform tích hợp phát hiện Domain bất thường (cụ thể ở đây là DGA Domain) sử dụng machine learing, tự động lấy log và phân tích theo chu kì thời gian tùy chỉnh, đồng thời kết quả phân loại cũng như các cảnh báo được tạo ra sẽ được gửi lên Elasticsearch và TheHive4.

Các tính năng của SELKS\_DGA:

* Đảm bảo các tính năng cơ bản của một SELKS:
* Tối ưu hóa phần cứng và nhiều lớp dựa trên hệ điều hành Linux.
* Sử dụng Suricata IDPS.
* Hệ thống phát hiện xâm nhập với các rules cụ thể của công ty/ trang web/ bộ phận của bạn.
* Giao diện quản lý web: ruleset, tham số cấu hình.
* Phân tích nhật ký thông qua Elasticsearch có thể mở rộng với phương pháp tìm kiếm và tìm kiếm theo thời gian thực.
* Giám sát an ninh mạng: HTTP, DNS, TLS, SSH, Khai thác tệp, ghi nhật ký cảnh báo và nhiều tính năng khác.
* Các tính năng bổ sung:
* Ghi nhật kí và tự động phân tích, trích xuất domain từ log SELKS chu kì 10s.
* Tích hợp ML để phân loại domain được trích xuất theo chu kì.
* Gửi kết quả phân loại domain lên Elasticsearch Index.
* Tích hợp gửi cảnh báo domain bất thường lên TheHive.

# PHẦN TRIỂN KHAI

## DGA

* **Đào tạo mô hình**

Mô hình dự đoán nhị phân một domain có phải là DGA hay không, được phát triển dựa trên <https://github.com/sudo-rushil/dga-intel-web>, với 1 số thay đổi/cải tiến:

* Thay đổi ở phần load\_tf\_dataset: luôn luôn padding đến độ dài 82 thay vì padding đến độ dài của domain dài nhất => khắc phục lỗi dữ liệu đầu vào không tương thích với mô hình.
* Thêm tập validation\_data vào quá trình training để cải thiện khả năng dự đoán của mô hình.
* Thêm bảng thống kê tỉ lệ dương tính thật/giả, âm tính thật/giả để dễ dàng quan sát khả năng dự đoán của mô hình.
* Thêm tính năng lưu lại kết quả train để sử dụng cho các mô hình sau này.

Cách sử dụng:

Truy cập [https://colab.research.google.com/github/Tuan164/DGA/blob/main/Domain\_Classifier\_Model\_5\_3\_2.ipynb](https://colab.research.google.com/github/Tuan164/D%09GA/blob/main/Domain_Classifier_Model_5_3_2.ipynb) và chạy tuần tự các lệnh. Sau khi chạy xong, ta sẽ được file “trained\_model\_5\_3\_2.h5” sử dụng cho việc dự đoán các domain.

* **Áp dụng mô hình**

Phần này sử dụng kết quả của mô hình đào tạo, với đầu vào là 1 domain, và đầu ra là xác suất là DGA của domain đó, nếu xác suất ≥ 0.5 thì nó sẽ cho kết quả domain đó là “DGA”, ngược lại domain đó là “Genuine”. Mô hình được phát triển dựa trên <https://github.com/sudo-rushil/dgaintel>, với 1 số biến thể khác:

* Hỗ trợ API cho việc dự đoán. (dockerhub: 1642001/dga\_detect:latest)
* Kết hợp với SELKS để phân tích domain của người dùng trong mạng, từ đó gửi trả kết quả lại cho SELKS và gửi cảnh báo lên TheHive nếu phát hiện DGA. (dockerhub: 1642001/selks\_dga:latest)

Cách sử dụng đơn giản nhất là sử dụng chức năng API thông qua docker image:

docker pull 1642001/dga\_detect:latest

docker run -d -p 0.0.0.0:5000:5000 1642001/dga\_detect

curl http://127.0.0.1:5000/<domain\_name>

## SELKS

Quy trình xây dựng hệ thống SELKS\_DGA:

* 1. Cài đặt SELKS lên máy vật lí:

*Yêu cầu*

* 2 cores
* 9 GB RAM trống
* Dung lượng ổ đĩa trống tối thiểu 10 GB (dung lượng ổ đĩa thực tế sẽ chủ yếu phụ thuộc vào số lượng quy tắc và lưu lượng truy cập trên mạng). Nên dùng loại 200GB + SSD.
* git,curl
* docker > 17.06.0 (sẽ được cài đặt trong quá trình thiết lập ban đầu SELKS)
* docker-compose > 1.27.0 (sẽ được cài đặt trong quá trình thiết lập ban đầu SELKS)

Phiên bản này của SELKS dựa trên docker và nhằm cung cấp việc triển khai và quản lý dễ dàng hơn. Quy trình cài đặt ở đây hợp lệ cho bất kỳ hệ điều hành Linux nào như Debian/CentOS/RedHat/Ubuntu hỗ trợ docker.

git clone https://github.com/StamusNetworks/SELKS.git

cd SELKS/docker/

./easy-setup.sh

docker-compose up -d

Khi các vùng chứa đã được thiết lập và đang chạy, bạn chỉ nên trỏ trình duyệt của mình đến <https://your.selks.IP.here/>. Nếu chọn cài đặt Portainer trong khi cài đặt, cần phải truy cập <https://your.selks.IP.here:9443> để đặt mật khẩu quản trị của portainer.

* 1. Thêm 2 services tương ứng vào file /opt/SELKS/docker/docker-compose.yml:

thehive:

container\_name: thehive

image: thehiveproject/thehive4:latest

restart: ${RESTART\_MODE:-unless-stopped}

volumes:

- ./containers-data/thehive:/data

ports:

- 9000:9000

networks:

network:

dga:

container\_name: SELKS\_DGA

image: 1642001/selks\_dga

restart: ${RESTART\_MODE:-unless-stopped}

depends\_on:

- evebox

- elasticsearch

- logstash

- kibana

- thehive

volumes:

- ./containers-data/DGA:/PredictDGA

networks:

network:

* 1. Trong thư mục cùng cấp với file docker-compose, thư mục containers-data chứa các thư mục được mount với các thư mục bên trong các container của các dịch vụ. Tại đây, ta sẽ thêm 2 thư mục DGA và thehive. Đối với DGA, ta sẽ down về từ [SELKS\_DGA\_main.rar](https://uithcm-my.sharepoint.com/:u:/g/personal/19522463_ms_uit_edu_vn/EZT3pAlqqWNCitZcfFZcOjoBTKl6FYlePJHSWowf41U3Sg?e=Qd3rFO) và cấp quyền thực thi cho các file. Ngoài ra ta cần thay đổi IP trong file config.py[1] thành IP tương ứng với máy chạy dịch vụ SELKS. Còn thehive thì sau khi chạy dịch vụ, nó sẽ tự động tạo ra các file để quản lí cơ sở dữ liệu, vì vậy ta cần cấp quyền ghi cho mọi user với thư mục này.

Cách sử dụng:

Tại thư mục /opt/SELKS/docker/ (thư mục chứa file docker-compose.yml [2]) thực hiện lệnh sau

$ docker-compose up -d

**Thông tin đăng nhập để sử dụng.**

* Thông tin đăng nhập:
  + User: selks-user
  + Password: selks-user
  + Mật khẩu gốc mặc định là: StamusNetworks
* Chúng ta cần phải xác thực để có thể truy cập vào giao diện web. User/pass mặc định là *selks-user/selks-user*. Lưu ý rằng SELKS sử dụng *https,* chúng ta chỉ có thể truy cập vào giao diện web thông qua *https.*
* Một liên kết trang của Kibana cũng có thể được tìm thấy bằng cách nhấp vào biểu tượng Stamus ở bên trái phía trên của Sciriuscùng với đó là giao diện quản lý rules.
* Elasticsearch, Logstash và Suricata được xây dựng và có thể được sử dụng như các dịch vụ tiêu chuẩn, ví dụ:

*systemctl restart suricata*

*systemctl stop logstash*

* Thông tin đăng nhập TheHive:

Khởi chạy trình duyệt và kết nối với http://YOUR\_SERVER\_ADDRESS:9000/.

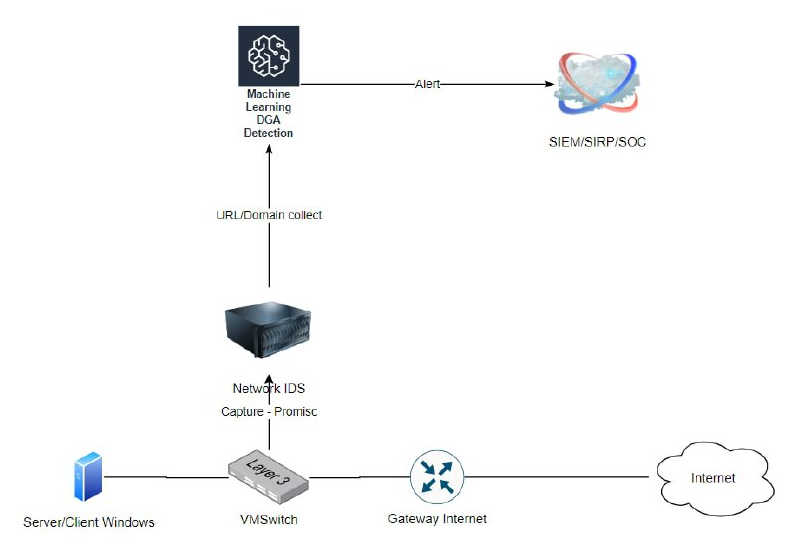
Người dùng quản trị mặc định là “admin@thehive.local” và có mật khẩu là “secret”.

**Remote máy chủ SELKS qua SSH.**

* Mặc định thì SSH được cho phép với tài khoản *selks-user* và mật khẩu là *selks-user.*

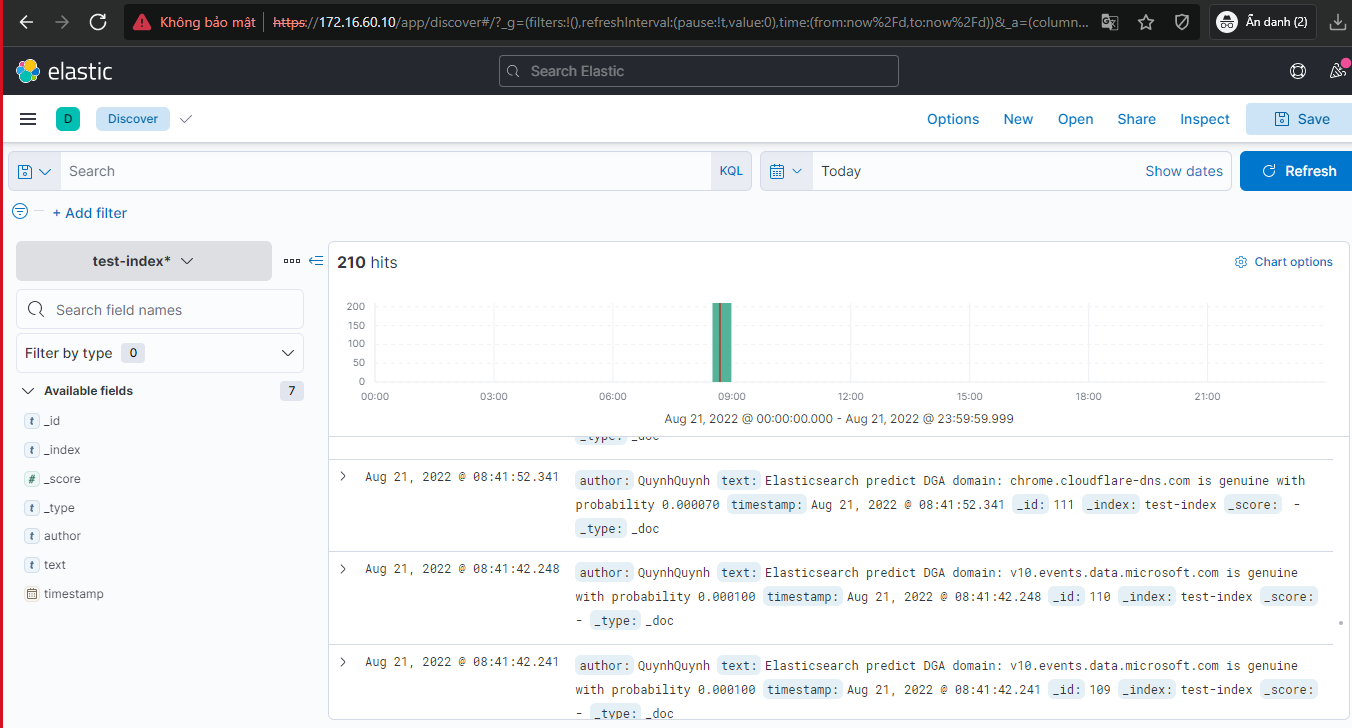
# DEMO

Mô hình demo:



Kết quả demo:

Sau mỗi 10s, hệ thống sẽ tự động request đến elasticsearch và lấy domain, sau đó nó sẽ phân tích domain đó và tự động gửi lại cho elasticsearch index. Nếu nó phát hiện DGA, nó sẽ gửi thêm alert đến thehive.



# PHỤ LỤC

[1] [config.py](https://uithcm-my.sharepoint.com/:u:/g/personal/19522463_ms_uit_edu_vn/Ef5jHDouxIJNhhmYHZoKwCIB640gR5dOkcRfLBbHWgwuKA): Cấu hình các tham số (IP, port, thời gian lặp…)

[2] [docker-compose.yml](https://uithcm-my.sharepoint.com/:u:/g/personal/19522463_ms_uit_edu_vn/EflsfyAH19ZHo6EcHt1vSdwBQ-4IhauHQ2GeSIlGrQoU_g): Khởi động các dịch vụ

[3] [SELKS\_DGA\_OVF.rar](https://uithcm-my.sharepoint.com/:u:/g/personal/19522463_ms_uit_edu_vn/EYMhoEaP5X1DidFonfuNFdgBc4KMb0ItSL0Uz55U_vP5LQ?e=Y6isnj): Các file hệ thống đã đóng gói

# TÀI LIỆU KHAM THẢO

[1] <https://github.com/sudo-rushil/dga-intel-web>

[2] <https://github.com/sudo-rushil/dgaintel>

[3] <https://www.kaggle.com/datasets/hydrobloquant/dgas-detection>

[4] <https://github.com/StamusNetworks/SELKS/>

[5] <https://docs.thehive-project.org/thehive/>

**HẾT**

1. [↑](#footnote-ref-2)