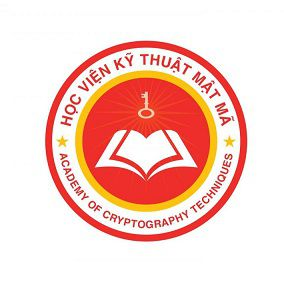
BAN CƠ YẾU CHÍNH PHỦ

**HỌC VIỆN KỸ THUẬT MẬT MÃ**

****

THỰC TẬP CƠ SỞ CHUYÊN NGÀNH

**NGHIÊN CỨU TRIỂN KHAI SURICATA VÀ ĐỀ XUẤT GIẢI PHÁP HỌC MÁY VÀO HỆ THỐNG**

|  |
| --- |
| Ngành: công nghệ thông tin  Chuyên ngành: An toàn thông tin |
| Sinh viên thực hiện:  Nguyễn Việt Hùng - AT140618  Hà Nội, 2020 |

# **Mục Lục**

[**Mục Lục**](#_gjdgxs) **……………………………………………………………………………………....2**

[**Danh mục bảng**](#_3znysh7) **……………………………………………………………………………………...**4

Danh mục hình vẽ……………………………………………………………………………………5

[**Lời cảm ơn**](#_tyjcwt) **…………………………………………………………………………………………...**6

[**Lời nói đầu**](#_3dy6vkm) **…………………………………………………………………………………………...**7

[**CHƯƠNG 1**](#_2s8eyo1). [**HỆ THỐNG PHÁT HIỆN VÀ NGĂN CHẶN XÂM NHẬP**](#_17dp8vu) **……………………….**9

[1.1. Hệ thống phát hiện xâm nhập IDS](#_3rdcrjn) ……………………………………………………9

[1.1.1 Khái niệm IDS](#_26in1rg) …………………………………………………………………..9

[1.1.2. Các kỹ thuật phát hiện xâm nhập](#_lnxbz9) ……………………………………………...10

[1.1.2.1. Signature-based Method](#_35nkun2) ………………………………………………...10

[1.1.2.2. Anomaly-based Method](#_1ksv4uv).………………………………………………..10

[1.2. Hệ thống ngăn chặn xâm nhập IPS](#_44sinio) ………………………………………………….10

[1.2.1. Khái niệm IPS](#_2jxsxqh) ………………………………………………………………....10

[1.2.2. Các phương pháp ngăn chặn xâm nhập mạng trái phép](#_z337ya) ………………………11

[**CHƯƠNG 2**](#_3j2qqm3). [**GIỚI THIỆU VỀ SURICATA**](#_1y810tw) **……………………………………………………...**12

[2.1. Giới thiệu chung về Suricata](#_4i7ojhp) ………………………………………………………...12

[2.2. Open Information Security Foundation](#_2xcytpi) ……………………………………………..12

[2.3. Sử dụng Suricata làm công cụ IDS và IPS](#_1ci93xb) ………………………………………….12

[2.4. Nhu cầu ứng dụng](#_3whwml4) …………………………………………………………………...13

[2.5. Lịch sử phát triển và các RFC liên quan](#_2bn6wsx) ……………………………………………13

[2.5.1. Lịch sử phát triển](#_qsh70q) ……………………………………………………………...13

2.5.2. Các RFC liên quan………………………………………………………………….14

[2.6. Kiến trúc xử lý của Suricata](#_3as4poj) ………………………………………………………...14

[**CHƯƠNG 3**](#_49x2ik5). [**RULE TRONG SURICATA**](#_2p2csry) **……………………………………………………….**16

[3.1. Giới thiệu](#_147n2zr) ……………………………………………………………………………16

[3.2. Rule Format](#_3o7alnk) …………………………………………………………………………16

[3.2.1. Action](#_23ckvvd) ………………………………………………………………………….16

[3.2.1.1. Pass](#_ihv636) ……………………………………………………………………...16

[3.2.1.2. Drop](#_32hioqz) ……………………………………………………………………..17

[3.2.1.3. Reject](#_1hmsyys) …………………………………………………………………....17

[3.2.1.4. Alert](#_41mghml) ……………………………………………………………………..17

[3.2.2. Protocol](#_2grqrue) ………………………………………………………………………..17

[3.2.3. Source and destination](#_vx1227) ………………………………………………………...19

[3.2.4. Ports (source and destination)](#_3fwokq0) ………………………………………………....20

[3.2.5. Direction](#_dza3a39gef3f) ……………………………………………………………………....21

[3.2.6. Rule options](#_4f1mdlm) …………………………………………………………………...23

[3.2.6.1. msg (message)](#_2u6wntf) …………………………………………………………..23

[3.2.6.2. reference](#_19c6y18) ………………………………………………………………....23

[3.2.6.3. gid](#_3tbugp1) ……………………………………………………………………….23

[3.2.6.4. sid](#_28h4qwu) ………………………………………………………………………..24

[3.2.6.5. rev](#_nmf14n) ……………………………………………………………………….24

[3.2.6.6. classtype](#_37m2jsg) ………………………………………………………………....24

[3.2.6.7. priority](#_1mrcu09) …………………………………………………………………..26

[3.2.6.8. metadata](#_46r0co2) ………………………………………………………………....26

[3.2.6.9. target](#_2lwamvv) …………………………………………………………………….27

[**CHƯƠNG 4**](#_3l18frh). [**CÀI ĐẶT VÀ CẤU HÌNH SURICATA**](#_206ipza) **………………………………………....**28

[4.1. Giới thiệu ELK](#_4k668n3) ……………………………………………………………………...28

[4.2. Mô hình triển khai](#_1egqt2p) …………………………………………………………………...28

[4.2.1. Mô hình LAB](#_2z48ci9yu7y2) ………………………………………………………………….28

[4.2.2. IP Planning](#_q2dxni4gnzox) …………………………………………………………………….29

[4.3. Cài đặt Suricata sử dụng ELK làm công cụ thu thập Log](#_3ygebqi) …………………………..30

[4.3.1. Cài đặt Suricata](#_2dlolyb) ………………………………………………………………..30

[4.3.1.1. Cài đặt](#_sqyw64) …………………………………………………………………...30

[4.3.1.2. Cấu hình tự động cập nhật rule](#_1rvwp1q) ………………………………………….31

[4.3.1.3. Cấu hình suricata](#_2r0uhxc) ………………………………………………………..32

[4.3.1.4. Cấu hình filebeat đẩy log về ELK](#_1664s55) ……………………………………....34

[4.4. Thử nghiệm các kiểu tấn công](#_3q5sasy) ……………………………………………………....35

[4.4.1. Ping of Death](#_25b2l0r) ………………………………………………………………….35

[4.4.2. Scan password](#_kgcv8k) ………………………………………………………………....36

[4.4.3. SQL Injection](#_34g0dwd) ………………………………………………………………….36

[**CHƯƠNG 5**](#_2iq8gzs). [**MỘT SỐ GIẢI PHÁP HỌC MÁY CHO HỆ THỐNG SURICATA**](#_xvir7l) **……………..**38

[5.1. OPNids](#_3hv69ve) ……………………………………………………………………………....38

[5.1.1. Hoạt động của OPNids](#_1x0gk37) ………………………………………………………..38

[5.2. Machine Learning Toolkit sử dụng để phân tích Botnets](#_4h042r0) ……………………………..38

[5.2.1. Sử dụng MLTK để phân tích quần thể botnet](#_1baon6m) …………………………………...38

[5.3. Sử dụng lập trình](#_2afmg28) ………………………………………………………………………….39

[**Kết luận**](#_39kk8xu) **……………………………………………………………………………………………...**40

[**Tài liệu tham khảo**](#_2nusc19) **…………………………………………………………………………………**41

[**PHỤ LỤC**](#_2250f4o) **…………………………………………………………………………………………….42**

### 

# 

# **Danh mục bảng**

# 

|  |
| --- |
| Bảng 3.1. Các toán tử sử dụng để kết hợp với source and destination……………..18  Bảng 3.2. Mẫu khai báo ip kết hợp với các toán tử của rule trong Suricata………..19  Bảng 3.3. Các toán tử sử dụng để kết hợp với port.………………………………...20  Bảng 3.4. Mẫu khai báo port kết hợp với các toán tử của rule trong suricata……...20  Bảng 3.5. Một số hệ thống nhận dạng tấn công……………………………………22  Bảng 3.6. Các loại tấn công và mức độ nghiêm trọng……………………….…...24  Bảng 3.7. Cặp khóa - giá trị trong metadata………………………………………..26 |

# **Danh mục hình vẽ**

|  |
| --- |
| Hình 2.1. Kiến trúc tổng quan của suricata………………………………………14  Hình 3.1. Minh họa hướng đi của gói tin trong mạng…………………………...22  Hình 4.1. Hình 4.1. Mô hình triển khai Suricata trên KVM…………………….29  Hình 4.2. Hình 4.2. Mô hình IP planning cho các máy ảo………………………29  Hình 8.1 Hình 8.1. Mẫu dữ liệu thống kê của Suricata khi đẩy lên ELK………..45 |

# **Lời cảm ơn**

Trong quá trinh thực hiện thực tập cơ sở này, chúng em đã nhận được sự giúp đỡ tận tình của cán bộ hướng dẫn là thầy TS. Phạm Văn Hưởng – Khoa Công nghệ thông tin Học viện Kỹ Thuật Mật Mã, ..

Chúng em xin gửi lời cảm ơn đến thầy và những những người đã tạo điều kiện tốt nhất để chúng em hoàn thành thực tập cơ sở này !

**Sinh viên thực hiện**

**Nguyễn Việt Hùng**

**Trần Thành Vinh**

**Nguyễn Hồng Quân**

# **Lời nói đầu**

# Xã hội ngày càng phát triển, Internet trở thành 1 phần không thể thiếu đối với từng cá nhân, doanh nghiệp, các tổ chức, trường học cũng như chính phủ. Internet du nhập vào Việt Nam được hơn 15 năm, đã trở thành công cụ, phương thức giúp cho các doanh nghiệp tiếp cận với khách hàng, cung cấp dịch vụ, quản lý dữ liệu của tổ chức một cách hiệu quả và nhanh chóng.

Cùng với sự phát triển mạnh mẽ của internet, các cuộc tấn công và xâm nhập mạng của những kẻ xấu vì thế cũng phát triển theo. Không chỉ trên thế giới mà ở Việt nam, vấn đề "an toàn thông tin" đã và đang trở thành vấn đề nóng bỏng. Sự đa dạng và phức tạp trong các loại hình tấn công đã gây ra nhiều khó khăn cho việc ngăn chặn và phòng chống.

Một hệ thống phòng chống và phát hiện xâm nhập sẽ giúp người quản trị có thể luôn luôn theo dõi và thu thập nhiều thông tin đáng giá cho quá trình chống lại các hình thức tấn công và xâm nhập đó.

Xuất phát từ những lý do nêu trên, chúng em chọn đề tài "Nghiên cứu triển khai Suricata và đề xuất giải pháp học máy vào hệ thống". Chuyên đề được thực hiện với mục đích tìm hiểu về Suricata để xây dựng hệ thống IDS/IPS cho các máy bên trong mạng máy tính.

Mục tiêu đặt ra khi thực hiện chuyên đề là:

1. Tìm hiểu về Suricata, nắm bắt được cách suricata hoạt động như một IDS hoặc như một IPS.

2. Nắm bắt cấu trúc rule của Suricata để triển khai và phát triển.

3. Triển khai Suricata làm IDS cho hệ thống mạng máy tính.

4. Đề xuất một số giải pháp học máy vào hệ thống.

Sau thời gian thực hiện, các mục tiêu về cơ bản đã đạt được. Tuy nhiên, để có thể triển khai và vận hành hệ thống IDS một cách trơn tru thực sự cần một lượng kiến thức về mạng máy tính rất sâu rộng và thời gian để trải qua các bài test cho hệ thống. Vì thời gian thực hiện tương đối ngắn nên có nhiều chỗ không tránh khỏi thiếu sót, em rất mong nhận được sự góp ý của thầy cô cũng như các bạn để chuyên đề này được hoàn thiện nhất có thể.

# 

# **CHƯƠNG 1**

# **HỆ THỐNG PHÁT HIỆN VÀ NGĂN CHẶN XÂM NHẬP**

## **1.1. Hệ thống phát hiện xâm nhập IDS**

Kỹ thuật phát hiện xâm nhập không phải là 1 kỹ thuật mới. Vì nó đã được áp dụng nhiều trong các lĩnh vực khác nhau chứ không chỉ riêng lĩnh vực an toàn thông tin của mạng máy tính. Ví dụ đơn giản nhất có thể thấy về kỹ thuật phát hiện xâm nhập đó là hệ thống cảnh báo bằng chuông báo động của một chiếc xe hay một ổ khóa. Nó hoạt động trên nguyên tắc rất đơn giản đó là khi ta bật chế độ cảnh báo, khi có người chạm vào chiếc xe hay ổ khóa đó, nó sẽ phát ra chuông cảnh báo để báo động rằng có người định xâm nhập trái phép vào chiếc xe hay ổ khóa.

Hệ thống phát hiện xâm nhập cũng tương tự như vậy, nó được xây dựng để bảo vệ tài nguyên hệ thống mạng trước những attacker. Do vậy, ta cần xây dựng một hệ thống IDS để phát hiện các dấu hiệu bất thường, cảnh báo khi có biểu hiện bất thường và giám sát các hoạt động ra vào hệ thống để phân tích và ngăn chặn kịp thời.

### **1.1.1 Khái niệm IDS**

IDS (Intrusion Detection System ) là thiết bị hoặc phần mềm có nhiệm vụ giám sát traffic mạng, các hành vi đáng ngờ và cảnh báo cho admin hệ thống. Mục đích của IDS là phát hiện và ngăn ngừa các hành động phá hoại bảo mật hệ thống, hoặc những hành động trong tiến trình tấn công như dò tìm, quét các cổng. IDS cũng có thể phân biệt giữa những cuộc tấn công nội bộ (từ chính nhân viên hoặc khách hàng trong tổ chức) và các cuộc tấn công bên ngoài (từ hacker). Trong 1 số trường hợp, IDS có thể phản ứng lại các traffic bất thường bằng cách chặn địa chỉ IP nguồn hoặc người dùng truy cập.

### **1.1.2. Các kỹ thuật phát hiện xâm nhập**

#### 1.1.2.1. Signature-based Method

IDS phát hiện các cuộc tấn công dựa trên kiến thức, liên quan đến việc tìm kiếm các chữ ký cụ thể, tổ hợp byte mà khi chúng xảy ra các tình huống như vậy thường là 1 hành vi độc hại.

Vì IDS dựa trên chữ ký đã biết trước có thể dễ dàng phát hiện các cuộc tấn công có mẫu chữ ký đã tồn tại trước đó nhưng đối với việc phát hiện các cuộc tấn công phần mềm độc hại mới rất khó phát hiện vì mẫu chữ ký của các cuộc tấn công này chưa được biết đến.

#### 1.1.2.2. Anomaly-based Method

Anomaly-based IDS được giới thiệu để phát hiện các cuộc tấn công phần mềm độc hại không xác định khi phần mềm độc hại mới được phát triển nhanh chóng. Giải pháp này theo dõi trong phạm vi cụ thể để tìm kiếm các trường hợp của hành vi độc hại (các trường hợp của hành vi độc hại do người phát triển định nghĩa). Đây là 1 công việc khó khăn, đôi khi có thể dẫn đến phát hiện xâm nhập giả.

## **1.2. Hệ thống ngăn chặn xâm nhập IPS**

Nếu hệ thống phát hiện xâm nhập (IDS) có khả năng phát hiện và cảnh báo các xâm nhập độc hại thì hệ thống ngăn chặn xâm nhập cũng có các khả năng đó nhưng đồng thời, nó sẽ đưa ra hành động ngăn chặn đối với các lưu lượng xấu đó. Vậy nên để hệ thống không chặn nhầm những lưu lượng tin cậy, ta cần triển khai IPS một cách thận trọng.

### **1.2.1. Khái niệm IPS**

Hệ thống ngăn chặn xâm nhập (IPS) là một dạng an ninh mạng hoạt động để phát hiện và ngăn chặn các mối đe dọa đã được xác định. Hệ thống ngăn chặn xâm nhập liên tục giám sát mạng của bạn, tìm kiếm các sự cố nguy hiểm có thể xảy ra và nắm bắt thông tin về chúng. IPS báo cáo các sự kiện này cho quản trị viên hệ thống và thực hiện hành động ngăn chặn, chẳng hạn như đóng các điểm truy cập và cấu hình tường lửa để ngăn chặn các cuộc tấn công trong tương lai. Các giải pháp IPS cũng có thể được sử dụng để xác định các vấn đề với chính sách bảo mật của công ty, ngăn chặn nhân viên và khách hàng trong mạng vi phạm các quy tắc mà các chính sách này có.

### **1.2.2. Các phương pháp ngăn chặn xâm nhập mạng trái phép**

Hệ thống ngăn chặn xâm nhập thường được cấu hình để sử dụng một số phương pháp khác nhau nhằm bảo vệ mạng khỏi bị truy cập trái phép. Bao gồm các phương pháp sau:

* Dựa trên chữ ký - Phương pháp dựa trên chữ ký sử dụng chữ ký xác định trước của các mối đe dọa mạng nổi tiếng. Khi một cuộc tấn công được bắt đầu khớp với một trong các chữ ký hoặc mẫu này, hệ thống sẽ thực hiện hành động cần thiết.
* Dựa trên sự bất thường - Phương pháp dựa trên sự bất thường giám sát mọi hành vi bất thường hoặc không mong muốn trên mạng. Nếu phát hiện sự bất thường, hệ thống sẽ chặn quyền truy cập vào máy chủ mục tiêu ngay lập tức.
* Dựa trên chính sách - Cách tiếp cận này yêu cầu quản trị viên cấu hình các chính sách bảo mật theo chính sách bảo mật của tổ chức và cơ sở hạ tầng mạng. Khi một hoạt động xảy ra vi phạm chính sách bảo mật, một cảnh báo sẽ được kích hoạt và gửi đến quản trị viên hệ thống.

# **CHƯƠNG 2**

# **GIỚI THIỆU VỀ SURICATA**

## **2.1. Giới thiệu chung về Suricata**

Suricata là một hệ thống phát hiện xâm nhập dựa trên mã nguồn mở. Nó được phát triển bởi Open Information Security Foundation (OISF). Công cụ này được phát triển không nhằm cạnh tranh hay thay thế các công tụ hiện có, nhưng nó sẽ mang lại những ý tưởng và công nghệ mới trong lĩnh vực an ninh mạng.

## **2.2. Open Information Security Foundation**

OISF là một tổ chức phi lợi nhuận nhằm mục đích xây dựng một cộng đồng hỗ trợ xung quanh các công nghệ bảo mật nguồn mở. Nó được thành lập vào năm 2008 và được dẫn dắt bởi một nhóm các lập trình viên, chuyên gia bảo mật và những người khác có chung niềm đam mê với bảo mật mã nguồn mở. OISF nhằm mục đích cung cấp các công nghệ bảo mật mạnh mẽ và phù hợp để giúp chống lại một loạt các mối đe dọa. Dự án chính của OISF là Suricata, nhưng họ cũng hỗ trợ và tham gia với cộng đồng thông qua một loạt các hội nghị, nhóm làm việc và các sự kiện đào tạo.

Tổ chức đã thiết lập quan hệ đối tác với Emerging Threats và các nhóm có lợi thế chiến lược khác để giúp cung cấp thông tin cập nhật và bộ quy tắc toàn diện cho công cụ của họ. OISF được tài trợ bởi Bộ An ninh Nội địa Hoa Kỳ (DHS) cũng như tập đoàn OISF, bao gồm nhiều công ty tư nhân. Để đáp lại sự hỗ trợ của họ, các công ty này được cấp giấy phép giới hạn không phải GPL

cho động cơ Suricata.

## **2.3. Sử dụng Suricata làm công cụ IDS và IPS**

Suricata có thể được sử dụng trong ba vai trò chính. Đơn giản nhất là thiết lập nó như một IDS dựa trên máy chủ, theo dõi lưu lượng của máy tính cá nhân. Điều này không hiệu quả hoặc thực tế, nhưng đó là cách tuyệt vời nếu là người mới tiếp cận với Suricata. Với khả năng là 1 IDS, Suricata có thể giám sát tất cả lưu lượng truy cập đi qua mạng và thông báo cho quản trị viên khi gặp bất kỳ điều gì độc hại. Khi Suricata được thiết lập như 1 IPS, nó có thể giám sát lưu lượng đến và đi. Nó có thể ngăn chặn lưu lượng độc hại trước khi xâm nhập vào mạng, cũng như cảnh báo cho quản trị viên.

Thiết lập Suricata làm IPS có vẻ là lựa chọn rõ ràng nhất vì thực ra chúng ta muốn ngăn chặn lưu lượng độc hại hơn là đơn thuần chỉ theo dõi chúng. Nhưng việc triển khai IDS thực sự không hề đơn giản như vậy, nếu không được định cấu hình đúng cách, rất có thể nó sẽ chặn cả các lưu lượng truy cập hợp lệ. Vậy nên để điều chỉnh các quy tắc khi sử dụng Suricata làm IPS thực sự là điều rất khó khăn. Nếu muốn sử dụng Suricata làm IPS, trước hết bạn nên sử dụng nó làm IDS để đảm bảo các quy tắc đó sẽ hoạt động đúng trong 1 khoảng thời gian dài.

## **2.4. Nhu cầu ứng dụng**

Với sự phát triển ngày càng nhiều các hình thức tấn công của tội phạm mạng(Cyber Crimes) hiện nay thì đối với một hệ thống mạng của doanh nghiệp, vấn đề an ninh phải được đặt lên hàng đầu.

Sử dụng một tường lửa với những công nghệ hiện đại và các chức năng có thể ngăn chặn các cuộc tấn công của các hacker là một giải pháp hữu hiệu cho các doanh nghiệp trong việc phát hiện, ngăn chặn các mối đe dọa nguy hiểm để bảo vệ hệ thống mạng.

## **2.5. Lịch sử phát triển và các RFC liên quan**

### **2.5.1. Lịch sử phát triển**

Vào tháng 12 năm 2009, một phiên bản beta đã được phát hành. Bản chuẩn đầu tiên phát hành tiếp theo vào tháng 7 năm 2010.

**2.5.2. Các RFC liên quan**

Jonkman, Matt (2009-12-31). “Suricata IDS Available for Download!”.

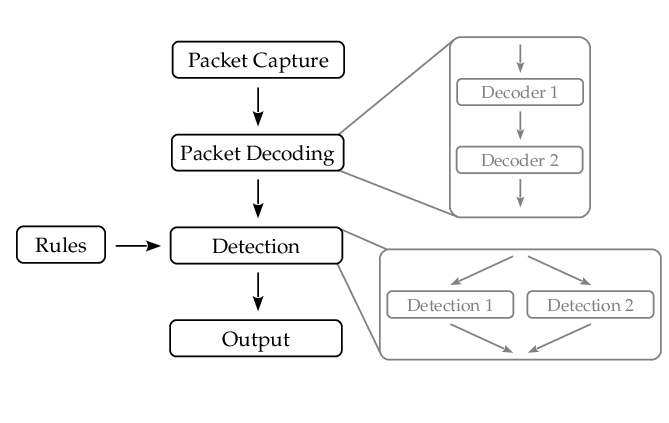
Seclists.org. Retrieved 2011-11-8.

“Suricata Features”. Retrieved 2012-10-06.

“Suricata All Features”. Retrieved 2012-10-06.

## **2.6. Kiến trúc xử lý của Suricata**

Kiến trúc của Suricata tương tự kiến trúc của Snort với một số sự khác biệt sau. Những gì tương ứng với phần tiền xử lý trong kiến trúc của Snort được chia làm hai trong Suricata, decoding và detection. Các module Decoding thêm thông tin và biểu diễn bên trong của các gói trong Suricata. Các module Detection dựa vào đại diện nội bộ này và các từ khóa chuyên nghiệp để sử dụng các quy tắc. Tổng quan về kiến trúc Suricata được thể hiện trong hình 2.1



*Hình 2.1. Kiến trúc tổng quan của suricata*

Đầu tiên, mỗi gói tin được xử lý trong các function Decoding, sau đó nó được xử lý bởi các module Detection. Các chức năng Decoding đọc gói tin và lưu dữ liệu đã giải mã vào 1 gói gửi lại nội bộ. Các chức năng Decoding được gọi cùng lúc trên gói. Có thể mở rộng chức năng Decoding bằng cách triển khai 1 chức năng giải mã mới và đặt nó vào quy trình giải mã.

Sau khi trải qua quá trình Decoding, các gói thông qua được Detection. Detection được điều chỉnh bởi các Rules và phụ thuộc vào bước Decoding. Khi các gói tin match với các rule được đưa ra, nó sẽ được xử lý để cảnh báo hoặc chặn gói tin đó, cũng có thể là bỏ qua ..v.v. sau đó đưa các thông báo đó ra output.

Suricata được viết bằng C vậy nên để phát triển các module cho Suricata cũng phải được viết bằng C. Nó không hỗ trợ C ++, vì 1 số điều trên nên nó khiến cho Suricata không phải là 1 công cụ tạo mẫu tốt nhất hiện có.

Suricata không nhanh bằng Snort trên máy tính sử dụng single-CPU. Tuy nhiên, nó được thiết kế để mở rộng quy mô trên các máy tính có hàng chục CPU. Cách tiếp cận đa luồng của nó khác với Snort. Suricata không yêu cầu cân bằng tải vì nó tự quản lý đa luồng trong khi đa luồng trong Snort được phân chia giám sát thành nhiều phần, mỗi phần được xử lý riêng lẻ. Ứng dụng này làm cho nó thân thiện hơn với người dùng.

# 

# **CHƯƠNG 3**

# **RULE TRONG SURICATA**

# **3.1. Giới thiệu**

## **3.2. Rule Format**

Signatures đóng một vai trò rất quan trọng trong Suricata. Trong hầu hết các trường hợp, mọi người đang sử dụng các rule có sẵn. Rule/signature bao gồm những điều sau:

* Action : Xác định những gì sẽ xảy ra khi chữ ký được khớp
* Header: Xác định giao thức, địa chỉ IP, Port và hướng của quy tắc
* Rule options: Xác định chi tiết cụ thể của rule

Dưới đây là ví dụ về rule:

|  |
| --- |
| drop tcp $HOME\_NET any -> $EXTERNAL\_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”; flow:established,to\_server; flowbits:isset,is\_proto\_irc; content:”NICK “; pcre:”/NICK .\*USA.\*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124; rev:2;) |

Trong ví dụ trên, "drop" là action, "tcp $HOME\_NET any -> $EXTERNAL\_NET any" là header và phần còn lại là các Options.

### **3.2.1. Action**

Tất cả các chữ ký đều có các thuộc tính khác nhau. Một trong số đó là thuộc tính Action. Điều này xác định điều gì sẽ xảy ra khi 1 chữ ký khớp. Có 4 loại Action. Dưới đây là phần tóm tắt về những gì xảy ra khi chữ khớp và chứa 1 trong những hành động đó:

#### 3.2.1.1. Pass

Nếu một chữ ký khớp và chứa pass, Suricata sẽ dừng quét gói và bỏ qua phần cuối của tất cả các quy tắc (chỉ dành cho gói hiện tại).

#### 3.2.1.2. Drop

Điều này liên quan đến chế độ IPS/inline. Nếu chương trình tìm thấy 1 chữ ký phù hợp mà rule chứa chữ ký đó có chứa drop, nó sẽ dừng ngay lập tức. Gói tin sẽ không được gửi đi nữa. Người nhận không được thông báo về những gì đang xảy ra, dẫn đến hết thời gian chờ (chắc chắn với TCP). Suricata tạo ra 1 cảnh báo cho gói tin này.

#### 3.2.1.3. Reject

Đây là 1 hoạt động từ chối gói tin, cả người nhận và người gửi đều nhận được 1 gói tin từ chối. Có hai loại gói tin "reject" sẽ được chọn tự động. Nếu gói vi phạm liên quan đến TCP, nó sẽ là Reset-packet. Đối với tất cả các giao thức khác, nó sẽ là 1 gói lỗi ICMP. Suricata cũng tạo ra 1 cảnh báo. Khi ở chế độ Inline/IPS, gói tin vi phạm cũng sẽ bị loại bỏ giống như hành động 'drop'.

#### 3.2.1.4. Alert

Nếu một chữ ký khớp và chứa cảnh báo, gói tin sẽ được coi như bất kỳ gói tin không đe dọa nào khác, ngoại trừ gói tin này, một cảnh báo sẽ được tạo bởi suricata. Chỉ quản trị viên hệ thống mới có thể nhận thấy cảnh báo này.

Inline/IPS có thể ngăn chặn lưu lượng mạng theo 2 cách. Một cách là drop và cách còn lại là reject.

Các quy tắc sẽ được tải theo thứ tự xuất hiện trong tệp. Nhưng chúng sẽ được xử lý theo 1 thứ tự khác. Chữ ký có các ưu tiên khác nhau, các chữ ký quan trọng nhất sẽ được quét trước. Thứ tự mặc định là: pass, drop, reject, alert.

### **3.2.2. Protocol**

|  |
| --- |
| drop **TCP** $HOME\_NET any -> $EXTERNAL\_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”; flow:established,to\_server; flowbits:isset,is\_proto\_irc; content:”NICK “; pcre:”/NICK .\*USA.\*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124; rev:2;) |

Từ khóa này trong chữ ký cho Suricata biết nó liên quan đến giao thức nào. Bạn có thể chọn giữa bốn giao thức cơ bản:

* tcp (for tcp-traffic)
* udp
* icmp
* ip (ip stands for 'all' or 'any')

Ngoài ra còn có 1 số giao thức ở tầng application, hay còn gọi là giao thức layer 7 mà ta có thể chọn. Đó là:

* http
* ftp
* tls
* smb
* dns
* ssh
* smtp
* imap
* msn
* modbus
* dnp3
* enip
* nfs
* ikev2
* krb5
* ntp
* dhcp

Tính khả dụng của các giao thức này tùy thuộc vào các giao thức đó có được bật tại tệp cấu hình "suricata.yaml" hay không.

Nếu bạn có chữ ký (chẳng hạn) là http, suricata đảm bảo rằng chữ ký chỉ có thể khớp nếu nó liên quan đến lưu lượng truy cập http.

### **3.2.3. Source and destination**

|  |
| --- |
| drop tcp **$HOME\_NET** any -> **$EXTERNAL\_NET** any (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”; flow:established,to\_server; flowbits:isset,is\_proto\_irc; content:”NICK “; pcre:”/NICK .\*USA.\*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124; rev:2;) |

Phần in đậm đầu tiên là Source, phần in đậm tiếp theo là Destination (Lưu ý: Hướng mũi tên là hướng đi của gói tin. Tức là gói tin đi từ home\_net ra bên ngoài mạng).

Với source và destination, ta chỉ định nguồn lưu lượng truy cập và đích của lưu lượng truy cập tương ứng.

Có thể gán các địa chỉ IP (cả IPV4 và IPV6 đều được hỗ trợ). Chúng có thể được kết hợp với các toán tử. Các toán tử được kết hợp thể hiện trong bảng 3.1.

*Bảng 3.1. Các toán tử sử dụng để kết hợp với source and destination*

|  |  |
| --- | --- |
| **Operator** | **Description** |
| : | port ranges |
| ! | exception/negation |
| [.., ..] | grouping |

Thông thường, bạn cũng sẽ sử dụng các biến, chẳng hạn như **$HOME\_NET** và **$EXTERNAL\_NET**. Tệp cấu hình chỉ định các địa chỉ IP mà ta quan tâm và các cài đặt này sẽ được sử dụng thay cho các biến trong các rule. Các ví dụ và ý nghĩa của nó được thể hiện trong bảng 3.2.

*Bảng 3.2. Mẫu khai báo ip kết hợp với các toán tử của rule trong Suricata*

|  |  |
| --- | --- |
| **Example** | **Meaning** |
| ! 1.1.1.1 | Every IP address but 1.1.1.1 |
| ![1.1.1.1, 1.1.1.2] | Every IP address but 1.1.1.1 and 1.1.1.2 |
| $HOME\_NET | Your setting of HOME\_NET in yaml |
| [$EXTERNAL\_NET, !$HOME\_NET] | EXTERNAL\_NET and not HOME\_NET |
| [10.0.0.0/24, !10.0.0.5] | 10.0.0.0/24 except for 10.0.0.5 |
| […, [….]] |  |
| […, ![…..]] |  |

### **3.2.4. Ports (source and destination)**

|  |
| --- |
| drop tcp $HOME\_NET **any** -> $EXTERNAL\_NET **any** (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”; flow:established,to\_server; flowbits:isset,is\_proto\_irc; content:”NICK “; pcre:”/NICK .\*USA.\*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124; rev:2;) |

Phần in đậm đầu tiên chính là source port, phần in đậm tiếp theo là destination port. Lưu lượng ra vào qua các port. Port mặc định cho HTTP là 80 trong khi đó HTTPS là 443.

Các port như HTTP, HTTPS, .. thường là các destination port. Source ports, tức là port tại ứng dụng đã gửi gói tin thường được hệ điều hành gán một cổng ngẫu nhiên.

Trong khi thiết lập các port, có thể sử dụng các toán tử đặc biệt. Xem bảng 3.3 và 3.4 để hiểu cách khai báo các port trong rule.

*Bảng 3.3. Các toán tử sử dụng để kết hợp với port*

|  |  |
| --- | --- |
| **Operator** | **Description** |
| : | port ranges |
| ! | exception/negation |
| [.., ..] | grouping |

*Bảng 3.4. Mẫu khai báo port kết hợp với các toán tử của rule trong suricata*

|  |  |
| --- | --- |
| **Example** | **Meaning** |
| [80, 81, 82] | port 80, 81 and 82 |
| [80: 82] | Range from 80 till 82 |
| [1024: ] | From 1024 till the highest port-number |
| !80 | Every port but 80 |
| [80:100,!99] | Range from 80 till 100 but 99 excluded |
| [1:80,![2,4]] | Range from 1-80, except ports 2 and 4 |
| [.., [..,..]] |  |

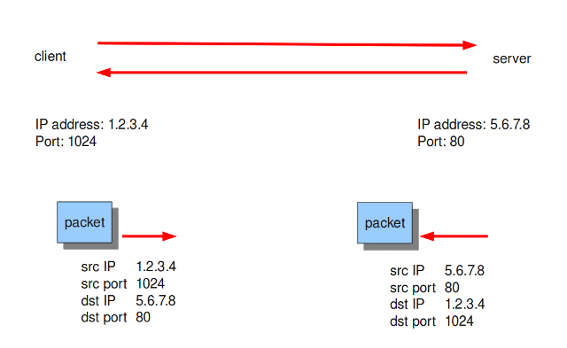
### 

### **3.2.5. Direction**

|  |
| --- |
| drop tcp $HOME\_NET any **->** $EXTERNAL\_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”; flow:established,to\_server; flowbits:isset,is\_proto\_irc; content:”NICK “; pcre:”/NICK .\*USA.\*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124; rev:2;) |

Direction cho biết chữ ký phải khớp theo cách nào. Gần như mọi chữ ký đều có mũi tên về phía tay phải (->). Nó có nghĩa là chỉ các gói có cùng hướng mới có thể khớp. Tuy nhiên, cũng có thể có 1 quy tắc phù hợp với cả 2 cách (<>).

Hình 4.1 sẽ minh họa cho ta thấy hướng đi của traffic . Giả sử có 1 máy khách có địa chỉ 1.2.3.4 và cổng 1024, và 1 máy chủ có địa chỉ 5.6.7.8, lắng nghe trên cổng 80 (HTTP). Máy khách gửi 1 tin nhắn đến máy chủ và máy chủ sẽ trả lời cùng câu trả lời của nó.



*Hình 3.1. Minh họa hướng đi của gói tin trong mạng*

Bây giờ, ta sẽ có 1 quy tắc như sau:

|  |
| --- |
| alert tcp 1.2.3.4 1024 -> 5.6.7.8 80 |

Chỉ có packet phía bên trái khớp với quy tắc trên, hướng chỉ định của gói tin phía bên phải chỉ ra rằng nó không khớp với quy tắc trên.

### **3.2.6. Rule** **options**

#### 3.2.6.1. msg (message)

Từ khóa msg cung cấp thông tin dạng văn bản về chữ ký và cảnh báo có thể có.

Format:

|  |
| --- |
| msg:"<message text>"; |

#### 3.2.6.2. reference

Từ khóa reference cho phép các quy tắc bao gồm các reference đến hệ thống nhận dạng tấn công bên ngoài. Plugin hiện hỗ trợ một số hệ thống cụ thể cũng như các URL duy nhất. Plugin này sẽ được sử dụng bởi các plugin đầu ra để cung cấp liên kết đến thông tin bổ sung về cảnh báo được tạo. Một số hệ thống được hỗ trợ kèm theo các url của hệ thống được thể hiện trong bảng 3.5.

*Bảng 3.5. Một số hệ thống nhận dạng tấn công*

|  |  |
| --- | --- |
| **System** | **URL Prefix** |
| bugtraq | http://www.securityfocus.com/bid/ |
| cve | http://cve.mitre.org/cgi-bin/cvename.cgi?name= |
| nessus | http://cgi.nessus.org/plugins/dump.php3?id= |
| arachnids | (currently down) http://www.whitehats.com/info/IDS |
| mcafee | http://vil.nai.com/vil/content/v\_ |
| osvdb | http://osvdb.org/show/osvdb/ |
| msb | http://technet.microsoft.com/en-us/security/bulletin/ |
| url | http:// |

Format:

|  |
| --- |
| reference:<id system>, <id>; [reference:<id system>, <id>;] |

#### 3.2.6.3. gid

Từ khóa gid có thể được sử dụng để cung cấp cho các nhóm chữ ký khác nhau một giá trị id khác. Suricata sử dụng gid mặc định 1. Có thể sửa đổi điều này. Thông thường nó sẽ được thay đổi và việc thay đổi nó không có ý nghĩa kỹ thuật. Bạn chỉ có thể nhận thấy nó trong cảnh báo.

Format:

|  |
| --- |
| gid:<generator id>; |

#### 3.2.6.4. sid

Từ khóa sid được sử dụng để xác định duy nhất các quy tắc Suricata. Thông tin này cho phép các plugin đầu ra xác định các quy tắc một cách dễ dàng. Tùy chọn này nên được sử dụng với từ khóa rev.

* 0 - 1000000: Dành riêng cho VRT Sourcefire
* 2000001-2999999: Được sử dụng bởi Emerging Threats
* 3000000 +: Sử dụng cho công cộng

Để tránh xung đột, bạn nên sử dụng các sid trên 3000000.

Format:

|  |
| --- |
| sid:<snort rules id>; |

#### 3.2.6.5. rev

Tùy chọn rev được sử dụng để biểu thị thời điểm quy tắc đã được thay đổi. Khi một quy tắc mới được tạo, nó sẽ được gán là rev: 1; để chỉ ra rằng đó là bản sửa đổi đầu tiên của quy tắc. Thay vì tạo sid mới mỗi khi thay đổi quy tắc, bạn nên giữ nguyên sid đó và tăng số sửa đổi. Trong trường hợp Snort hoặc Suricata gặp phải sid trùng lặp, họ sẽ sử dụng quy tắc có số sửa đổi cao hơn.

Format:

|  |
| --- |
| classtype:<class name>; |

#### 3.2.6.6. classtype

Từ khóa classtype cung cấp thông tin về việc phân loại các quy tắc và cảnh báo. Nó bao gồm tên ngắn, tên dài và mức độ ưu tiên. Ví dụ, nó có thể cho biết liệu 1 quy tắc chỉ là thông tin bình thường hay là 1 vụ hack.

Các loại tấn công được liệt kê trong bảng 3.6. Với mỗi loại classtype sẽ ứng với các mức độ nghiêm trọng . Hiện nay có 4 mức độ ưu tiên mặc định. 1 (high) là nghiêm trọng nhất và 4 (very low) là ít nghiêm trọng nhất.

*Bảng 3.6. Các loại tấn công và mức độ nghiêm trọng*

|  |  |  |
| --- | --- | --- |
| **Classtype** | **Description** | **Priority** |
| attempted-admin | Attempted Administrator Privilege Gain | high |
| attempted-user | Attempted User Privilege Gain | high |
| inappropriate-content | Inappropriate Content was Detected | high |
| policy-violation | Potential Corporate Privacy Violation | high |
| shellcode-detect | Executable code was detected | high |
| successful-admin | Successful Administrator Privilege Gain | high |
| successful-user | Successful User Privilege Gain | high |
| trojan-activity | A Network Trojan was detected | high |
| unsuccessful-user | Unsuccessful User Privilege Gain | high |
| web-application-attack | Web Application Attack | high |
| attempted-dos | Attempted Denial of Service | medium |
| attempted-recon | Attempted Information Leak | medium |
| bad-unknown | Potentially Bad Traffic | medium |
| default-login-attempt | Attempt to login by a default username and password | medium |
| denial-of-service | Detection of a Denial of Service Attack | medium |
| misc-attack | Misc Attack | medium |
| non-standard-protocol | Detection of a non-standard protocol or event | medium |
| rpc-portmap-decode | Decode of an RPC Query | medium |
| successful-dos | Denial of Service | medium |
| successful-recon-largescale | Large Scale Information Leak | medium |
| successful-recon-limited | Information Leak | medium |
| suspicious-filename-detect | A suspicious filename was detected | medium |
| suspicious-login | An attempted login using a suspicious username was detected | medium |
| system-call-detect | A system call was detected | medium |
| unusual-client-port-connection | A client was using an unusual port | medium |
| web-application-activity | Access to a potentially vulnerable web application | medium |
| icmp-event | Generic ICMP event | low |
| misc-activity | Misc activity | low |
| network-scan | Detection of a Network Scan | low |
| not-suspicious | Not Suspicious Traffic | low |
| protocol-command-decode | Generic Protocol Command Decode | low |
| string-detect | A suspicious string was detected | low |
| unknown | Unknown Traffic | low |
| tcp-connection | A TCP connection was detected | very low |

#### 3.2.6.7. priority

Từ khóa priority đi kèm với 1 giá trị số bắt buộc có thể nằm trong khoảng từ 1 đến 255. Các số từ 1 đến 4 thường được sử dụng nhất. Chữ ký có mức độ ưu tiên cao hơn sẽ được kiểm tra trước. Mức độ ưu tiên cao nhất là 1. Thông thường chữ ký đã được xác định độ ưu tiên thông qua classtype. Nếu trong 1 chữ ký có cả classtype và priority thì có thể nó sẽ bỏ qua classtype.

Format:

|  |
| --- |
| priority:<priority integer>; |

#### 3.2.6.8. metadata

Từ khóa metadata cho phép thêm thông tin phi chức năng vào chữ ký. Mặc dù định dạng là dạng tự do, nhưng bạn nên bám vào các cặp khóa, giá trị (key value) vì Suricata có thể đưa chúng vào các cảnh báo trước. Hãy xem bảng 3.7 để

*Bảng 3.7. Cặp khóa - giá trị trong metadata*

|  |  |  |
| --- | --- | --- |
| **Key** | **Description** | **Value Format** |
| engine | Indicate a Shared Library Rule | "shared" |
| soid | Shared Library Rule Generator and SID | gid | sid |
| service | Target-Based Service Identifier | "http" |

Format:

|  |
| --- |
| metadata:key1 value1;  metadata:key1 value1, key2 value2; |

#### 3.2.6.9. target

Từ khóa target cho phép người viết quy tắc chỉ định mặt nào của cảnh báo là mục tiêu của cuộc tấn công. Nếu được chỉ định, sự kiện cảnh báo được tăng cường để chứa thông tin về nguồn và đích.

Format:

|  |
| --- |
| target:[src\_ip|dest\_ip] |

Nếu giá trị là src\_ip thì IP nguồn trong sự kiện được tạo (trường src\_ip trong JSON) là mục tiêu của cuộc tấn công. Nếu target được đặt thành dest\_ip thì target là IP đích trong sự kiện được tạo.

# 

# **CHƯƠNG 4**

# **CÀI ĐẶT VÀ CẤU HÌNH SURICATA**

## **4.1. Giới thiệu ELK**

## 

ELK là từ viết tắt của ba dự án mã nguồn mở: Elasticsearch, Logstash và Kibana, phục vụ cho công việc logging. Các phần mềm có chức năng lần lượt như :

* Elasticsearch: Cơ sở dữ liệu để lưu trữ, tìm kiếm và query log.
* Logstash: Tiếp nhận log từ nhiều nguồn, sau đó xử lý log và ghi dữ liệu vào Elasticsearch
* Kibana: Giao diện để quản lý, thống kê log, cho phép người dùng trực quan hóa dữ liệu bằng bảng và đồ thị trong Elasticsearch.

Điểm mạnh của ELK là khả năng thu thập, hiển thị, truy vấn theo thời gian thực. Có thể đáp ứng truy vấn một lượng dữ liệu cực lớn.

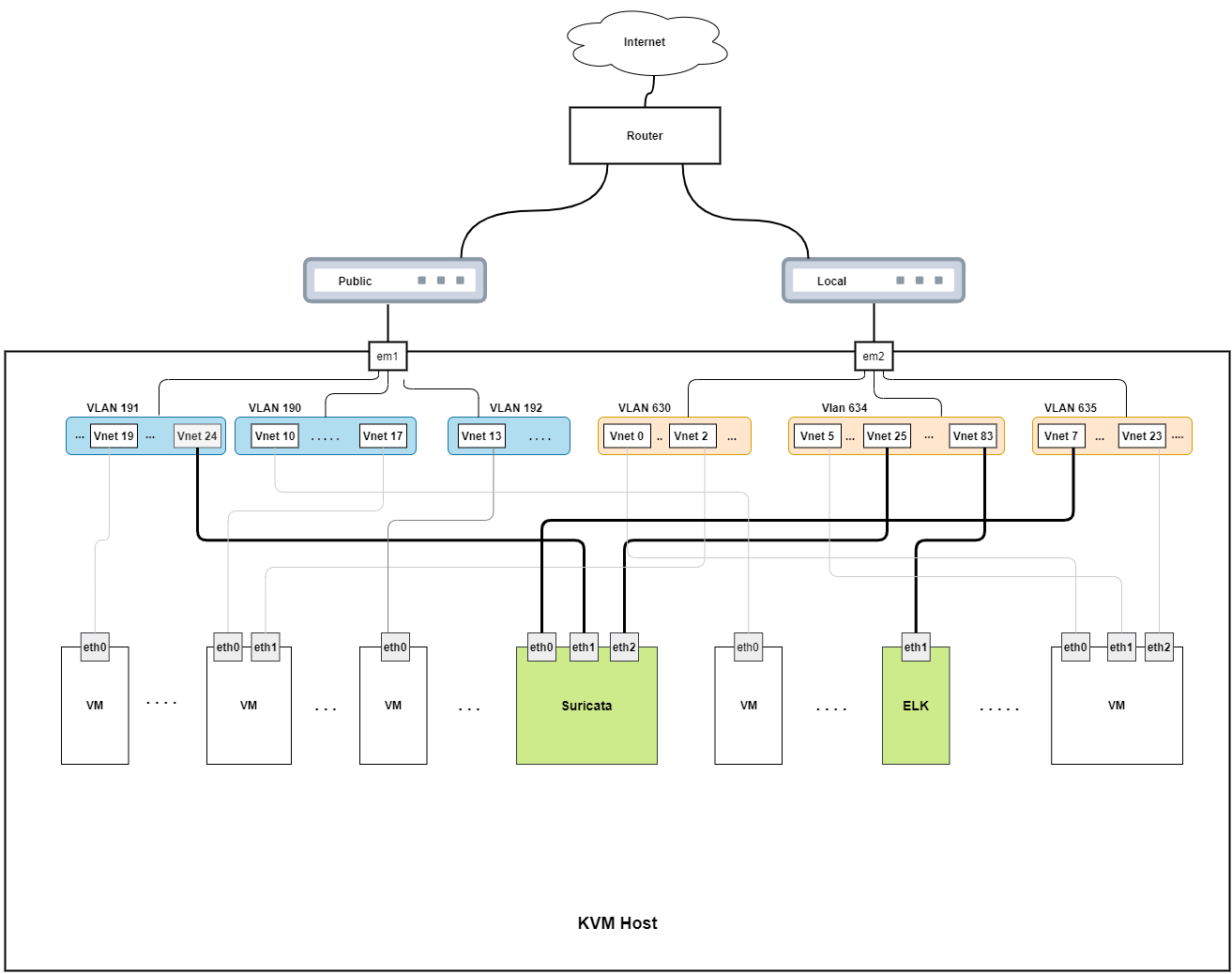
## **4.2. Mô hình triển khai**

### **4.2.1. Mô hình LAB**

KVM có 2 card mạng là em1 và em2, em1 là card mạng cung cấp cho các đường IP public, em2 là card mạng cung cấp đường IP private.

Tại máy ảo Suricata có card eth1, được sử dụng kỹ thuật mirror port để tất cả các traffic của các máy có IP public khi ra vào mạng sẽ đồng thời được đẩy về eth1.

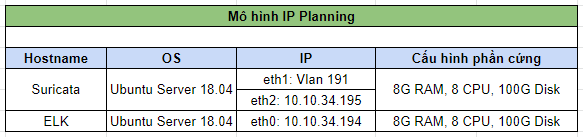
Các máy ảo Suricata và ELK lần lượt có các card mạng là eth2 và eth1 được kết nối tới cùng 1 Switch ảo có VLAN 634 để chúng có thể giao tiếp với nhau qua mạng local. Khi suricata bắt được các gói tin ra vào mạng, và đưa ra cảnh báo về những kết nối bất thường. Các cảnh báo đó được gửi về ELK để phân tích thông qua mạng local như hình 4.1.



*Hình 4.1. Mô hình triển khai Suricata trên KVM*

### **4.2.2. IP Planning**

Cấu hình phần cứng của các máy ảo Suricata và ELK phải đủ khỏe để có thể chịu tải tốt. Sơ đồ phân phối địa chỉ IP và cấu hình phần cứng của các máy ảo được thể hiện trong hình 4.2.



Hình 4.2. Mô hình IP planning cho các máy ảo

## **4.3. Cài đặt Suricata sử dụng ELK làm công cụ thu thập Log**

### **4.3.1. Cài đặt Suricata**

#### 4.3.1.1. Cài đặt

Suricata có thể được cài đặt theo 2 cách đó là cài đặt Suricata từ source hoặc cài đặt suricata từ binary packages. Tuy nhiên trong tài liệu này sẽ chỉ đề cập đến cách cài đặt từ source.

Update ubuntu và cài đặt các gói cần thiết:

|  |
| --- |
| apt update -y |

Để biên dịch suricata, trước tiên cần cài các thư viện và tiêu đề phát triển của chúng:

|  |
| --- |
| sudo apt -y install libpcre3 libpcre3-dbg libpcre3-dev build-essential autoconf \  automake libtool libpcap-dev libnet1-dev libyaml-0-2 libyaml-dev zlib1g zlib1g-dev \  libcap-ng-dev libcap-ng0 make libmagic-dev libjansson-dev libjansson4 pkg-config wget |

Cài đặt công cụ tự động cập nhật rule Suricata:

|  |
| --- |
| apt install python-pip -y |

|  |
| --- |
| pip install --upgrade suricata-update |

|  |
| --- |
| ln -s /usr/local/bin/suricata-update /usr/bin/suricata-update |

Suricata hoạt động như 1 IDS, để nó bao gồm cả các tính năng của IPS:

|  |
| --- |
| apt -y install libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink0 |

Tiếp theo ta sẽ tải về Suricata phiên bản 4.1 và giải nén tập tin:

|  |
| --- |
| cd  wget https://www.openinfosecfoundation.org/download/suricata-4.1.2.tar.gz  tar xzf suricata-4.1.2.tar.gz |

Điều hướng đến thư mục và biên dịch khả năng IPS:

|  |
| --- |
| cd suricata-4.1.2  ./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --localstatedir=/var |

Biên dịch và cài đặt Suricata 4.1:

|  |
| --- |
| make  make install-full |

Sau khi cài đặt, file cấu hình được đặt tại ***/etc/suricata/suricata.yaml***

#### 

#### 4.3.1.2. Cấu hình tự động cập nhật rule

Cấu hình tự động cập nhật rule:

|  |
| --- |
| apt install perl  wget http://nchc.dl.sourceforge.net/project/oinkmaster/oinkmaster/2.0/oinkmaster-2.0.tar.gz  tar zxvf oinkmaster-2.0.tar.gz  cd oinkmaster-2.0  cp oinkmaster.pl /usr/local/bin/  chmod +x /usr/local/bin/oinkmaster.pl  cp oinkmaster.conf /etc/suricata/ |

|  |
| --- |
| echo "url = http://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz">> /etc/suricata/oinkmaster.conf |

|  |
| --- |
| mkdir /etc/suricata/rules  oinkmaster.pl -C /etc/suricata/oinkmaster.conf -o /etc/suricata/rules |

Kiểm tra các rule:

|  |
| --- |
| ls /etc/suricata/rules/\*.rules |

|  |
| --- |
| echo "0 0 \* \* sun root oinkmaster.pl -C /etc/suricata/oinkmaster.conf -o /etc/suricata/rules">> /etc/crontab |

#### 

#### 4.3.1.3. Cấu hình suricata

Truy cập **/etc/suricata/suricata.yaml** và chỉnh sửa 1 số thông số cấu hình:

* Khai báo **HOME\_NET**:

|  |
| --- |
| 15 HOME\_NET: "[103.104.105.0/24, 103.104.106..0/24, 103.104.107.0/24]"  16 #HOME\_NET: "[192.168.0.0/16]"  17 #HOME\_NET: "[10.0.0.0/8]"  18 #HOME\_NET: "[172.16.0.0/12]"  19 #HOME\_NET: "any"  20  21 EXTERNAL\_NET: "!$HOME\_NET"  22 #EXTERNAL\_NET: "any" |

* Khai báo interface được mirror port để nhận các traffic đi qua Switch

|  |
| --- |
| 625 af-packet:  626 - interface: eth1  627 # Number of receive threads. "auto" uses the number of cores  628 #threads: auto  629 # Default clusterid. AF\_PACKET will load balance packets based on flow.  630 cluster-id: 99 |

* Khai báo đường dẫn tới thư mục chứa rule và khai báo những rule mà ta sẽ sử dụng cho hệ thống của mình.

|  |
| --- |
| 1851 default-rule-path: /etc/suricata/rules  1852 rule-files:  1853 - suricata.rules  1854 - emerging-malware.rules  1855 ##  1856 ## Advanced rule file configuration. |

Sau khi khai báo trong file **/etc/suricata/suricata.yaml,** tiếp theo sẽtạo file systemd để Suricata tự động khởi chạy:

|  |
| --- |
| cat <<EOF> /lib/systemd/system/suricata.service  [Unit]  Description=suricata NIDS Daemon  After=syslog.target network.target  [Service]  Type=simple  ExecStart=/usr/bin/suricata -c /etc/suricata/suricata.yaml -i eth1  [Install]  WantedBy=multi-user.target  EOF |

Khởi động Suricata:

|  |
| --- |
| systemctl daemon-reload  systemctl start suricata  systemctl enable suricata |

|  |
| --- |
| systemctl status suricata |

#### 4.3.1.4. Cấu hình filebeat đẩy log về ELK

|  |
| --- |
| apt install -y openjdk-8-jre apt-transport-https wget gnupg gnupg2 gnupg1 |

|  |
| --- |
| wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add - |

|  |
| --- |
| echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list  apt update -y |

|  |
| --- |
| apt install filebeat -y |

|  |
| --- |
| mv /etc/filebeat/filebeat.yml /etc/filebeat/filebeat.yml.bk |

|  |
| --- |
| cat <<EOF> /etc/filebeat/filebeat.yml  filebeat.inputs:  - type: log  enabled: true  paths:  - /var/log/suricata/eve.json  fields:  event.type: suricata  output.logstash:  hosts: ["10.10.34.194:5044"]  #ssl.certificate\_authorities: ["/etc/pki/root/ca.pem"]#ssl.certificate: "/etc/pki/client/cert.pem"#ssl.key: "/etc/pki/client/cert.key"  EOF |

|  |
| --- |
| systemctl enable filebeat  systemctl start filebeat  systemctl status filebeat |

## **4.4. Thử nghiệm các kiểu tấn công**

### **4.4.1. Ping of Death**

Một gói tin ICMP thông thường có kích thước chỉ khoảng từ 56 đến 84 byte. Tuy nhiên, bất kỳ gói IPV4 nào (bao gồm cả ping) có thể lớn đến 65,535 byte. Một số hệ thống máy tính không bao giờ được thiết kế để xử lý đúng một gói ping lớn hơn kích thước gói tối đa vì nó vi phạm [Giao thức Internet được](https://en.wikipedia.org/wiki/Internet_Protocol) ghi trong [RFC 791](https://tools.ietf.org/html/rfc791).[[1]](https://en.wikipedia.org/wiki/Ping_of_death#cite_note-ERICK2008-1) Lợi dụng điều này, attacker gửi các gói tin ICMP có kích thước lớn tới máy chủ nhằm chiếm đường truyền và ngăn chặn việc cung cấp dịch vụ của máy chủ.

Rule sau sẽ kiểm tra kích thước gói tin ICMP được gửi tới, nếu gói tin có kích thước lớn hơn bình thường sẽ đưa ra cảnh báo.

|  |
| --- |
| alert icmp any any -> $HOME\_NET any (msg:"ICMP Packet Large Size"; dsize: >200; sid: 3000001;) |

Thử ping tới hệ thống với gói tin lớn hơn 200 byte, sau khi kiểm tra log sẽ thu được cảnh báo như sau:

|  |
| --- |
| 10/02/2020-13:08:46.766876 [\*\*] [1:3000001:0] ICMP Packet Large Size [\*\*] [Classification: (null)] [Priority: 3] {ICMP} 103.101.160.130:8 -> 103.101.161.33:0  10/02/2020-13:08:46.766902 [\*\*] [1:3000001:0] ICMP Packet Large Size [\*\*] [Classification: (null)] [Priority: 3] {ICMP} 103.101.161.33:0 -> 103.101.160.130:8 |

### **4.4.2. Scan password**

Sử dụng file chứa rule trong bộ rule có sẵn đã tải về trước đó. Hãy enable file có tên là ***emerging-scan.rules*** vào file cấu hình của Suricata (***suricata.yaml***). Sau đó thực hiện tấn công Brute-force vào hệ thống. Sau khi tấn công, kiểm tra log sẽ có những cảnh báo với tiêu đề **SCAN** như sau:

|  |
| --- |
| 10/02/2020-13:18:25.859964 [\*\*] [1:2001972:20] ET SCAN Behavioral Unusually fast Terminal Server Traffic Potential Scan or Infection (Inbound) [\*\*] [Classification: Detection of a Network Scan] [Priority: 3] {TCP} 124.158.10.5:53142 -> 103.101.161.86:3389  10/02/2020-13:18:50.860896 [\*\*] [1:2001972:20] ET SCAN Behavioral Unusually fast Terminal Server Traffic Potential Scan or Infection (Inbound) [\*\*] [Classification: Detection of a Network Scan] [Priority: 3] {TCP} 42.112.31.178:62298 -> 103.101.161.86:3389 |

### **4.4.3. SQL Injection**

SQL injection là một lỗ hổng bảo mật web cho phép kẻ tấn công can thiệp vào các truy vấn mà ứng dụng thực hiện đối với cơ sở dữ liệu của nó. Nó thường cho phép kẻ tấn công xem dữ liệu mà chúng thường không thể lấy được. Điều này có thể bao gồm dữ liệu thuộc về người dùng khác hoặc bất kỳ dữ liệu nào khác mà bản thân ứng dụng có thể truy cập. Trong nhiều trường hợp, kẻ tấn công có thể sửa đổi hoặc xóa dữ liệu này, gây ra các thay đổi liên tục đối với nội dung hoặc hành vi của ứng dụng.

Một rule cảnh báo kẻ tấn công có thể cố gắng để lấy thông tin phiên bản SQL Server trên uri.

|  |
| --- |
| alert http $EXTERNAL\_NET any -> $HTTP\_SERVERS any (msg:"ET WEB\_SERVER Possible Attempt to Get SQL Server Version in URI using SELECT VERSION"; flow:established,to\_server; content:"SELECT"; nocase; http\_uri; content:"VERSION"; nocase; distance:1; http\_uri; reference:url,support.microsoft.com/kb/321185; reference:url,doc.emergingthreats.net/2011037; classtype:web-application-attack; sid:2011037; rev:5; metadata:created\_at 2010\_07\_30, updated\_at 2020\_08\_20;) |

Thử tấn công gửi các request chứa SELECT và VERSION trên uri của web.

|  |
| --- |
| echo "GET /form.php?q=1+UNION+SELECT+VERSION%28%29 HTTP/1.1\r\nHost: 127.0.0.1\r\n\r\n" | nc 103.101.161.33 80 |

Sau khi gửi request chứa các lệnh kiểm tra version trên uri của web. Liền có các cảnh báo của suricata trên hệ thống như sau:

|  |
| --- |
| 10/11/2020-22:56:33.840300 [\*\*] [1:2006446:13] ET WEB\_SERVER Possible SQL Injection Attempt UNION SELECT [\*\*] [Classification: Web Application Attack] [Priority: 1] {TCP} 58.186.116.167:34901 -> 103.101.161.33:80  10/11/2020-22:56:33.840300 [\*\*] [1:2011037:5] ET WEB\_SERVER Possible Attempt to Get SQL Server Version in URI using SELECT VERSION [\*\*] [Classification: Web Application Attack] [Priority: 1] {TCP} 58.186.116.167:34901 -> 103.101.161.33:80 |

# 

# 

# **CHƯƠNG 5**

# **MỘT SỐ GIẢI PHÁP HỌC MÁY CHO HỆ THỐNG SURICATA**

## **5.1. OPNids**

OPNids được phát triển bởi công ty CounterFlow AI và Deciso. OPNids được xây dựng dựa trên Suricata IDS, cung cấp một lớp trí thông minh dựa trên học máy để giúp cải thiện hoạt động ứng phó sự cố và săn tìm mối đe dọa.

### **5.1.1. Hoạt động của OPNids**

Cốt lõi của OPNids là Công cụ học máy DragonFly (MLE), sử dụng mô hình phân tích dữ liệu trực tuyến để nhập dữ liệu mạng tốc độ đường truyền từ Suricata. Theo Andrew Fast, trưởng nhóm khoa học dữ liệu tại CounterFlow AL: “ Hầu hết các kỹ thuật học máy (machine learning) là những gì truyền thống được gọi là kỹ thuật hàng loạt (batch techniques). Chúng tôi thực hiện cách tiếp cận khác, với phân tích trực tiếp, một nhánh mới hơn của học máy mà không được sử dụng rộng rãi”.

Với phân tích trực tuyến, thay vì xem xét một nhóm dữ liệu được thu thập, DragonFly MLE đang thu thập dữ liệu thống kê và đưa ra quyết định khi dữ liệu này chảy qua IDS.

OPNids cũng cho phép gửi các xử lý của dữ liệu để phân tích bổ sung, với cách tiếp cận là filebeat, thường được sử dụng cho ELastic Stack. Như vậy, dữ liệu OPNids có thể được chuyển tiếp đến công cụ tìm kiếm ELastic để phân tích bổ sung. Kafa cũng được sử dụng để xem dữ liệu trong elastic 1 cách trực quan nhất.

## **5.2.** [**Machine Learning Toolkit**](https://splunkbase.splunk.com/app/2890/) **sử dụng để phân tích Botnets**

Machine Learning Toolkit (MLTK) cho phép người dùng tạo, xác thực, quản lý và vận hành các mô hình học máy thông qua giao diện người dùng có hướng dẫn. MLTK không phải là một giải pháp mặc định mà là một cách để tạo học máy tùy chỉnh. Bạn phải có kiến ​​thức về miền, kiến ​​thức về Ngôn ngữ xử lý tìm kiếm Splunk (SPL), kinh nghiệm nền tảng Splunk và các kỹ năng hoặc kinh nghiệm về khoa học dữ liệu để sử dụng MLTK.

# 

### **5.2.1. Sử dụng MLTK để phân tích quần thể botnet**

MLTK phân tích dựa trên những lần đăng nhập không thành công của các địa chỉ IP cụ thể. Danh sách các địa chỉ IP bị nghi ngờ này có thể được phân tích cho các tính năng khác nhau như địa lý, đăng ký IANA, tần suất truy cập. v.v.

Kết hợp danh sách các mối đe dọa trên với dữ liệu netflow phát hiện xâm nhập thụ động sẽ tạo ra một tập dữ liệu phong phú để xây dựng một mô hình. Việc bổ sung theo ngữ cảnh và thông tin chi tiết về nỗ lực truy cập ở cấp độ gói cung cấp thông tin chi tiết về hoạt động được thực hiện bởi IP đó hoặc khối các IP trong 24h.

# 

## **5.3. Sử dụng lập trình**

Ngoài các giải pháp được hỗ trợ, có thể sử dụng lập trình để giảm thiểu các công việc phải ngồi theo dõi log trong suricata. Chẳng hạn như viết 1 bot để theo dõi log của các rule phát hiện cve. [[2](https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures)] Nếu có bất cứ cảnh báo nào trong các rule phát hiện các attacker đang khai thác lỗ hổng trong hệ thống thì cảnh báo về các nền tảng như mail, telegram, ...

# 

# **Kết luận**

# 

Các chương trong chuyên đề đã thể hiện các mục tiêu đặt ra khi thực hiện chuyên đề về cơ bản đều đã đạt được. Cụ thể như:

Chương 1 và chương 2 là hệ thống các khái niệm về IDS/IPS để nắm được cách hoạt động của suricata khi sử dụng làm IDS/IPS.

Chương 3 là phần tìm hiểu về hệ thống rule của suricata, nắm bắt được cấu trúc rule trong suricata để quản lý các rule có sẽ trong hệ thống cũng như khi cần thiết có thể tự viết các rule không có sẵn nhưng cần thiết cho hệ thống.

Sau khi nghiên cứu về Suricata và cấu trúc rule trong suricata, chương 4 là phần triển khai suricata làm IDS cho hệ thống mạng và thực hiện sử dụng rule với các cuộc tấn công nhỏ để kiểm tra sự hoạt động của suricata trong hệ thống.

Chương cuối cùng là chương 5, đây là chương đưa ra các giải pháp học máy có thể áp dụng vào hệ thống. Về cơ bản trên thế giới cung cấp rất nhiều các giải pháp học máy của các công ty lớn nhưng ngoài những giải pháp có sẵn, cũng có thể tự mình phát triển các giải pháp riêng cho hệ thống. Tuy các giải pháp phát triển riêng có thể không được bằng các giải pháp có sẵn, nhưng nó gần với nhu cầu thực tế của mỗi hệ thống.

Tuy rằng các vấn đề cơ bản đã được gia quyết nhưng vẫn có rất nhiều vấn đề khi triển khai trong suricata chưa được đề cập tới. Trước hết, chuyên đề này mới chỉ đề cập đến việc sử dụng, cài đặt và vận hành ban đầu của 1 hệ thống IDS, vẫn còn những vấn đề triển khai IPS hoặc các vấn đề phát sinh khi vận hành hệ thống vẫn chưa được đề cập tới. Việc vận hành trơn tru hệ thống IDS hoặc có thể sử dụng Suricata làm IPS là 1 điều khó khăn hơn rất nhiều. Nếu như 1 sai sót nhỏ trong hệ thống IPS có thể dẫn tới chặn nhầm các traffic vô hại mà bỏ qua cho các traffic độc hại. Việc giải quyết các vấn đề này cũng là mục tiêu nghiên cứu, phát triển tương lai của chuyên đề.

# 

# **Tài liệu tham khảo**

# 

[1]. Erickson, Jon (2008). [*HACKING the art of exploitation*](https://archive.org/details/hackingartexploi00eric) (2nd ed.). San

Francisco: NoStarch Press. p. [256](https://archive.org/details/hackingartexploi00eric/page/n269). [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [1-59327-144-1](https://en.wikipedia.org/wiki/Special:BookSources/1-59327-144-1).

[2]. ["CVE – Common Vulnerabilities and Exposures"](https://cve.mitre.org/). [Mitre Corporation](https://en.wikipedia.org/wiki/Mitre_Corporation).

2007-07-03. Retrieved 2009-06-18. CVE is sponsored by the National

Cyber Security Division of the U.S. Department of Homeland Security.

# 

# **PHỤ LỤC**

**Cài đặt ELK tích hợp với dự án sýnesis™ Lite for Suricata**

Cài đặt ELasticsearch

|  |
| --- |
| apt-get update -y |

|  |
| --- |
| apt install -y openjdk-8-jre apt-transport-https wget nginx git curl gnupg gnupg2 gnupg1 |

|  |
| --- |
| wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add - |

|  |
| --- |
| echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list |

|  |
| --- |
| apt update -y |

|  |
| --- |
| apt install elasticsearch -y |

|  |
| --- |
| sed -i 's/#network.host: 192.168.0.1/network.host: localhost/g' /etc/elasticsearch/elasticsearch.yml |

|  |
| --- |
| systemctl start elasticsearch  systemctl enable elasticsearch  systemctl status elasticsearch |

Sau khi cài đặt có thể kiểm tra bằng lệnh **curl -X GET** [**http://localhost:9200**](http://localhost:9200)

Cài đặt logstash và kibana

|  |
| --- |
| apt-get install -y logstash |

|  |
| --- |
| systemctl start logstash  systemctl enable logstash  systemctl status logstash |

|  |
| --- |
| apt-get install kibana |

|  |
| --- |
| sed -i 's/#server.host: "localhost"/server.host: "0.0.0.0"/'g /etc/kibana/kibana.yml |

|  |
| --- |
| systemctl start kibana  systemctl enable kibana  systemctl status kibana |

Cấu hình logstash nhận log từ Suricata

|  |
| --- |
| cd  git clone https://github.com/robcowart/synesis\_lite\_suricata.git |

|  |
| --- |
| mv /root/synesis\_lite\_suricata/logstash/synlite\_suricata /etc/logstash/ |

|  |
| --- |
| mv /root/synesis\_lite\_suricata/logstash/synlite\_suricata /etc/logstash/ |

|  |
| --- |
| cat <<EOF> /etc/logstash/pipelines.yml  - pipeline.id: synlite\_suricata  path.config: "/etc/logstash/synlite\_suricata/conf.d/\*.conf"  EOF |

|  |
| --- |
| sed -i 's/\-Xmx1g/\-Xmx4g/g' /etc/logstash/jvm.options  sed -i 's/\-Xms1g/\-Xms4g/g' /etc/logstash/jvm.options |

|  |
| --- |
| echo "indices.query.bool.max\_clause\_count: 8192">> /etc/elasticsearch/elasticsearch.yml  echo "search.max\_buckets: 250000">> /etc/elasticsearch/elasticsearch.yml |

Chỉnh sửa IP thành IP của ELK

|  |
| --- |
| vi /etc/logstash/synlite\_suricata/conf.d/30\_output\_elasticsearch.logstash.conf |

Di chuyển thư mục daemon

|  |
| --- |
| mv /root/synesis\_lite\_suricata/logstash.service.d /etc/systemd/system/ |

Chạy biến môi trường

|  |
| --- |
| cd /etc/logstash/ |

|  |
| --- |
| Environment="SYNLITE\_SURICATA\_RESOLVE\_IP2HOST=false"  Environment="SYNLITE\_SURICATA\_NAMESERVER=127.0.0.1"  Environment="SYNLITE\_SURICATA\_DNS\_HIT\_CACHE\_SIZE=25000"  Environment="SYNLITE\_SURICATA\_DNS\_HIT\_CACHE\_TTL=900"  Environment="SYNLITE\_SURICATA\_DNS\_FAILED\_CACHE\_SIZE=75000"  Environment="SYNLITE\_SURICATA\_DNS\_FAILED\_CACHE\_TTL=3600" |

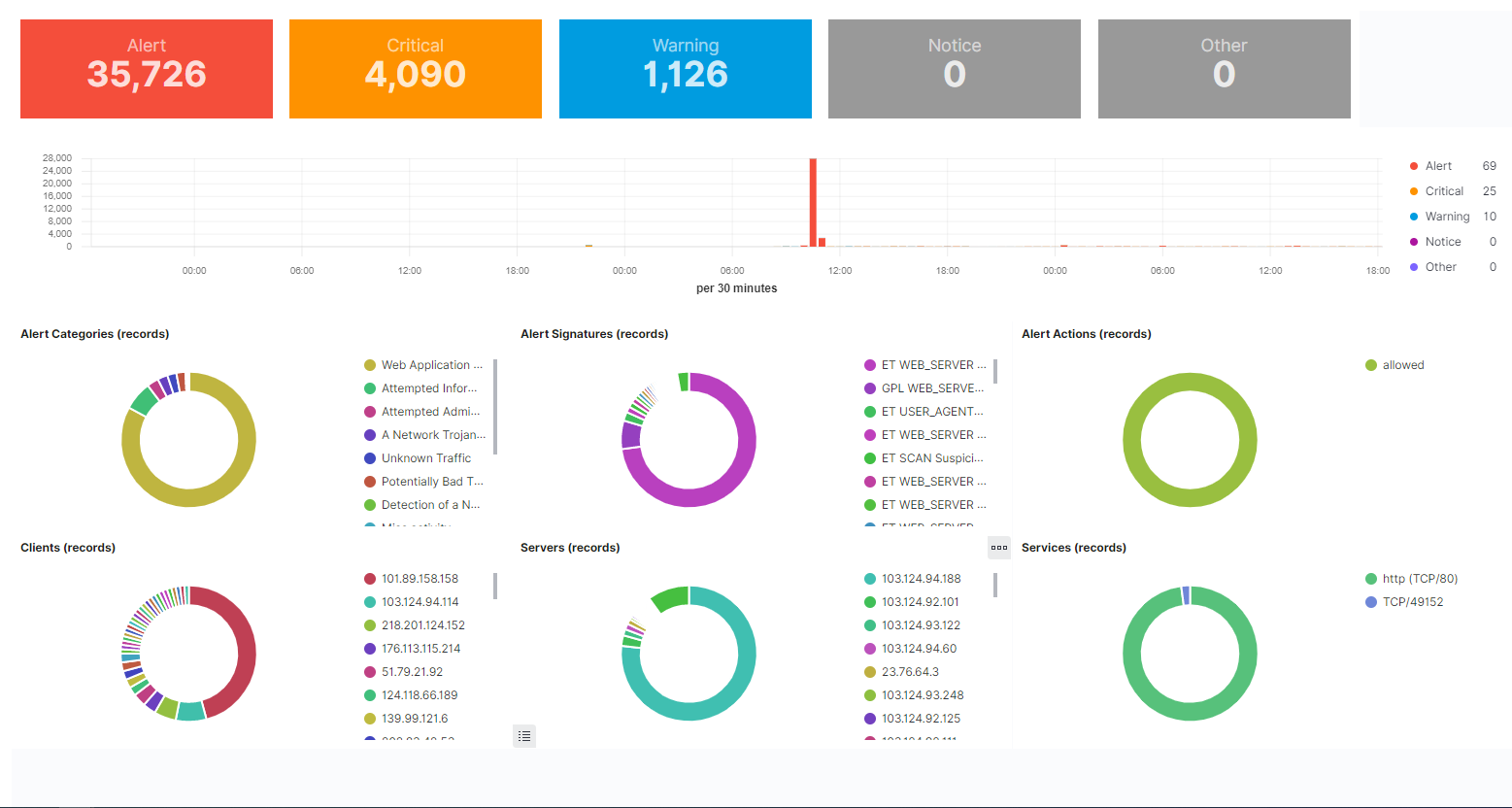
|  |
| --- |
| Environment="SYNLITE\_SURICATA\_DICT\_PATH=/etc/logstash/synlite\_suricata/dictionaries"  Environment="SYNLITE\_SURICATA\_TEMPLATE\_PATH=/etc/logstash/synlite\_suricata/templates"  Environment="SYNLITE\_SURICATA\_GEOIP\_DB\_PATH=/etc/logstash/synlite\_suricata/geoipdbs"  Environment="SYNLITE\_SURICATA\_GEOIP\_CACHE\_SIZE=8192"  Environment="SYNLITE\_SURICATA\_GEOIP\_LOOKUP=true"  Environment="SYNLITE\_SURICATA\_ASN\_LOOKUP=true"  Environment="SYNLITE\_SURICATA\_CLEANUP\_SIGS=false" |

Truy cập [***https://raw.githubusercontent.com/robcowart/synesis\_lite\_suricata/master/kibana/synlite\_suricata.kibana.7.1.x.json***](https://raw.githubusercontent.com/robcowart/synesis_lite_suricata/master/kibana/synlite_suricata.kibana.7.1.x.json) để tải về file json.

Sau đó truy cập kibana trên web để import file này vào kibana tại ***Stack Management*** -> ***Saved Objects***

Khởi động lại máy ảo ELK và truy cập Kibana để kiểm tra.

Khi log được đẩy lên ELK lập tức được thống kê dưới dạng biểu đồ dashboard tương tự như hình 8.1.



*Hình 8.1. Mẫu dữ liệu thống kê của Suricata khi đẩy lên ELK*