TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM **KHOA HỆ THỐNG THÔNG TIN VÀ VIẾN THÁM**

****

**BÁO CÁO ĐỒ ÁN MÔN BẢO MẬT MẠNG MÁY TÍNH VÀ HỆ THỐNG NGÀNH CÔNG NGHỆ THÔNG TIN**

**NGHIÊN CỨU VÀ TRIỂN KHAI HỆ THỐNG IDS/IPS SURICATA CÓ SỬ DỤNG MÁY HỌC ĐỂ PHÁT HIỆN VÀ PHÒNG CHỐNG CÁC HÌNH THỨC TẤN MẠNG.**

**Thành viên nhóm 15:**

**Đồng Văn Tuấn Đạt : 0850080015**

**Lâm Hoàng Nguyên Di : 0850080009**

**Võ Trương Phát Tài :0850080044**

**Lớp : 08\_ĐH\_CNPM**

**Khóa : 2019-2023**

**TP. Hồ Chí Minh, tháng 2 năm 2023**

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM **KHOA HỆ THỐNG THÔNG TIN VÀ VIẾN THÁM**

****

**BÁO CÁO ĐỒ ÁN MÔN BẢO MẬT MẠNG MÁY TÍNH VÀ HỆ THỐNG NGÀNH CÔNG NGHỆ THÔNG TIN**

**NGHIÊN CỨU VÀ TRIỂN KHAI HỆ THỐNG IDS/IPS SURICATA CÓ SỬ DỤNG MÁY HỌC ĐỂ PHÁT HIỆN VÀ PHÒNG CHỐNG CÁC HÌNH THỨC TẤN MẠNG.**

**Thành viên nhóm 15:**

**Đồng Văn Tuấn Đạt : 0850080015**

**Lâm Hoàng Nguyên Di : 0850080009**

**Võ Trương Phát Tài :0850080044**

**Lớp : 08\_ĐH\_CNPM**

**Khóa : 2019-2023**

**TP. Hồ Chí Minh, tháng 2 năm 2023**

**BẢNG PHÂN CÔNG CÔNG VIỆC**

**BẢNG PHÂN CÔNG CÔNG VIỆC VÀ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH**

Ghi chú: Ngoài công việc được phân công, mọi thành viên phải tham gia đọc và góp ý nội dung của các thành viên khác.

Tất cả công việc đều theo quyết định của nhóm trường Đồng Văn Tuấn Đạt, nếu có sai sót trong bài mong thầy thông cảm và bỏ qua.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Công việc |
| 1 | Đồng Văn Tuấn Đạt | 0850080015 | Phân tích thiết kế hệ thống, CSDL và một vài chức năng chính. |
| 2 | Lâm Hoàng Nguyên Di | 0850080009 | Thiết kế giao diện và các chức năng chính khác. |

**LỜI MỞ ĐẦU**

Việc bảo vệ hệ thống mạng luôn là một ưu tiên hàng đầu trong bất kỳ tổ chức hay doanh nghiệp nào. Trong thời đại công nghệ 4.0, việc sử dụng các phần mềm giám sát mạng để phát hiện và ngăn chặn các cuộc tấn công trở nên ngày càng phổ biến hơn. Một trong những phần mềm giám sát mạng được ưa chuộng hiện nay là Suricata - một hệ thống IDS/IPS mã nguồn mở, linh hoạt và hiệu quả.

Tuy nhiên, để có thể phát hiện và phòng chống các cuộc tấn công một cách hiệu quả, Suricata cần được trang bị thêm các tính năng thông minh sử dụng máy học. Điều này giúp cho Suricata có thể tự động học và phát hiện các hình thức tấn công mới một cách nhanh chóng và chính xác.

Trong bài nghiên cứu này, chúng tôi sẽ trình bày về cách triển khai hệ thống IDS/IPS Suricata sử dụng máy học để phát hiện và phòng chống các hình thức tấn công mạng. Chúng tôi sẽ tập trung vào việc áp dụng các kỹ thuật máy học như học sâu (deep learning), học tăng cường (reinforcement learning), và học không giám sát (unsupervised learning) để cải thiện độ chính xác và tốc độ phát hiện của hệ thống.

Bài nghiên cứu này mong muốn đóng góp cho việc nghiên cứu và phát triển các giải pháp bảo mật mạng mới, giúp cho các tổ chức và doanh nghiệp có thể bảo vệ hệ thống mạng của mình một cách an toàn và hiệu quả hơn.

**Chúng em xin chân thành cảm ơn !**

**Mục lục**

[**I. Giới thiệu đề tài.** 1](#_Toc131449394)

[**1.1. Lý do chọn đề tài 1**](#_Toc131449395)

[**1.2. Đối tượng 1**](#_Toc131449396)

[**1.3. Phạm vi 2**](#_Toc131449397)

[**1.4. Mục đích nghiên cứu 2**](#_Toc131449398)

[**1.5. Kỹ thuật công nghệ 2**](#_Toc131449399)

[**1.6. Cơ sở lý thuyết 2**](#_Toc131449400)

[1.6.1. Giới thiệu về Suricata 2](#_Toc131449401)

[1.6.2. Giới thiệu về VMware 4](#_Toc131449402)

[1.6.3. Tìm hiểu các dịch vụ IDS/IPS cứng và mềm. 5](#_Toc131449403)

[1.6.4. Tìm hiểu về các tập luật phát hiện xâm nhập. 6](#_Toc131449404)

[1.6.5. Tìm hiểu về máy học trong việc phát hiện xâm nhập. 6](#_Toc131449405)

[1.6.6. Tìm hiểu các hình thức tấn công mạng. 7](#_Toc131449406)

[**1.7. Luật trong Suricata 8**](#_Toc131449407)

[1.7.1. Rules format 8](#_Toc131449408)

[1.7.2. Rule Header 9](#_Toc131449409)

[1.7.3. TCP 17](#_Toc131449410)

[1.7.4. ICMP 18](#_Toc131449411)

[1.7.4. HTTP 19](#_Toc131449412)

[1.7.5. FLOW 22](#_Toc131449413)

[**1.8. Cấu trúc đồ án. 23**](#_Toc131449414)

[**II. Phân tích đề tài** 24](#_Toc131449415)

[**2.1. Phân tích yêu cầu hệ thống 24**](#_Toc131449416)

[**2.2. Yêu cầu chức năng 24**](#_Toc131449417)

[**2.3. Yêu cầu phi chức năng 25**](#_Toc131449418)

[2.3.1. Độ tin cậy 25](#_Toc131449419)

[2.3.2. Khả năng sử dụng 25](#_Toc131449420)

[2.3.4. Bảo mật 25](#_Toc131449421)

[2.3.5. Khả năng hỗ trợ 25](#_Toc131449422)

[2.3.6. Yêu cầu cấp phép 25](#_Toc131449423)

[2.3.7. Pháp lý, bản quyền và các thông báo khác 25](#_Toc131449424)

[2.3.8. Tiêu chuẩn áp dụng 26](#_Toc131449425)

[**2.4. Các công việc cần giải quyết 26**](#_Toc131449426)

[**III. Thiết kế** 27](#_Toc131449427)

[**3.1.** **Mô hình hoạt động của Suricata** 27](#_Toc131449428)

[**3.2.** **Các thành phần chính của hệ thống bao gồm:** 28](#_Toc131449429)

[**3.3. Kiến trúc Của Suricata 28**](#_Toc131449430)

[*1.3.1. Module giải mã gói dữ liệu* 29](#_Toc131449431)

[*1.3.2. Module tiền xử lý* 29](#_Toc131449432)

[*1.3.3. Module phát hiện* 30](#_Toc131449433)

[1.3.4. Module bản ghi và cảnh báo 31](#_Toc131449434)

[1.3.5. Module kết xuất thông tin 31](#_Toc131449435)

[**3.4. Suricata IPS 32**](#_Toc131449436)

[**IV. Thực hiện** 38](#_Toc131449437)

[**4.1. Cách cài đặt Suricata 38**](#_Toc131449438)

[4.1.1. Điều kiện cần để cài đặt Suricata 38](#_Toc131449439)

[4.1.2. Các bước để cài đặt Suricata 38](#_Toc131449440)

[**4.2. Các bước để cấu hình Suricata 41**](#_Toc131449441)

[**4.3. Cập nhật Suricata Rulesets 43**](#_Toc131449442)

[**4.4. Thử nghiệm Suricata Rules 45**](#_Toc131449443)

[**V. Kết luận và hướng phát triển** 47](#_Toc131449444)

[**5.1. Ưu điểm 47**](#_Toc131449445)

[**5.2. Nhược điểm 47**](#_Toc131449446)

[**5.3. Hướng phát triển 47**](#_Toc131449447)

[**CÁC TÀI LIỆU THAM KHẢO** 48](#_Toc131449448)

**DANH MỤC SƠ ĐỒ, HÌNH**

[Hình 1**:** Cấu trúc luật trong Suricata 9](file:///D:\Học%20kì%20II%20năm%204\bảo%20mật%20máy%20tinh%20thầy%20huynh\đồ%20án\mẫu%20đồ%20án%20Nhóm%2015.docx#_Toc131449806)

[Hình 2. *Có một máy khách có địa chỉ IP 1.2.3.4 và cổng 1024 và máy chủ có địa chỉ IP 5.6.7.8, nghe trên cổng 80 (thường là HTTP)* 11](#_Toc131449807)

[Hình 3.Bảng các tuỳ chọn của Reference 13](#_Toc131449808)

[Hình 4. Một số tuỳ chọn của Ipopts 15](#_Toc131449809)

[Hình 5 Các thành phần suricata 29](#_Toc131449810)

# **I. Giới thiệu đề tài.**

## 1.1. Lý do chọn đề tài

Tình trạng tấn công mạng ngày càng trở nên phổ biến: Trong thời đại công nghệ thông tin hiện nay, mối đe dọa từ các cuộc tấn công mạng là rất lớn và ngày càng phổ biến. Việc phát triển hệ thống bảo mật mạng là một ưu tiên hàng đầu để đảm bảo an toàn cho các tổ chức, doanh nghiệp và người dùng cá nhân.

Suricata là một hệ thống IDS/IPS mạnh mẽ: Suricata là một hệ thống IDS/IPS mã nguồn mở, có khả năng phát hiện và ngăn chặn các cuộc tấn công mạng. Nó được sử dụng rộng rãi trên các hệ thống mạng để cung cấp tính năng bảo mật mạnh mẽ.

Các kỹ thuật máy học đang được ứng dụng rộng rãi trong bảo mật mạng: Các kỹ thuật máy học như học sâu, học tăng cường và học không giám sát đang được ứng dụng rộng rãi trong các giải pháp bảo mật mạng. Sử dụng các kỹ thuật này để tăng cường khả năng phát hiện và phòng chống các cuộc tấn công mạng sẽ là một lợi thế đáng kể cho hệ thống IDS/IPS Suricata.

Các nghiên cứu và triển khai trước đó chưa sử dụng đầy đủ các kỹ thuật máy học: Mặc dù có nhiều nghiên cứu và triển khai về hệ thống IDS/IPS, tuy nhiên, rất ít các nghiên cứu và triển khai sử dụng đầy đủ các kỹ thuật máy học để cải thiện khả năng phát hiện và phòng chống tấn công mạng.

Với những lý do trên, chúng tôi quyết định chọn đề tài "Nghiên cứu và triển khai hệ thống IDS/IPS Suricata có sử dụng máy học để

## 1.2. Đối tượng

Đối tượng được nhắm đến trong đề tài "Nghiên cứu và triển khai hệ thống IDS/IPS Suricata có sử dụng máy học để phát hiện và phòng chống các hình thức tấn mạng" là các tổ chức, doanh nghiệp hoặc cá nhân có nhu cầu tăng cường bảo mật mạng và ngăn chặn các cuộc tấn công mạng đối với hệ thống mạng của mình. Nghiên cứu này sẽ tập trung vào việc triển khai hệ thống IDS/IPS Suricata với sự hỗ trợ của các kỹ thuật máy học để cải thiện khả năng phát hiện và phòng chống các hình thức tấn công mạng. Do đó, đối tượng được nhắm đến là các chuyên gia bảo mật, các nhà quản trị hệ thống, nhà phát triển phần mềm và các nhà nghiên cứu trong lĩnh vực bảo mật mạng.

## 1.3. Phạm vi

Phạm vi của đề tài "Nghiên cứu và triển khai hệ thống IDS/IPS Suricata có sử dụng máy học để phát hiện và phòng chống các hình thức tấn mạng" bao gồm:

* + Tìm hiểu các dịch vụ IDS/IPS cứng và mềm.
  + Tìm hiểu về các tập luật phát hiện xâm nhập.
  + Tìm hiểu về máy học trong việc phát hiện xâm nhập.
  + Xây dựng các tình huống tấn công mạng và thực nghiệm các phương pháp để phòng thủ bằng IDS/IPS bằng tập luật và máy học.
  + Tìm hiểu và triển khai được các hình thức tấn công mạng.

Tóm lại, phạm vi của đề tài này tập trung vào việc áp dụng các kỹ thuật máy học vào hệ thống IDS/IPS Suricata để tăng cường khả năng phát hiện và phòng chống các cuộc tấn công mạng, đồng thời đánh giá hiệu quả của hệ thống sau khi được cải thiện.

## 1.4. Mục đích nghiên cứu

Mục đích chính của nghiên cứ này là nâng cao khả năng phát hiện và phòng chống tấn công mạng của hệ thống IDS/IPS Suricata bằng cách áp dụng các kỹ thuật máy học như học sâu, học tăng cường và học không giám sát, đóng góp vào việc nâng cao khả năng phát hiện và phòng chống các cuộc tấn công mạng, giúp các tổ chức, doanh nghiệp và cá nhân tăng cường bảo mật mạng và giảm thiểu thiệt hại do các cuộc tấn công mạng gây ra..

## 1.5. Kỹ thuật công nghệ

* Môi trường mô phỏng và kiểm thử: VMware (Ubuntu 20.04)
* Hệ thống phát hiện xâm nhập mạng IDS
* Hệ thống ngăn chặn xâm nhập mạng IPS
* Công nghệ giám sát an ninh mạng NSM

## 1.6. Cơ sở lý thuyết

### 1.6.1. Giới thiệu về Suricata

Suricata là một hệ thống phát hiện xâm nhập mạng (Intrusion Detection System - IDS) và phòng thủ xâm nhập mạng (Intrusion Prevention System - IPS) mã nguồn mở. Nó được phát triển bởi OISF (Open Information Security Foundation), một tổ chức phi lợi nhuận nhằm hỗ trợ việc phát triển các công cụ an ninh mạng mã nguồn mở.

Suricata có khả năng phát hiện và ngăn chặn các cuộc tấn công mạng thông qua việc kiểm tra gói tin mạng truyền tải qua mạng. Nó có thể kiểm tra các gói tin theo nhiều tiêu chuẩn, bao gồm cả các chuẩn của IPSec và IPv6. Ngoài ra, Suricata hỗ trợ nhiều ngôn ngữ lập trình, cho phép người dùng tùy chỉnh các luật để phát hiện các cuộc tấn công mạng.

Suricata có thể được triển khai trên nhiều nền tảng khác nhau, bao gồm cả Linux, FreeBSD và Windows. Nó có thể hoạt động như một hệ thống độc lập hoặc tích hợp với các hệ thống an ninh khác để cung cấp một giải pháp an ninh mạng toàn diện.

Suricata được sử dụng rộng rãi trong các hệ thống an ninh mạng để phát hiện và ngăn chặn các cuộc tấn công mạng. Nó cũng được sử dụng trong các dự án nghiên cứu và giảng dạy để nghiên cứu và phát triển các kỹ thuật mới liên quan đến an ninh mạng.

**Suricata** là một công cụ giám sát an ninh mạng (NSM) sử dụng tập hợp các chữ ký do cộng đồng tạo ra và do người dùng xác định (còn được gọi là các quy tắc) để kiểm tra và xử lý lưu lượng mạng. Suricata có thể tạo các sự kiện nhật ký, kích hoạt cảnh báo và giảm lưu lượng khi phát hiện các gói hoặc yêu cầu đáng ngờ đối với bất kỳ dịch vụ khác nhau đang chạy trên máy chủ.

Theo mặc định, Suricata hoạt động như một hệ thống phát hiện xâm nhập thụ động (IDS) để quét các lưu lượng đáng ngờ trên máy chủ hoặc mạng. Nó sẽ tạo ra và ghi lại các cảnh báo để điều tra thêm. Nó cũng có thể được định cấu hình như một hệ thống ngăn chặn xâm nhập đang hoạt động (IPS) để ghi nhật ký, cảnh báo và chặn hoàn toàn lưu lượng mạng phù hợp với các quy tắc cụ thể.

### 1.6.2. Giới thiệu về VMware



VMware là một công ty công nghệ đa quốc gia của Mỹ chuyên cung cấp các giải pháp ảo hóa máy chủ và hệ thống lưu trữ dữ liệu. Các sản phẩm của VMware bao gồm VMware Workstation, VMware Fusion, VMware ESXi, VMware vSphere và VMware vCenter Server.

VMware Workstation là một phần mềm ảo hóa máy tính cá nhân cho phép người dùng chạy nhiều hệ điều hành trên cùng một máy tính vật lý. VMware Fusion là phiên bản tương tự dành cho máy tính Mac. VMware ESXi là một hypervisor (máy ảo trên máy vật lý) được cài đặt trực tiếp trên máy chủ để quản lý các máy ảo. VMware vSphere là một giải pháp quản lý máy chủ ảo được xây dựng trên nền tảng ESXi và VMware vCenter Server cung cấp các công cụ quản lý cho vSphere.

VMware cho phép người dùng tạo ra các máy ảo (virtual machine) để chạy các ứng dụng và hệ điều hành khác nhau trên cùng một máy tính vật lý. Việc sử dụng máy ảo giúp tiết kiệm chi phí phần cứng, giảm tình trạng tiêu thụ điện năng và tăng tính linh hoạt khi triển khai các ứng dụng.

VMware được sử dụng rộng rãi trong các doanh nghiệp và tổ chức để triển khai các ứng dụng và hệ thống mạng. Nó cũng được sử dụng trong các dự án nghiên cứu và giảng dạy để cung cấp các môi trường máy tính ảo cho sinh viên và nhà nghiên cứu.

### 1.6.3. Tìm hiểu các dịch vụ IDS/IPS cứng và mềm.

* Dịch vụ IDS/IPS cứng:
* Các dịch vụ IPS/IDS cứng (hardware-based IPS/IDS) là các giải pháp bảo mật mạng được triển khai dưới dạng phần cứng. Các thiết bị IPS/IDS cứng thường được đặt trên nhiều điểm trong mạng để giám sát và phát hiện các hoạt động xâm nhập, tấn công và các lỗ hổng bảo mật khác. Các thiết bị IPS/IDS cứng này thường được tích hợp sẵn các tính năng bảo mật như giám sát lưu lượng mạng, phân tích gói tin, ngăn chặn tấn công và báo cáo các sự kiện bảo mật.
* Những tính năng chính của IPS/IDS cứng bao gồm:

1. Phát hiện xâm nhập: IPS/IDS cứng phát hiện các hành vi xâm nhập vào hệ thống, bao gồm các cuộc tấn công mạng.
2. Báo động: IPS/IDS cứng báo động ngay lập tức khi phát hiện xâm nhập.
3. Chặn: IPS/IDS cứng có khả năng ngăn chặn các cuộc tấn công bằng cách cấu hình và triển khai các chính sách an ninh.
4. Quản lý: IPS/IDS cứng cung cấp một giao diện quản lý để giám sát hoạt động và cấu hình thiết bị.

* Các loại IPS/IDS cứng phổ biến bao gồm Cisco IPS, Snort, Juniper IPS, Fortinet IPS, Check Point IPS, Palo Alto IPS.
* Dịch vụ IDS/IPS mềm:
* Các dịch vụ IPS/IDS mềm là các phần mềm cung cấp chức năng phát hiện và ngăn chặn các cuộc tấn công mạng. Chúng thường được cài đặt trên các máy tính hoặc máy chủ và thực hiện các chức năng bảo mật mạng bằng cách giám sát các gói tin mạng và phát hiện các hành vi không hợp lệ.
* Các tính năng chính của IPS/IDS mềm bao gồm:

1. Giám sát lưu lượng mạng: IPS/IDS theo dõi các gói dữ liệu trong mạng để phát hiện các hành vi đáng ngờ hoặc các tấn công mạng.
2. Phát hiện tấn công: IPS/IDS có khả năng phát hiện các hành vi xâm nhập hoặc tấn công từ bên ngoài, bao gồm các cuộc tấn công từ mạng, từ các thiết bị đầu cuối hoặc từ các ứng dụng.
3. Ngăn chặn tấn công: IPS/IDS có khả năng ngăn chặn các hành vi xâm nhập hoặc tấn công từ bên ngoài bằng cách chặn các gói tin độc hại hoặc ngăn chặn các kết nối mạng đáng ngờ.
4. Báo cáo và cảnh báo: IPS/IDS có khả năng tạo ra các báo cáo và cảnh báo về các cuộc tấn công hoặc các hành vi đáng ngờ trong mạng.

* Các sản phẩm IPS/IDS mềm phổ biến bao gồm Snort, Suricata, OSSEC, Bro và Security Onion.

### 1.6.4. Tìm hiểu về các tập luật phát hiện xâm nhập.

* Các tập luật phát hiện xâm nhập (Intrusion Detection Rule) là các quy tắc được sử dụng bởi các hệ thống phát hiện xâm nhập để xác định các hoạt động bất thường trên mạng. Những quy tắc này được xây dựng dựa trên các mẫu hoạt động xâm nhập đã được biết đến và được lưu trữ trong các cơ sở dữ liệu.
* Một số liên quan đến các tập luật phát hiện xâm nhập bao gồm:
* Các tập luật phát hiện xâm nhập được sử dụng để phát hiện các hành vi xâm nhập thông qua việc so khớp các mẫu với lưu lượng mạng hoặc các hành vi độc hại đã biết trước đó.
* Các tập luật phát hiện xâm nhập thường được viết bằng các ngôn ngữ đặc biệt như SNORT, Suricata, hay Bro.
* Mỗi tập luật thường bao gồm các phần như header, địa chỉ IP, cổng, nội dung, và các thông số liên quan.
* Các tập luật phát hiện xâm nhập được cập nhật thường xuyên để đảm bảo khả năng phát hiện các hành vi xâm nhập mới nhất.
* Tuy nhiên, việc sử dụng các tập luật phát hiện xâm nhập cần được cân nhắc kỹ lưỡng, vì chúng có thể gây ra các báo động giả và không thể phát hiện được tất cả các hành vi xâm nhập.

### 1.6.5. Tìm hiểu về máy học trong việc phát hiện xâm nhập.

* Máy học (Machine Learning) được sử dụng rộng rãi trong việc phát hiện xâm nhập (Intrusion Detection) để giải quyết vấn đề về số lượng lớn các sự cố và độ chính xác của các phương pháp phát hiện thông thường.
* Các phương pháp máy học phát hiện xâm nhập có thể được chia thành hai loại chính: phân loại dựa trên kết quả học tập giám sát (supervised learning) và phân loại dựa trên kết quả học tập không giám sát (unsupervised learning).
* Trong phân loại dựa trên kết quả học tập giám sát, một mô hình được xây dựng từ dữ liệu đào tạo được gắn nhãn (labeled training data) và sau đó được sử dụng để phân loại các tập tin hoặc kết nối mạng mới. Các thuật toán phổ biến cho phân loại dựa trên kết quả học tập giám sát bao gồm Naive Bayes, Decision Tree và Random Forest.
* Trong phân loại dựa trên kết quả học tập không giám sát, các kỹ thuật như phân cụm (clustering) và phân tích thành phần chính (principal component analysis) được sử dụng để phát hiện các hành vi xâm nhập bất thường mà không cần phải biết trước các khuôn mẫu xâm nhập.
* Các kỹ thuật máy học cũng được sử dụng để xây dựng các hệ thống phát hiện xâm nhập dựa trên hành vi (behavior-based intrusion detection systems), bằng cách đánh giá các hành vi mạng và máy tính trong thời gian thực để phát hiện các hoạt động xâm nhập.
* Tuy nhiên, các hệ thống phát hiện xâm nhập dựa trên máy học cũng đối mặt với các thách thức như độ tin cậy của dữ liệu đào tạo và khả năng xử lý các mô hình phức tạp trong thời gian thực.

### 1.6.6. Tìm hiểu các hình thức tấn công mạng.

Các hình thức tấn công mạng là những phương thức, kỹ thuật được sử dụng để xâm nhập, tấn công vào các hệ thống mạng, thiết bị mạng, ứng dụng mạng hoặc dữ liệu mạng với mục đích gây tổn hại, truy cập trái phép hoặc lấy cắp thông tin.

Các công cụ mô phỏng tấn công mạng như Metasploit Framework, Cobalt Strike, kali Linux để thực hiện các kịch bản tấn công mạng

Một số hình thức tấn công mạng phổ biến hiện nay:

* Phishing: Lừa đảo người dùng để thu thập thông tin cá nhân hoặc tài khoản đăng nhập.
* Malware: Phần mềm độc hại được cài đặt trên máy tính hoặc thiết bị của người dùng để đánh cắp thông tin hoặc kiểm soát máy tính.
* DDoS: Tấn công chối dịch bằng cách gửi lượng lớn yêu cầu truy cập đến một máy chủ hoặc trang web để làm cho nó bị quá tải và không thể phục vụ các yêu cầu từ người dùng hợp lệ.
* Man-in-the-middle (MITM): Tấn công giữa người, khi kẻ tấn công giả mạo kết nối để theo dõi, thay đổi hoặc lấy cắp thông tin.
* SQL injection: Tấn công vào hệ thống cơ sở dữ liệu bằng cách chèn mã độc vào các trường nhập liệu để lấy thông tin.
* Cross-site scripting (XSS): Tấn công vào trang web bằng cách chèn mã độc vào trường nhập liệu để tấn công người dùng khác truy cập trang web đó.
* Password cracking: Tấn công vào tài khoản người dùng bằng cách thử các mật khẩu khác nhau.
* Wi-Fi eavesdropping: Người tấn công sử dụng kết nối Wi-Fi để nghe lén hoặc theo dõi hoạt động của người dùng.

## 1.7. Luật trong Suricata

### 1.7.1. Rules format

Chữ ký đóng một vai trò rất quan trọng trong Suricata. Trong hầu hết các dịp mọi người đang sử dụng các quy tắc hiện có.

Tài liệu Rules Suricata này giải thích tất cả về chữ ký, làm thế nào để đọc, điều chỉnh và tạo ra chúng.

Một quy tắc/chữ ký bao gồm:

* Các Action , xác định những gì sẽ xảy ra khi các chữ ký.
* Các Header , xác định giao thức, địa chỉ IP, cổng và chỉ đạo của các quy tắc.
* Các rules option, xác định các chi tiết cụ thể của quy tắc.

Một ví dụ về quy tắc như sau:

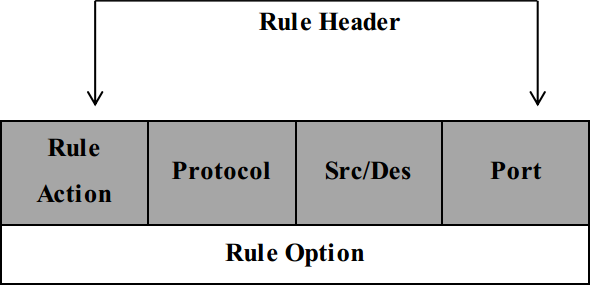
*drop tcp $HOME\_NET any -> $EXTERNAL\_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”; flow:established,to\_server;*

*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 này, Drop là Action, tcp là Header và phần trong ngoặc là rules option.

### 1.7.2. Rule Header



Hình **:** Cấu trúc luật trong Suricata

* Rule Action

Mục đầu tiên trong một luật đó chính là phần rule action, rule action sẽ nói cho Suricata biết phải làm gì khi thấy các gói tin phù hợp với các luật đã được quy định sẵn. Có 4 hành động mặc định trong Suricata đó là: pass (cho qua), drop (chặn gói tin), reject, alert (cảnh báo):

* + ***Pass:*** nếu signature được so sánh trùng khớp và chỉ ra là pass thì

Suricata sẽ thực hiện dừng quét gói tin và bỏ qua tất cả các luật phía sau đối với gói tin này.

* + ***Drop:*** nếu chương trình tìm thấy một signature hợp lệ và nó chỉ ra là

drop thì gói tin đó sẽ bị hủy bỏ và dừng truyền ngay lập tức, khi đó gói tin không thể đến được nơi nhận.

* + ***Reject:*** là hành động bỏ qua gói tin, bỏ qua ở cả bên nhận và bên gửi.

Suricata sẽ tạo ra một cảnh báo với gói tin này.

* + ***Alert:*** nếu signature được so sánh là hợp lệ và có chứa một alert thì gói tin đó sẽ được xử lý giống như với một gói tin không hợp lệ. Suricata sẽ tạo ra một cảnh báo.
* Protocol

Trường tiếp theo trong luật đó là protocol. Các giao thức mà Suricata hiện đang phân tích các hành vi bất thường đó là TLS, SSH, SMTP (tải thư điện tử qua mạng internet), IMAP (đặt sự kiểm soát email trên mail server), MSN, SMB (chia sẻ file), TCP, UDP, ICMP và IP, DNS, HTTP, HTTPS.

• **IP Address**

Mục tiếp theo của phần header đó là địa chỉ IP. Các địa chỉ này dùng để kiểm tra nơi đi và nơi đến của một gói tin. Địa chỉ ip đó có thể là địa chỉ của một máy đơn hoặc cũng có thể là địa chỉ của một lớp mạng. Từ khóa “any” được sử dụng để định nghĩa một địa chỉ bất kỳ.

Một địa chỉ ip sẽ được viết dưới dạng *ip\_address/netmask*. Điều này có nghĩa là nếu netmask là /24 thì lớp mạng đó là lớp mạng C, /16 là lớp mạng B hoặc /32 là chỉmột máy đơn. Ví dụ: địa chỉ 192.168.1.0/24 có nghĩa là một dải máy có địa chỉ IP từ 192.168.1.1-192.168.1.255.

Trong hai địa chỉ IP trong một luật Suricata thì sẽ có một địa chỉ IP nguồn và một địa chỉ IP đích. Việc xác định đâu là địa chỉ nguồn, đâu là địa chỉ đích phụ thuộc vào “→”.

Ngoài ra toán tử phủ định có thể được áp dụng cho việc định địa chỉ IP. Có nghĩa là khi sử dụng toán tử này thì Suricata sẽ bỏ qua việc kiểm tra địa chỉ của gói tin đó. Toán tử đó là “!”. Ngoài ra ta có thể định nghĩa một danh sách các địa chỉ IP bằng cách viết liên tiếp chúng cách nhau bởi một dấu “,”.

**Ví dụ:**

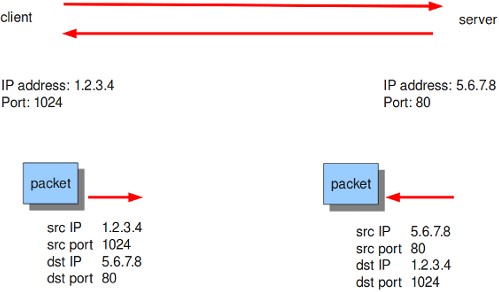
*Alert TCP any any → ![192.168.1.0/24, 172.16.0.0/16] 80 (msg: “Access”)*

**Port**

Port có thể được định nghĩa bằng nhiều cách. Với từ khóa “any” giống như địa chỉIP để chỉ có thể sử dụng bất kỳ port nào. Gán một port cố định, ví dụ như gán kiểm tra ở port 80 http hoặc port 22 ssh. Ngoài ra ta cũng có thể sử dụng toán tử phủ định để bỏ qua một port nào đó hoặc liệt kê một dải các port.

**Điều hướng**

Toán tử hướng “→” chỉ ra đâu là hướng nguồn, đâu là hướng đích. Phần địa chỉ IP và port ở phía bên trái của toán tử được coi như là địa chỉ nguồn và port nguồn, phần bên phải được coi như địa chỉ đích và port đích. Ngoài ra còn có toán tử “<>” Suricata sẽ xem cặp địa chỉ/port nguồn và đích là như nhau. Nghĩa là nó sẽ ghi/phân tích ở cả hai phía của cuộc hội thoại.



Hình . *Có một máy khách có địa chỉ IP 1.2.3.4 và cổng 1024 và máy chủ có địa chỉ IP 5.6.7.8, nghe trên cổng 80 (thường là HTTP)*

**Ví dụ:**

*Alert tcp 1.2.3.4 1024 -> 5.6.7.8 80*

* + - **Rule Option**

Rule Options chính là trung tâm của việc phát hiện xâm nhập. Nội dung chứa các dấu hiệu để xác định một cuộc xâm nhập. Nó nằm ngay sau phần Rule Header và được bọc bởi dấu ngoặc đơn “()”. Tất cả các rule options sẽ được phân cách nhau bởi dấu chấm phẩy “;”, phần đối số sẽ được tách ra bởi dấy hai chấm “:”.

Có 4 loại rule options chính bao gồm:

* + - * ***General:*** Tùy chọn này cung cấp thông tin về luật đó nhưng không có bất cứảnh hưởng nào trong quá trình phát hiện.
      * ***Payload:***Tùy chọn liên quan đến phần tải trong một gói tin.
      * ***Non-payload:***Bao gồm các tùy chọn không liên quan đến phần tải của gói tin (header).
      * ***Post-detection:*** Các tùy chọn này sẽ gây ra những quy tắc cụ thể sau khi một luật đã được kích hoạt.

***Các thành phần khác trong Rule:***

* General

Msg

Msg (Message): được dùng để cho biết thêm thông tin về từng signature và các cảnh báo. Phần đầu tiên sẽ cho biết tên tập tin của signature và phần này quy ước là phải viết bằng chữ in hoa. Định dạng của msg như sau:

*msg: “. ”;*

Sid

Sid (signature id): cho ta biết định danh riêng của mỗi signature. Định danh này được bắt đầu với số. Định dạng của sid như sau:

*sid:123;*

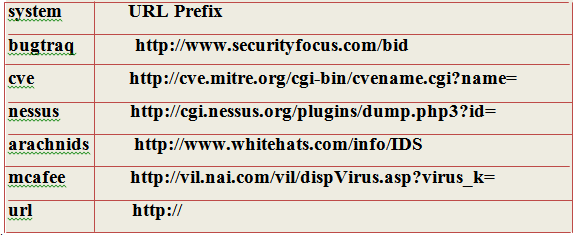
Rev

Rev (revision): mỗi sid thường đi kèm với một rev. Rev đại diện cho các phiên bản của signature. Mỗi khi signature được sửa đổi thì số rev sẽ được tăng lên bởi người tạo ra. Định dạng của rev như sau:

*rev:123;*

Reference

Reference: cung cấp cho ta địa chỉ đến được những nơi chứa các thông tin đầy đủ về signature. Các tham chiếu có thể xuất hiện nhiều lần trong một signature. Ví dụ về một tham chiếu như sau:



*reference: url,* [*www.info.nl*](http://www.info.nl/)

Hình .Bảng các tuỳ chọn của Reference

Classtype

Classtype: cung cấp thông tin về việc phân loại các lớp quy tắc và cảnh báo.

Mỗi lớp bao gồm một tên ngắn gọn, một tên đầy đủ và mức độ ưu tiên.

Priority

Priority: chỉ ra mức độ ưu tiên của mỗi signature. Các giá trị ưu tiên dao động từ 1 đến 255, nhưng thường sử dụng các giá trị từ 1 -> 4. Mức ưu tiên cao nhất là

Những signature có mức ưu tiên cao hơn sẽ được kiểm tra trước. Định dạng như sau:

*priority:1;*

Metadata

Metadata: Suricata sẽ bỏ qua những gì viết sau metadata. Định dạng như sau:

*metadata: ;*

Content

Content: thể hiện nội dung chúng ta cần viết trong signature, nội dung này được đặt giữa 2 dấu nháy kép. Nội dung là các byte dữ liệu, có 256 giá trị khác nhau (0- 255). Chúng có thể là các ký tự thường, ký tự hoa, các ký tự đặc biệt, hay là các mã hexa tương ứng với các ký tự và các mã hexa này phải được đặt giữa 2 dấu gạch dọc. Định dạng của một nội dung như sau:

*content: ”. ”;*

Nocase

Nocase: được dùng để chỉnh sửa nội dung thành các chữ thường, không tạo rasự khác biệt giữa chữ hoa và chữ thường. Nocase cần được đặt sau nội dung cần chỉnh sửa.

Within

Within: được dùng cùng với distance, để chỉ độ rộng của các byte cần kiểm tra sau một nội dung với khoảng cách cho trước đó.

Dsize

Dsize: được dùng để tìm một payload có độ dài bất kỳ.

*dsize:min<>max;*

Rpc

Rpc (Remote Procedure Call): là một ứng dụng cho phép một chương trình máy tính thực hiện một thủ tục nào đó trên một máy tính khác, thường được sử dụng cho quá trình liên lạc. Định dạng của rpc như sau:

*rpc:<application number>, [<version number>|\*], [<procedure number>|*

*\*]>;*

Replace

Replace được dùng để thay đổi nội dung của payload, điều chỉnh lưu lượng mạng. Việc sửa đổi nội dung của payload chỉ có thể được thực hiện đối với gói dữ liệu cá nhân. Sau khi thực hiện thay đổi nội dung xong thì Suricata sẽ thực hiện tính toán lại trường checksum.

* **Non-Payload**
* **IP**

**TTL**

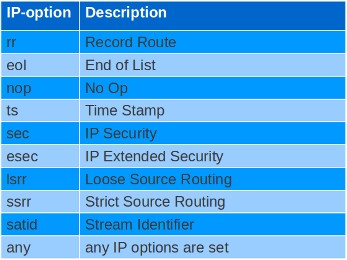
Được sử dụng để kiểm tra về thời gian sống, tồn tại tên mạng của một địa chỉ IP cụ thể trong phần đầu của mỗi gói tin. Giá trị time-to-live (thời gian sống), xác định thời gian tối đa mà mỗi gói tin có thể được lưu thông trên hệ thống mạng. Nếu giá trị này về 0 thì gói tin sẽ bị hủy bỏ. Thời gian sống được xác định dựa trên số

hop, khi đi qua mỗi hop/router thì thời gian sống sẽ bị trừ đi 1. Cơ chế này nhằm hạn chế việc gói tin lưu thông trên mạng vô thời hạn. Định dạng của một ttl như sau:

*ttl:<number>;*

ipopts

Chúng ta có thể xem và tùy chỉnh các tùy chọn cho việc thiết lập các địa chỉ IP. Việc thiết lập các tùy chọn cần được thực hiện khi bắt đầu một quy tắc. Một số tùy chọn có thể sử dụng:



Hình . Một số tuỳ chọn của Ipopts

Định dạng của một ipopts như sau:

*ipopts: <name>;*

**sameip**

Mỗi gói tin sẽ có một địa chỉ IP nguồn và đích. Chúng ta có thể sử dụng sameip để kiểm tra xem địa chỉ IP nguồn và đích có trùng nhau hay không. Định dạng của sameip như sau:

*sameip;*

Ip\_proto

Được dùng để giúp ta lựa chọn giao thức. Ta có thể chọn theo tên hoặc số tương ứng với từng giao thức. Có một số giao thức phổ biến sau:

*1*

*6*

*17*

*47*

*50*

*51*

*58*

*ICMP TCP UDP GRE ESP*

*AH*

*Internet Control Message Transmission Control Protocol*

*User Datagram*

*General Routing Encapsulation Encap Security Payload for IPv6*

*Authentication Header for Ipv6*

*IPv6-ICMP ICMP for Ipv6*

Định dạng của ip\_proto như sau:

*ip\_proto:<number/name>;*

Id

Được sử dụng để định danh cho các phân mảnh của gói tin được truyền đi. Khi gói tin truyền đi sẽ được phân mảnh, và các mảnh của một gói tin sẽ có ID giống nhau. Việc này giúp ích cho việc ghép lại gói tin một cách dễ dàng. Định dạng như sau:

*id:<number>;*

Geoip

Cho phép xác định địa chỉ nguồn, đích để gói tin lưu thông trên mạng.

**Fragbits**

Được dùng để kiểm tra các phân mảnh của gói tin. Nó bao gồm các cơ chế sau:

*M - More Fragments*

*D - Do not Fragment R - Reserved Bit*

*+*

*\**

*!*

*match on the specified bits, plus any others match if any of the specified bits are set*

*match if the specified bits are not set*

Định dạng của một Fragbits như sau:

*fragbits:[\*+!]<[MDR]>;*

Fragoffset

Kiểm tra sự phù hợp trên các giá trị thập phân của từng mảnh gói tin trên trường offset. Nếu muốn kiểm tra phân mảnh đầu tiên của gói tin, chúng ta cần kết hợp fragoffset 0 với các tùy chọn fragment khác. Các tùy chọn fragment như sau:

*< match if the value is smaller than the specified value*

*> match if the value is greater than the specified value*

*! match if the specified value is not present*

Định dạng của fragoffset:

*fragoffset:[!|<|>]<number>;*

### 1.7.3. TCP

* Sed

Là một số ngẫu nhiên được tạo ra ở cả bên nhận và bên gửi gói tin để kiểm tra số thứ tự của các gói tin đến và đi. Máy khách và máy chủ sẽ tự tạo ra một số seq riêng của mình. Khi một gói tin được truyền thì số seq này sẽ tăng lên 1. Seq giúp chúng ta theo dõi được những gì diễn ra khi một dòng dữ liệu được truyền đi.

* Ack

Được sử dụng để kiểm tra xem gói tin đã được nhận bởi nơi nhận hay chưa trong giao thức kết nối TCP. Số thứ tự của ACK sẽ tăng lên tương ứng với số byte dữ liệu đã được nhận thành công.

* Window

Được sử dụng để kiểm tra kích thước của cửa sổ TCP. Kích thước cửa sổ TCP là một cơ chế dùng để kiểm soát các dòng dữ liệu. Cửa sổ được thiết lập bởi người nhận, nó chỉ ra số lượng byte có thể nhận để tránh tình trạng bên nhận bị tràn dữ liệu. Giá trị kích thước của cửa sổ có thể chạy từ 2 đến 65.535 byte.

### 1.7.4. ICMP

* Itype

Cung cấp cho việc xác định các loại ICMP. Các thông điệp khác nhau sẽ được phân biệt bởi các tên khác nhau hay các giá trị khác nhau.

Định dạng của itype như sau:

*itype:min<>max;*

*itype:[<|>]<number>;*

* Icode

Cho phép xác định mã của từng ICMP để làm rõ hơn cho từng gói tin ICMP. Định dạng của icode như sau:

*icode:min<>max;*

*icode:[<|>]<number>;*

* Icmp\_id

Mỗi gói tin ICMP có một giá trị ID khi chúng được gửi. Tại thời điểm đó, người nhận sẽ trả lại tin nhắn với cùng một giá trị ID để người gửi sẽ nhận ra và kết nối nó đúng với yêu cầu ICMP đã gửi trước đó. Định dạng của một icmp\_id như sau:

*icmp\_id:<number>;*

* Icmp\_seq

Được sử dụng để kiểm tra số thứ tự của ICMP. Định dạng của icmp\_seq như sau:

*icmp\_seq:<number>;*

### 1.7.4. HTTP

Có các sửa đổi nội dung bổ sung có thể cung cấp các khả năng dành riêng cho giao thức ở lớp ứng dụng. Thông tin thêm có thể được tìm thấy tại [Payload](https://suricata.readthedocs.io/en/suricata-5.0.0/rules/payload-keywords.html) [Keyword](https://suricata.readthedocs.io/en/suricata-5.0.0/rules/payload-keywords.html)s Các từ khóa này đảm bảo chữ ký chỉ kiểm tra các phần cụ thể của lưu lượng mạng. Chẳng hạn, để kiểm tra cụ thể về URI yêu cầu, cookie hoặc cơ quan phản hồi hoặc yêu cầu HTTP, v.v.

Tất cả các từ khóa HTTP là sửa đổi. Lưu ý sự khác biệt giữa sửa đổi nội dung và bộ đệm dính. Xem [từ khóa sửa đổ](https://suricata.readthedocs.io/en/suricata-5.0.0/rules/intro.html#rules-modifiers)i để biết thêm thông tin.

**content modifiers** nhìn lại quy tắc, ví dụ:

alert http any any -> any any (content:"index.php"; http\_uri; sid:1;)

**sticky buffers** được đặt đầu tiên và tất cả các từ khóa theo sau nó áp dụng cho bộ đệm đó, ví dụ:

alert http any any -> any any (http\_response\_line; content:"403 Forbidden"; sid:1;)

* Các từ khóa yêu cầu:



* **Các từ khóa phản hồi**:



* http\_method: chỉ ra các phương thức được áp dụng với các request http. Các phương thức http: GET, POST, PUT, HEAD, DELETE, TRACE, OPTIONS, CONNECT và PATCH.
* http\_uri và http\_raw\_uri: chỉ ra đường dẫn tới nơi chứa nội dung yêu cầu.
* http\_header: chỉ ra phương thức sử dụng, địa chỉ cần truy cập tới và tình trạng kết nối.
* http\_cookie.
* http\_user\_agent: là một phần của http\_header, chỉ ra thông tin về trình duyệt của người dùng.
* http\_client\_body: chỉ ra các yêu cầu của máy trạm.
* http\_stat\_code: chỉ ra mã trạng thái của server mà máy trạm yêu cầu kết nối tới.
* http\_stat\_msg: các dòng tin thông báo về tình trạng máy chủ, hay tình trạng về việc đáp ứng các yêu cầu kết nối của máy trạm.
* http\_server\_body: chỉ ra nội dung đáp trả các yêu cầu từ máy trạm của máy chủ.
* File\_data: chỉ ra nội dung, đường dẫn tới file chứa dữ liệu được yêu cầu.

### 1.7.5. FLOW

* Flowbits

Gồm 2 phần, phần đầu mô tả các hành động được thực hiện, phần thứ 2 là tên của flowbit. Các hành động của flowbit:

*flowbits: set, name các flow.*

*flowbits: isset, name*

*Được dùng để thiết lập các điều kiện/tên cho*

*Có thể được sử dụng trong các luật để đảm*

*bảo rằng sẽ tạo ra một cảnh báo khi các luật là phù hợp và các điều kiện sẽ*

*được thiết lập trong flow.*

*flowbits: toggle, name flowbits: unset, name trong luật.*

*flowbits: isnotset, name*

*Dùng để đảo ngược các thiết lập hiện tại.*

*Được sử dụng để bỏ các thiết lập về điều kiện*

*Được sử dụng để đảm bảo rằng sẽ tạo ra*

*một cảnh báo khi các luật là phù hợp và các điều kiện sẽ không được thiết*

*lập trong flow.*

* Flow

Có thể được sử dụng để kết nối các thư mục chứa các flow lại với nhau. Các flow có thể được đi từ hoặc đến từ Client/Server và các flow này có thể ở trạng thái được thiết lập hoặc không. Việc kết nối các flow có thể xảy ra các trường hợp sau:

## 1.8. Cấu trúc đồ án.

* Chương I: Giới thiệu đề tài

Xác định mục tiêu nghiên cứu: Trong bước này, chúng ta cần xác định mục tiêu nghiên cứu, đó là tìm hiểu và triển khai hệ thống IDS/IPS sử dụng Suricata để phát hiện và phòng chống các hình thức tấn công mạng.

* Chương II: Phân tích đề tài

Phân tích yêu cầu chức năng và phi chức năng của hệ thống: Chúng ta cần phân tích và xác định các yêu cầu chức năng và phi chức năng của hệ thống, bao gồm các chức năng phát hiện tấn công, quản lý cảnh báo, thực thi chính sách bảo mật, ghi nhật ký và báo cáo.

* Chương III: Thiết kế

Thiết kế kiến trúc hệ thống: Sau khi phân tích yêu cầu chức năng và phi chức năng, chúng ta cần thiết kế kiến trúc hệ thống, bao gồm cấu hình Suricata, thiết lập các quy tắc và chính sách bảo mật, cài đặt các phần mềm và phần cứng cần thiết để triển khai hệ thống.

* Chương IV: Thực hiện

Tiến hành thử nghiệm và đánh giá: Sau khi triển khai hệ thống, chúng ta cần thử nghiệm và đánh giá hiệu suất của hệ thống. Quá trình này bao gồm việc phát hiện và phòng chống các cuộc tấn công mạng, kiểm tra tính chính xác của các cảnh báo, đánh giá hiệu suất và tối ưu hóa hệ thống.

* Chương V: Kết luện Và Hướng phát triển

`Tổng kết và trình bày kết quả: Cuối cùng, chúng ta cần tổng kết và trình bày kết quả của đề tài nghiên cứu, bao gồm các bài học rút ra được từ quá trình triển khai hệ thống, đánh giá hiệu suất và tính khả thi của hệ thống trong thực tế.

# **II. Phân tích đề tài**

## 2.1. Phân tích yêu cầu hệ thống

Để triển khai thành công hệ thống IDS/IPS Suricata có sử dụng máy học để phát hiện và phòng chống các hình thức tấn mạng, các yêu cầu cần được đảm bảo, bao gồm:

* Phần cứng: Hệ thống cần được cấu hình với phần cứng đủ mạnh để xử lý lưu lượng mạng lớn, đảm bảo hiệu suất cao của hệ thống. Phần cứng phải được nâng cấp để đáp ứng nhu cầu phát triển của hệ thống.
* Hệ điều hành: Hệ thống cần được cài đặt với hệ điều hành phù hợp để hỗ trợ các phần mềm và công nghệ cần thiết cho việc triển khai hệ thống IDS/IPS.
* Phần mềm: Hệ thống cần được cài đặt với các phần mềm cần thiết để triển khai hệ thống IDS/IPS Suricata, bao gồm Suricata, các thư viện và các công cụ phân tích mạng.
* Mạng: Hệ thống cần được triển khai trên mạng có sự đa dạng về các dòng gói tin, các loại giao thức và các địa chỉ IP khác nhau để đảm bảo khả năng phát hiện các cuộc tấn công mạng đầy đủ và chính xác.
* Cơ chế phân tích: Hệ thống cần được cấu hình và điều chỉnh sao cho có khả năng phân tích và phát hiện các mẫu tấn công mạng mới, bao gồm cả việc áp dụng các kỹ thuật máy học như học sâu, học tăng cường và học không giám sát.
* Quản lý và bảo trì: Hệ thống cần được quản lý và bảo trì định kỳ để đảm bảo tính ổn định và hiệu quả của hệ thống. Việc quản lý và bảo trì bao gồm cập nhật các phiên bản mới nhất của phần mềm, điều chỉnh cấu hình, đảm bảo tính bảo mật của hệ thống và thực hiện các hoạt động khác như backup và khôi phục dữ liệu.

Tổng quan, hệ thống IDS/IPS Suricata có sử dụng máy học để phát hiện và phòng chống các hình thức tấn mạng yêu cầu một cơ sở hạ tầng mạnh mẽ, các phần mềm và công cụ cần thiết để tri

## 2.2. Yêu cầu chức năng

* Phát triển và triển khai hệ thống IDS/IPS Suricata sử dụng các kỹ thuật máy học để phát hiện và ngăn chặn các hình thức tấn công mạng.
* Xây dựng các bộ dữ liệu để huấn luyện và đánh giá hiệu suất của hệ thống.
* Phát hiện các cuộc tấn công mạng: Suricata có khả năng phát hiện và báo cáo các cuộc tấn công mạng, bao gồm các cuộc tấn công được thực hiện thông qua mạng hoặc ứng dụng.

## 2.3. Yêu cầu phi chức năng

### 2.3.1. Độ tin cậy

Hệ thống phải có tính bảo mật cao để đảm bảo an toàn cho dữ liệu và hệ thống

Suricata phải đảm bảo độ tin cậy cao để không bị sai sót trong việc phát hiện các cuộc tấn công mạng.

### 2.3.2. Khả năng sử dụng

Suricata phải có khả năng tích hợp với các công cụ và hệ thống an ninh khác để có thể cung cấp cho người dùng thông tin chi tiết về các hoạt động diễn ra trên mạng. **2.3.3. Hiệu suất**

Hệ thống phải có khả năng hoạt động ổn định và có tính sẵn sàng cao để đáp ứng yêu cầu của người dùng

### 2.3.4. Bảo mật

Hệ thống phải có khả năng mở rộng để đáp ứng nhu cầu của các doanh nghiệp và tổ chức lớn

### 2.3.5. Khả năng hỗ trợ

Hệ thống phải có khả năng mở rộng để đáp ứng nhu cầu của các doanh nghiệp và tổ chức lớn. Suricata phải có khả năng dễ dàng bảo trì và nâng cấp để người quản trị hệ thống có thể duy trì và cập nhật hệ thống một cách dễ dàng và tiết kiệm thời gian.

### 2.3.6. Yêu cầu cấp phép

Phần mềm Suricata là một hệ thống IDS/IPS mã nguồn mở

### 2.3.7. Pháp lý, bản quyền và các thông báo khác

Phần mềm Suricata là một hệ thống IDS/IPS mã nguồn mở hợp pháp theo quy định của pháp luật.

### 2.3.8. Tiêu chuẩn áp dụng

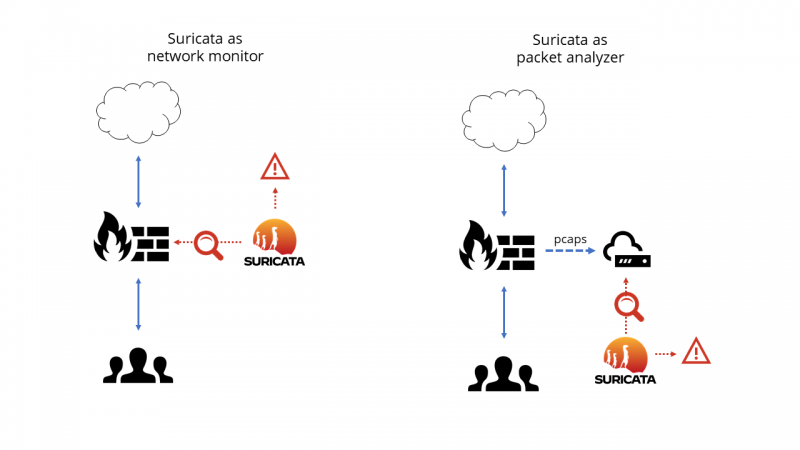
Áp dụng cho người dùng muốn giải trí, vẽ thỏa sức sáng tạo và giúp người dùng thoải mái hơn.

## 2.4. Các công việc cần giải quyết

* Tìm hiểu các dịch vụ IDS/IPS cứng và mềm.
* Tìm hiểu về các tập luật phát hiện xâm nhập.
* Tìm hiểu về máy học trong việc phát hiện xâm nhập.
* Xây dựng các tình huống tấn công mạng và thực nghiệm các phương pháp để phòng thủ bằng IDS/IPS bằng tập luật và máy học.
* Tìm hiểu và triển khai được các hình thức tấn công mạng.

# **III. Thiết kế**

1. **Mô hình hoạt động của Suricata**

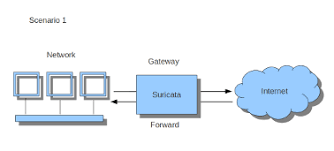


có thể được chia thành các bước chính như sau:

1. Tiếp nhận và phân tích gói tin mạng: Suricata nhận các gói tin mạng từ các thiết bị khác trong mạng, sau đó phân tích nội dung của các gói tin này để tìm kiếm các dấu hiệu của các cuộc tấn công.
2. Phát hiện và phân loại cuộc tấn công: Nếu Suricata phát hiện ra các dấu hiệu của cuộc tấn công trong các gói tin mạng, nó sẽ phân loại loại tấn công này và đưa ra cảnh báo cho quản trị viên mạng.
3. Thực thi các chính sách và quy tắc bảo mật: Dựa trên các chính sách và quy tắc bảo mật được thiết lập trước đó, Suricata sẽ thực hiện các hành động phù hợp để đối phó với tấn công này, chẳng hạn như chặn kết nối hoặc cảnh báo cho quản trị viên.
4. Ghi nhật ký và báo cáo: Suricata cũng có khả năng ghi lại các hoạt động và tạo ra báo cáo về các cuộc tấn công đã được phát hiện và các hành động đã được thực hiện để đối phó với chúng.

Tóm lại, mô hình hoạt động của Suricata là một quy trình liên tục và động, giúp ngăn chặn và đối phó với các cuộc tấn công mạng trong thời gian thực.

1. **Các thành phần chính của hệ thống bao gồm:**

****

* Thiết bị mạng (Network Devices): Là các thiết bị như Router, Firewall, Switch, hay các thiết bị mạng khác để giám sát và bắt các gói tin mạng.
* Suricata Sensors: Là các máy chủ hoặc thiết bị mạng được cài đặt Suricata, chịu trách nhiệm phát hiện và chặn các tấn công mạng.
* Suricata Manager: Là một phần mềm quản lý hệ thống Suricata Sensors. Suricata Manager giúp tập hợp, phân tích và lưu trữ các dữ liệu mà Suricata Sensors thu thập được. Nó cũng cung cấp các tính năng quản lý như cấu hình hệ thống, cảnh báo tấn công, xác thực người dùng, v.v.
* Cơ sở dữ liệu (Database): Lưu trữ dữ liệu về các tấn công mạng được phát hiện bởi Suricata Sensors và được quản lý bởi Suricata Manager.
* Hệ thống máy học (Machine Learning System): Là một phần của Suricata Manager, được sử dụng để phân tích và đưa ra các quyết định phát hiện các tấn công mạng. Hệ thống này được huấn luyện bằng các tập dữ liệu mẫu để tìm ra các mẫu tấn công và học cách phát hiện chúng.
* Giao diện người dùng (User Interface): Là một ứng dụng web, cung cấp cho người quản trị hệ thống các tính năng quản lý, giám sát và cảnh báo tấn công mạng.

## 3.3. Kiến trúc Của Suricata

Suricata được phát triển dựa trên snort nên nó vẫn giữ nguyên kiến trúc bên trong của snort. Kiến trúc của nó có nhiều thành phần, với mỗi thành phần có một chức năng riêng.

**Các thành phần chính suricata là:**

- modul giải mã gói tin

- modul tiền xử lý (preprocessors)

- modul phát hiện

- modul bản ghi và cảnh báo (logging and alerting system)

- modul kết xuất thông tin



Hình Các thành phần suricata

**Kiến trúc của suricata**

Khi suricata hoạt động nó sẽ thực hiện lắng nghe và thu bắt tất cả các gói tin nào di chuyển qua nó. Các gói tin sau khi bị bắt được đưa và modul giải mã gói tin. Tiếp theo gói tin sẽ được đưa vào modul tiền xử lý, rồi đưa vào modul phát hiện. Tại đây, tùy theo việc có phát hiện được xâm nhập hay không mà gói tin có thể được lưu thông tiếp hay được đưa vào modul bản ghi và cảnh báo để xử lý. Khi các cảnh báo được xác định modul kết xuất thông tin sẽ thực hiện việc đưa cảnh báo ra theo đúng định dạng mong muốn. Sau đây ta sẽ đi sâu vào nghiên cứu chi tiết hơn.

### *1.3.1. Module giải mã gói dữ liệu*

Suricata sử dụng thư viện PCap để bắt mọi gói tin trên mạng lưu thông qua hệ thống. Mỗi gói tin sau khi được giải mã sẽ được đưa tiếp vào modul tiền xử lý.

### *1.3.2. Module tiền xử lý*

Modul tiền xử lý là một modul rất quan trọng đối với bất kỳ hệ thống IDS nào để có thể chuẩn bị gói dữ liệu đưa vào cho modul phát hiện phân tích. Ba nhiệm vụ chính của modul này là:

- Kết hợp lại các gói tin: khi một lượng dữ liệu lớn được gửi đi, thông tin sẽ bị chia nhỏ thành nhiều gói tin. Khi suricata nhận được các gói tin này thì nó phải thực hiện ghép lại thành hình dạng ban đầu, từ đó mới thực hiện các công việc xử lý tiếp. Như ta đã biết khi một phiên làm việc diễn ra, sẽ có rất nhiều gói tin được trao đổi trong phiên đó. Một gói tin riêng rẽ sẽ không có trạng thái và nếu công việc phát hiện xâm nhập chỉ dựa vào gói tin đó sẽ không đem lại hiệu quả cao. Modul tiền xử lý giúp suricata hiểu được các phiên làm việc khác nhau từ đó giúp đạt được hiệu quả cao hơn trong việc phát hiện xâm nhập.

- Giải mã và chuẩn hóa giao thức (decode/normalize): công việc phát hiện xâm nhập dựa trên dấu hiệu nhận dạng nhiều khi bị thất bại khi kiểm tra các giao thức có dữ liệu có thể được thực hiện dưới nhiều hình thức khác nhau. Ví dụ: một web server có thể chấp nhận nhiều dạng URL như URL viết dưới dạng mã hexa/unicode, URL chấp nhận cả dấu / hay \ hoặc nhiều ký tự này liên tiếp cùng lúc. Chẳng hạn ta có dấu hiệu nhậ dạng “scripts/iiaadmin”, kẻ tấn công có thể vượt qua bằng cách tùy biến các yêu cầu gửi đến web server như sau:

*“scripts//iisadmin” “scripts/examples/../iisadmin” “scripts\iisadmin” “scripts/.\iisadmin”*

- Hoặc thực hiện mã hóa các chuỗi này dưới dạng khác. Nếu suricata chi thực hiện đơn thuần việc so sánh dữ liệu nhận dạng sẽ xảy ra tình trạng bỏ sót các hành vi xâm nhập. Do vậy, một số modul tiền xử lý 24 phải có nhiệm vụ giải mã và chỉnh sửa, sắp xếp lại các thông tin đàu vào này để thông tin khi đưa đến modul phát hiện cơ thể phát hiện được mà không bỏ sót.

- Phát hiện các xâm nhập bất thường (nonrule/anormal):thường dùng để đối phó với các xâm nhập không thể hoặc rất khó phát hiện được bằng luật thông thường hoặc các dấu hiệu bất thường trong giao thức. Các modul tiền xử lý dạng này có thể phát hiện xâm nhập theo bất cứ cách nào mà ta nghĩ ra từ đó tăng thêm tính năng cho suricata. Ví dụ: một plugin tiền xử lý có nhiêm vụ thống kê thông lượng mạng tại thời điểm bình thường để rồi khi có thông lượng bất thường xảy ra nó có thể tính toán, phát hiện và đưa ra cảnh báo.

### *1.3.3. Module phát hiện*

Đây là modul quan trong nhất. Nó chịu trách nhiệm phát hiện các dấu hiệu xâm nhập. Modul phát hiện sử dụng các luật được định nghĩa sẵn để so sánh với dữ liệu thu thập được từ đó xác định có xâm nhập xảy ra hay không. Rồi tiếp theo mới có thể thực hiện công việc ghi log, tạo báo cáo, kết xuất thông tin.

Một vấn đề quan trọng trong modul phát hiện là vấn đề thời gian xử lý các gói tin: IDS thường nhận được rất nhiều gói tin và bản thân nó cũng có rất nhiều luật xử lý. Vì vậy có thể mất những khoảng thời gian khác nhau cho việc xử lý các gói tin khác nhau. Và khi thông lượng qua mạng quá lớn có thể xảy ra việc bỏ sót hoặc không phản hồi đúng lúc. Khả năng xử lý của modul phát hiện dựa trên yếu tố như: số lượng các luật, tốc độ của hệ thống mạng.

Một modul phát hiện cũng có khả năng tách các phần của gói tin ra và áp dụng các luật trên từng phần của gói tin. Các phần đó có thể là:

- IP header

- Header của tầng vận chuyển: TCP, UDP

- Header của tầng ứng dụng: DNS header, HTTP header, …

- Phần tải của gói tin (bạn cũng có thể áp dụng các luật lên các phần dữ

liệu được truyền đi trong gói tin).

Một vấn đề trong modul phát hiện là việc xử lý thế nào khi một gói tin bị phát hiện đã được đánh thứ tự ưu tiên nên một gói tin khi bị phát hiện bởi nhiều luật khác nhau, cảnh báo được đưa ra ứng với luật có mức ưu tiên cao nhất.

### 1.3.4. Module bản ghi và cảnh báo

Tùy thuộc vào modul phát hiện có nhận dạng được xâm nhập hay không mà gói tin có thể bị ghi bản ghi hoặc đưa ra cảnh báo. Các file log là các file text, dữ liệu trong đó có thể được ghi dưới nhiều định dạng khác nhau chẳng hạn tcpdump.

### 1.3.5. Module kết xuất thông tin

Modul này có thể thức hiện các thao tác khác nhau tùy thuộc theo việc bạn muốn lưu kết quả xuất ra như thế nào. Tùy theo việc cấu hình hệ thống mà có thể thực hiện các công việc như là:

- Ghi log file

- Ghi syslog: syslog là một chuẩn lưu trữ các file log được sử dụng rất nhiều trên các hệ thống unix, linux.

- Ghi cảnh báo vào cơ sở dữ liệu.

- Tạo file log dạng xml: việc này rất thuận tiện cho việc trao đổi dữ liệu và chia sẻ dữ liệu.

- Cấu hình lại router, firewall.

- Gửi các cảnh báo được gói trong gói tin sử dụng giao thức SNMP. Các gói tin dạng này sẽ được gửi tới một SNMP server từ đó giúp cho việc quản lý các cảnh báo và hệ thống IDS một cách tập trung và thuật tiện.

- Gửi thông điệp SMB (server message block) tới các máy tính windows.

Ta cũng có thể tự viết modul kết xuất thông tin riêng tùy thoe mục đích sử dụng.

## 3.4. Suricata IPS

Suricata là một *IPS* (Hệ thống ngăn chặn xâm nhập), một hệ thống để phân tích xâm nhập mạng. Phần mềm phân tích tất cả lưu lượng truy cập trên tường lửa để tìm kiếm các cuộc tấn công và sự bất thường đã biết. Khi phát hiện một cuộc tấn công hoặc sự bất thường, hệ thống có thể quyết định có chặn lưu lượng truy cập hay chỉ đơn giản là lưu sự kiện trên nhật ký ( /var/log/suricata/fast.log). Suricata có thể được cấu hình bằng cách sử dụng các bộ quy tắc được tổ chức trong các danh mục thống nhất. Mỗi thể loại có thể được đặt thành:

• **Enable**: quy tắc khớp lưu lượng truy cập từ danh mục này sẽ được báo cáo.

• **Block**: quy tắc khớp lưu lượng truy cập từ danh mục này sẽ bị loại bỏ.

• **Disable:** quy tắc từ danh mục này được bỏ qua.

Việc sử dụng IPS tác động đến tất cả lưu lượng truy cập trên tường lửa. Hãy chắc chắn rằng bạn hiểu đầy đủ tất cả các hàm ý trước khi kích hoạt nó. Cụ thể, chú ý đến việc chặn các quy tắc có thể dừng cập nhật cho chính hệ thống.

Loại quy tắc:

• Activex: Tấn công và lỗ hổng (CVE, v.v.) liên quan đến Activex.

• Phản ứng tấn công:

Phản hồi chỉ ra sự xâm nhập khi tải xuống tập tin LMhost, một số biểu ngữ nhất định, lệnh tiêu diệt Metasploit Meterpreter được phát hiện, v.v ... Chúng được thiết kế để bắt các kết quả của một cuộc tấn công thành công. Những thứ như Id id = root, hoặc thông báo lỗi cho thấy sự thỏa hiệp có thể đã xảy ra.

• Botcc (Bot Command and Control):

Chúng được tự động phát sinh từ một số nguồn Botnet đang hoạt động đã biết và đã được xác nhận và các máy chủ chỉ huy và điều khiển khác. Cập nhật hàng ngày, nguồn dữ liệu chính là Shadowserver.org. Các quy tắc khối lệnh và điều khiển của Bot được tạo từ Shadowerver.org, cũng như spyeyetracker, palevotracker và zeustracker. Các quy tắc được nhóm theo cổng cung cấp độ trung thực cao hơn với cổng đích được sửa đổi theo quy tắc.

• Botcc Portgrouped:

Tương tự như trên, nhưng được nhóm theo cổng đích.

• Chat:

Xác định lưu lượng truy cập liên quan đến nhiều khách hàng trò chuyện, irc và hoạt động đăng ký có thể.

• CIArmy:

Trí tuệ tập thể đã tạo ra các quy tắc IP để chặn dựa trên www.cinsscore.com.

• Compromised:

Đây là danh sách các máy chủ bị xâm nhập đã biết, được xác nhận và cập nhật

hàng ngày. Bộ này thay đổi từ một trăm đến một vài quy tắc hunderd tùy thuộc vào các nguồn dữ liệu. Đây là một bản tổng hợp của một số nguồn dữ liệu riêng tư nhưng có độ tin cậy cao. Hâm nóng: Snort không xử lý IP phù hợp với tải thông minh. Nếu cảm biến của bạn đã được đẩy đến giới hạn, bộ này sẽ thêm tải đáng kể. Chúng tôi khuyên bạn nên ở lại với các quy tắc botcc trong trường hợp tải cao.

• Current Events:

Danh mục cho các chiến dịch hoạt động và sống ngắn. Danh mục này bao gồm các bộ dụng cụ khai thác và phần mềm độc hại sẽ bị lão hóa và loại bỏ nhanh chóng do tính chất ngắn ngủi của mối đe dọa. Chẳng hạn, các mục có cấu hình cao mà chúng tôi không mong đợi sẽ có các chiến dịch lừa đảo lâu dài liên quan đến thảm họa. Đây là những quy tắc mà chúng tôi không có ý định giữ trong quy tắc lâu dài hoặc cần phải được kiểm tra trước khi chúng được xem xét để đưa vào. Thông thường, đây sẽ là các trang web đơn giản cho URL nhị phân Storm trong ngày, các trang web để bắt các ứng dụng dễ bị tổn thương mới được tìm thấy của CLSID, nơi chúng tôi không có bất kỳ chi tiết nào về khai thác, v.v.

•Decoder-events:

Các quy tắc này ghi nhật ký các sự kiện chuẩn hóa liên quan đến giải mã.

DNS:

Quy tắc cho các cuộc tấn công và lỗ hổng liên quan đến DNS. Ngoài ra loại lạm dụng dịch vụ cho những thứ như đường hầm.

DOS:

Nỗ lực phát hiện tấn công từ chối dịch vụ. Dự định ngăn chặn tấn công DOS gửi đến và chỉ dẫn.

Drop:

Các quy tắc để chặn các thư rác được liệt kê trên mạng. Dựa trên IP. Đây là danh sách được cập nhật hàng ngày của danh sách Spamhaus DROP (Không lộ trình hoặc ngang hàng). Chủ yếu là những kẻ gửi thư rác chuyên nghiệp.

**Dshield:**

Quy tắc dựa trên IP cho những kẻ tấn công được xác định Dshield. Danh sách cập nhật hàng ngày của danh sách những kẻ tấn công hàng đầu DShield. Cũng rất đáng tin cậy.

Khai thác:

Khai thác không được bao gồm trong thể loại dịch vụ cụ thể. Quy tắc phát hiện khai thác trực tiếp. Nói chung nếu bạn đang tìm kiếm một cửa sổ khai thác, Veritas, v.v., chúng sẽ ở đây. Những thứ như SQL injection và những thứ tương tự, chúng được khai thác, có thể loại riêng.

• Các tập tin:

Ví dụ quy tắc sử dụng chức năng xử lý và trích xuất tệp trong Suricata.

• FTP:

Quy tắc cho các cuộc tấn công, khai thác và lỗ hổng liên quan đến FTP. Cũng bao gồm cơ bản không có hoạt động FTP độc hại nào cho mục đích ghi nhật ký, chẳng hạn như đăng nhập, v.v.

• Games:

Quy tắc xác định lưu lượng truy cập trò chơi và các cuộc tấn công chống lại các trò chơi đó. World of Warcraft, Starcraft và các trò chơi trực tuyến phổ biến khác có hoạt động ở đây. Chúng tôi không có ý định gắn nhãn những thứ xấu xa này, chỉ là chúng không phù hợp với mọi môi trường.

• HTTP-Events:

Quy tắc để ghi nhật ký các sự kiện cụ thể của giao thức HTTP, thường hoạt động bình thường.

• Inappropriate:

Quy tắc để xác định các hoạt động liên quan đến các trang web đen. Bao gồm trang web Khiêu dâm, Khiêu dâm Kiddy, các trang web bạn không nên truy cập tại nơi làm việc, v.v. Chúng thường khá nặng Regex và do đó tải cao và thường xuyên bị lỗi. Chỉ chạy những thứ này nếu bạn thực sự quan tâm.

• Phần mềm độc hại:

Phần mềm độc hại và phần mềm gián điệp liên quan, không có ý định tội phạm rõ ràng. Ngưỡng để đưa vào bộ này thường là một số hình thức theo dõi dừng hoạt động tội phạm rõ ràng. Bộ này ban đầu được dự định chỉ là phần mềm gián điệp. Điều đó đủ cho một số loại quy tắc thực sự. Ranh giới giữa phần mềm gián điệp và những thứ độc hại hoàn toàn đã bị xóa nhòa kể từ khi chúng tôi bắt đầu thiết lập này. Có nhiều thứ hơn là phần mềm gián điệp ở đây, nhưng hãy yên tâm rằng không có gì ở đây là thứ bạn muốn chạy trên mạng hoặc PC của bạn. Có các móc nối URL cho các bản cập nhật đã biết được mô tả, các chuỗi Tác nhân người dùng của phần mềm độc hại đã biết và tải các phần mềm khác.

• Phần mềm độc hại di động:

Cụ thể đối với các nền tảng di động: Phần mềm độc hại và phần mềm gián điệp liên quan, không có mục đích phạm tội rõ ràng.

• Netbios:

Các quy tắc để nhận dạng, cũng như các cuộc tấn công, khai thác và lỗ hổng liên quan đến Netbios. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký hoạt động của người dùng.

• P2P:

Quy tắc xác định lưu lượng truy cập ngang hàng và các cuộc tấn công chống lại. Bao gồm torrents, edonkey, Bittorrent, Gnutella, Limewire, v.v.

• Chính sách:

Danh mục nhận dạng ứng dụng. Bao gồm chữ ký cho các ứng dụng như DropBox và Google Apps, v.v. Cũng bao gồm các giao thức cổng, DLP cơ bản như số thẻ tín dụng và số an sinh xã hội. Bao gồm trong bộ này là các quy tắc cho những thứ thường không được chính sách của công ty hoặc tổ chức không cho phép. Myspace, Ebay, v.v.

• Shellcode:

Phát hiện Shellcode từ xa. Shellcode từ xa được sử dụng khi kẻ tấn công muốn nhắm mục tiêu vào một quá trình dễ bị tấn công đang chạy trên một máy khác trên mạng cục bộ hoặc mạng nội bộ. Nếu được thực hiện thành công, shellcode có thể cung cấp cho kẻ tấn công quyền truy cập vào máy đích trên toàn mạng. Mã hóa từ xa thường sử dụng các kết nối ổ cắm TCP / IP tiêu chuẩn để cho phép kẻ tấn công truy cập vào vỏ trên máy mục tiêu. Shellcode như vậy có thể được phân loại dựa trên cách thiết lập kết nối này: nếu shellcode có thể thiết lập kết nối này, nó được gọi là shell shell ngược hoặc một shellcode kết nối lại vì shellcode kết nối lại với máy của kẻ tấn công.

• SMTP:

Các quy tắc cho các cuộc tấn công, khai thác và các lỗ hổng liên quan đến SMTP. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký.

• TELNET:

Quy tắc cho các cuộc tấn công và lỗ hổng liên quan đến dịch vụ TELNET. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký.

• VOIP:

Quy tắc cho các cuộc tấn công và lỗ hổng liên quan đến môi trường VOIP. SIP, h.323, RTP, v.v.

• SQL:

Các quy tắc cho các cuộc tấn công, khai thác và các lỗ hổng liên quan đến SQL. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký.

• SNMP:

Quy tắc cho các cuộc tấn công, khai thác và lỗ hổng liên quan đến SNMP. Cũng bao gồm các quy tắc phát hiện hoạt động cơ bản của giao thức cho mục đích ghi nhật ký.

• Trojan:

Phần mềm độc hại có ý định tội phạm rõ ràng. Các quy tắc ở đây phát hiện phần mềm độc hại đang truyền, đang hoạt động, lây nhiễm, tấn công, cập nhật và bất cứ điều gì khác có thể phát hiện trên dây. Đây cũng là một quy tắc rất quan trọng để chạy nếu phải chọn.

• EveBox:

EveBox là một trang web dựa trên công cụ quản lý sự kiện và cảnh báo cho các sự kiện do Suricata tạo ra.

# **IV. Thực hiện**

## 4.1. Cách cài đặt Suricata

### 4.1.1. Điều kiện cần để cài đặt Suricata

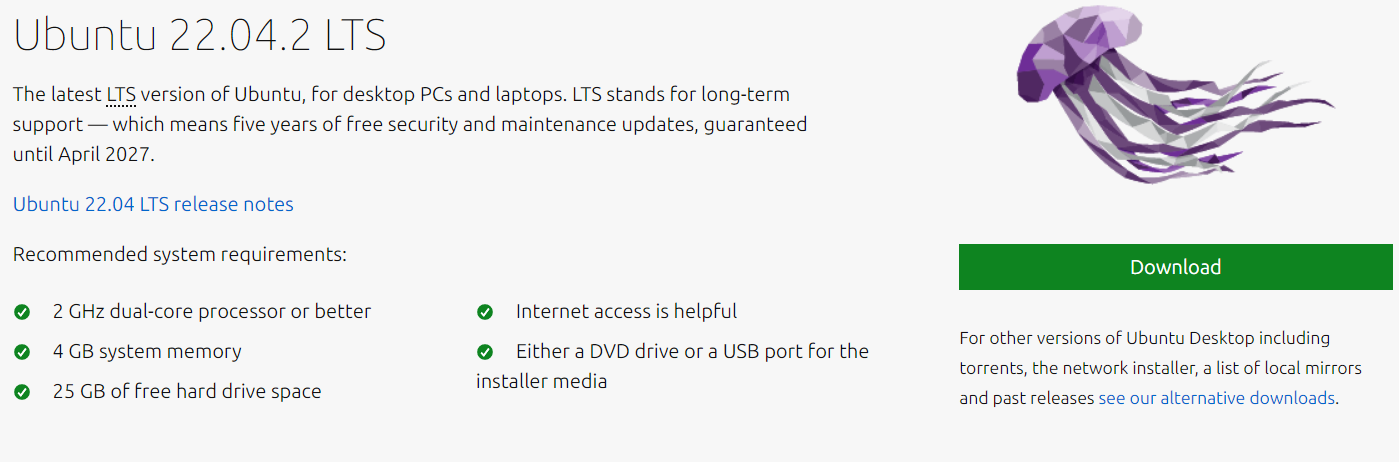
Để cài đặt Suricata chúng ta cần có một máy chủ Ubuntu 20.04 với 2 CPU trở lên.

### 4.1.2. Các bước để cài đặt Suricata

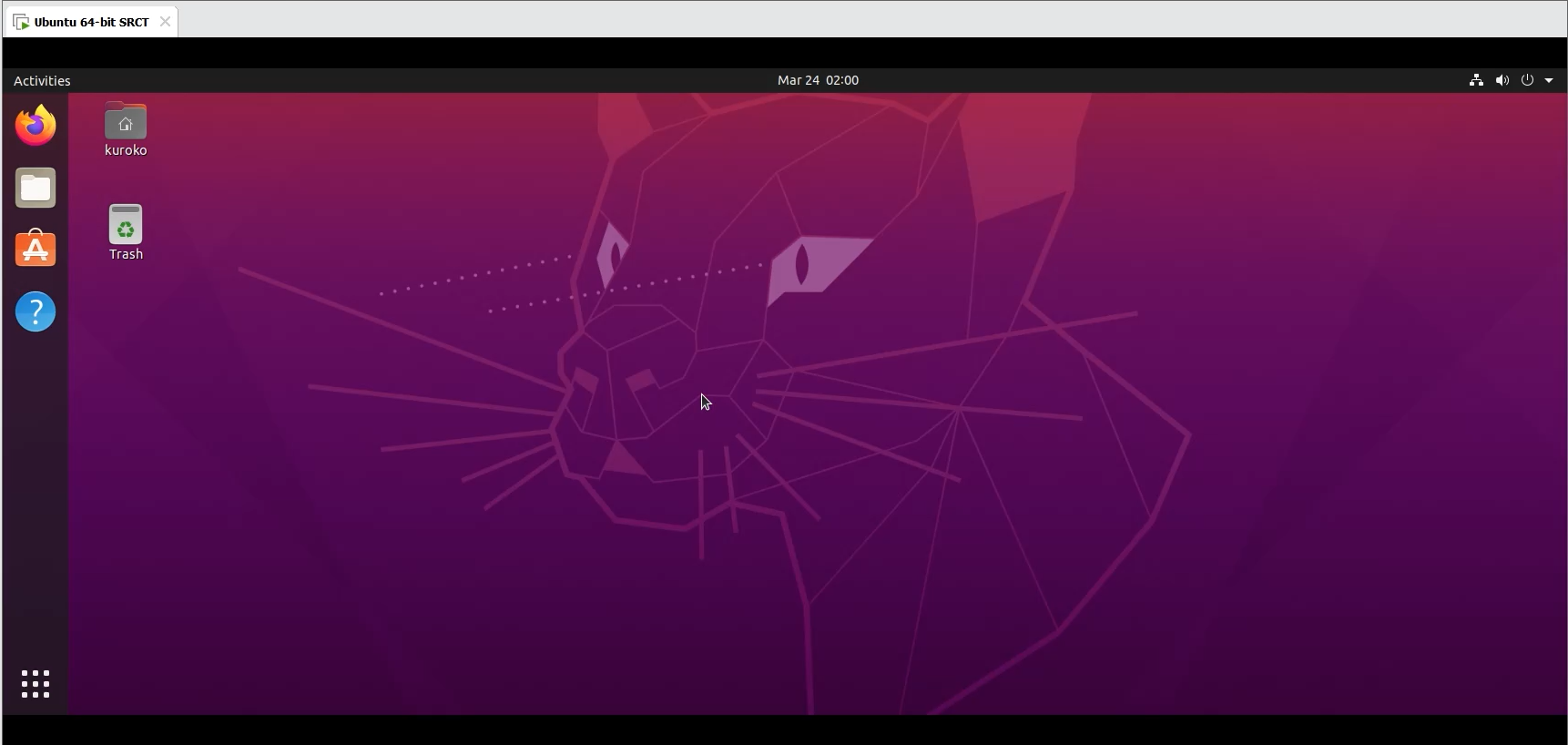
**Bước 1:** Cài đặt máy ảo trên **Vmware Workstation Pro**.



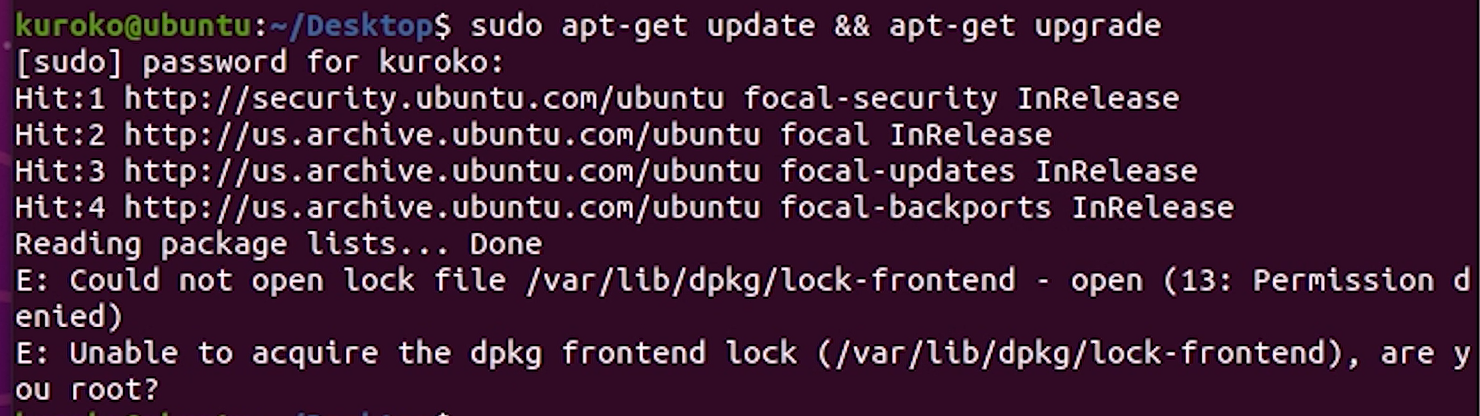
**Bước 2:** Tải file **ISO** của **Ubuntu**, để cài vào máy ảo. Tải qua đường link này <https://ubuntu.com/download/desktop> . Chọn phiên bản phù hợp và tải xuống.



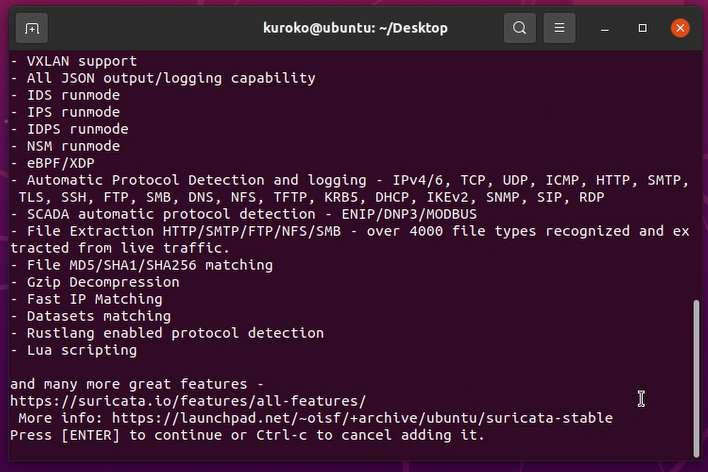
**Bước 3:** Cài đặt và cấu hình máy ảo. Là chúng ta đã xong phần máy ảo.



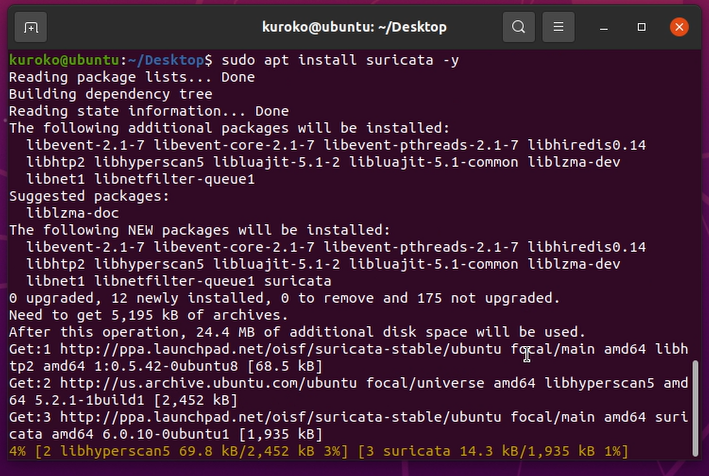
**Bước 4:** Mở **Terminal** và chạy lệnh “**sudo apt-get update && apt-get upgrade**” để cập nhật hệ thống.



**Bước 5:** Chạy lệnh “**sudo add-apt-repository ppa:oisf/suricata-stable**” để thêm repository của Open Information Security Foundation (OISF) vào hệ thống của Ubuntu. Nhấn Enter để cập nhật các pakage.



**Bước 6:** Chạy lệnh “**sudo apt-install suricata -y**” để cái đặt Suricata về máy ảo.



**Bước 7:** Sau khi đã cài đặt thành công, chúng ta sẽ có 3 lệnh sẽ liên quan đến Suricata.

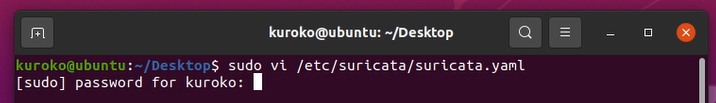
**Lệnh 1** dùng để khởi động Suricata “**sudo systemctl enable suricata.service**”.

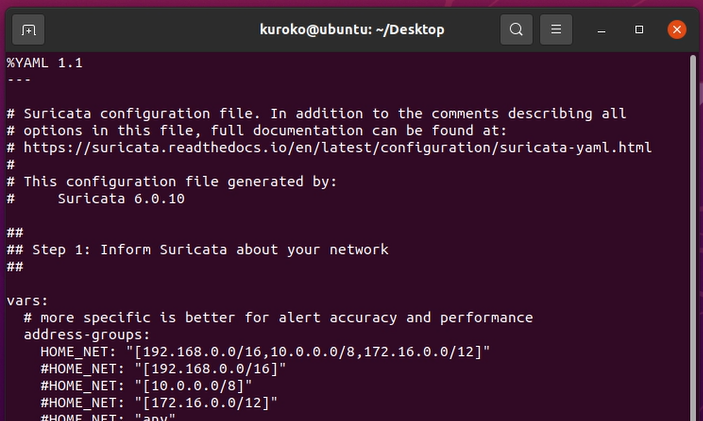
**Lệnh 2** dùng để kiểm tra trạng thái của Suricata “**sudo systemctl status suricata.service**”.

**Lệnh 3** dùng để tắt Suricata “**sudo systemctl stop suricata.service**”.

## 4.2. Các bước để cấu hình Suricata

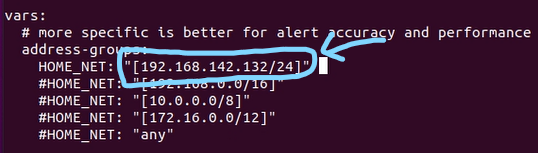
**Bước 1:** Chạy Terminal và chạy lệnh “**sudo vi /etc/suricata/suricata.yaml**” để mở bản cấu hình của Suricata.



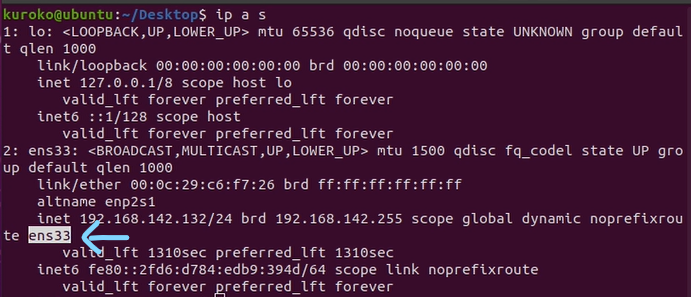


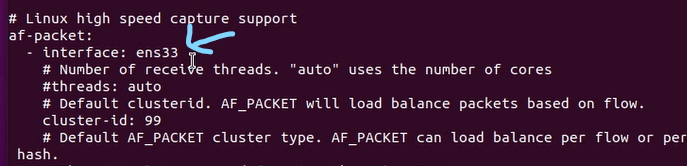
**Bước 2:** Sau khi mở bản ghi suricata.yaml, chúng ta sẽ tiến hành chỉnh những cái sau.

**Chỉnh HOME\_NET:** Chúng ta mở song song Terminal khác để lấy địa chỉ ipv4 của máy ảo và bỏ vào bản ghi tại dòng **HOME\_NET**. Chạy lệnh “**ip a s**”.

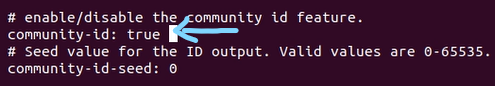


**Chỉnh Interface:** Khi chạy lệnh “ip a s” thì nó cung cấp cho ta địa chỉ của máy ảo và Interface trong đó. Sau đó chúng ta gõ “**/af-packet**” để tới dòng đó và chỉnh Interface.





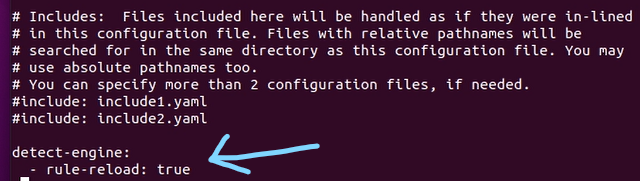
**Chỉnh Community-ID:** Bằng cách gõ lệnh “**/community-id**” để tới dòng đó và chỉnh cho Community-id = **true**.



**Thêm Live Rule Reloading:** Việc này sẽ giúp cho chúng ta thêm, xóa và chỉnh sửa quy tắc mà không cần khởi động lại tiến trình Suricata đang chạy. Hãy thêm dòng này vào đoạn cuối cùng của bản ghi **suricata.yaml** như trong hình dưới.

**detect-engine:**

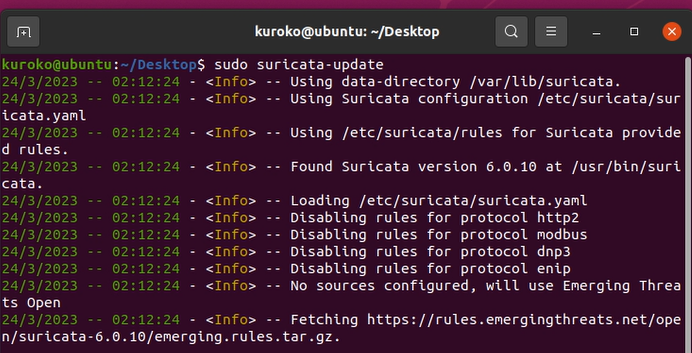
**- rule-reload: true**



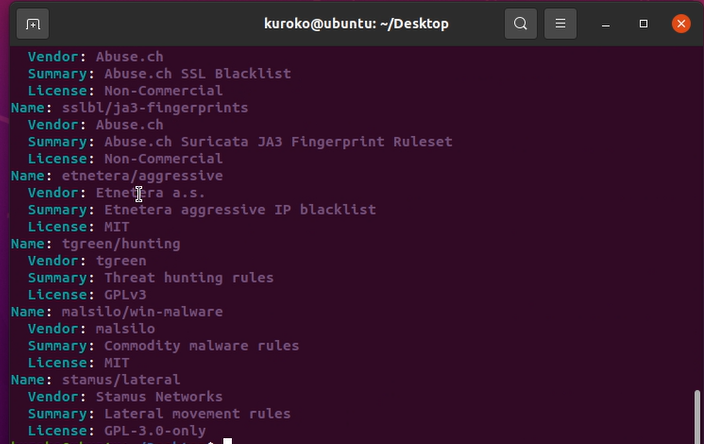
Và thế là chúng ta đã xong phần cấu hình cho Suricata.

## 4.3. Cập nhật Suricata Rulesets

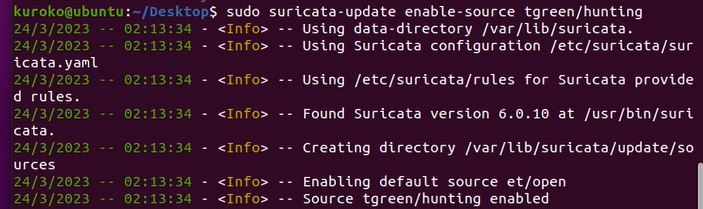
**Bước 1:** Chạy lệnh “**sudo suricata-update**” để tải các bộ quy tắc về máy áo.



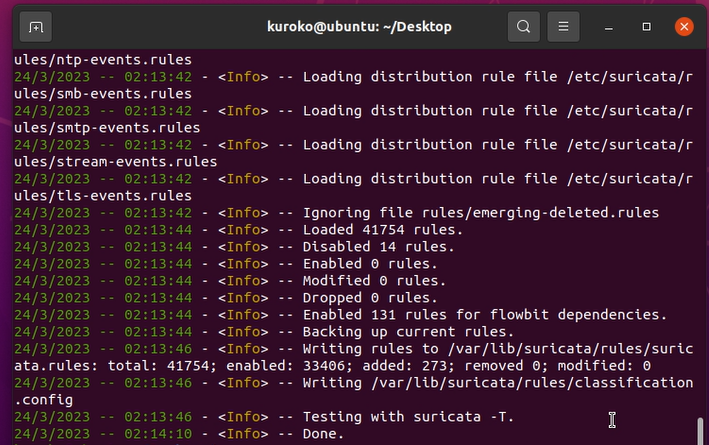
**Bước 2:** Chạy lệnh “**sudo suricata-update list-sources**” để xem danh sách các bộ quy tắc của các nhà cung cấp từ miễn phí đến trả phí.



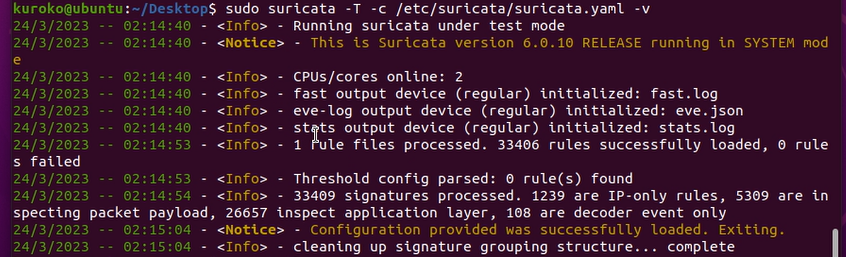
**Bước 3:** Chạy lệnh “**sudo suricata-update enable-source tgreen/hunting**” để tải bộ quy tắc từ nhà cung cấp về máy ảo. Đó chỉ là ví dụ tải về 1 quy tắc từ 1 nhà cung cấp, các bạn có thể tải quy tắc từ các nhà cung cấp khác. “**sudo suricata-update enable-source <Tên của bộ quy tắc>**”.



**Bước 4:** Sau khi chúng ta tải xong 1 bộ quy tắc thì chúng ta chạy lệnh “**sudo suricata-update**” để thêm bộ quy tắc đó vào.



**Bước 5:** Chạy lệnh “**sudo suricata -T -c /ect/suricata/suricata.yaml**” để xác thực cấu hình cho Suricata và xong phần thêm các bộ quy tắc.



## 4.4. Thử nghiệm Suricata Rules

Thử nghiệm dựa theo Quickstart: Bộ quy tắc ET Open mà chúng ta đã tải xuống chứa hơn 30000 quy tắc. Việc kiểm tra xem Suricata có đang phát hiện lưu lượng truy cập đáng ngờ với cấu hình mà chúng ta đã tạo hay không.

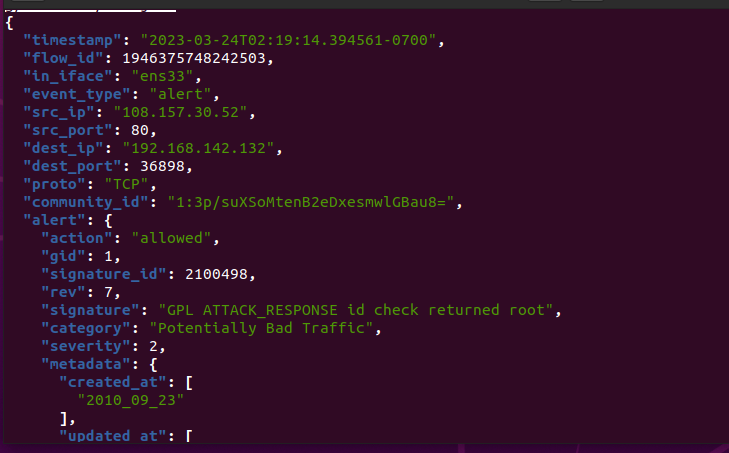
Thực thi lệnh sau tạo một yêu cầu HTTP, yêu cầu này sẽ trả về phản hồi phù hợp với quy tắc cảnh báo của Suricata “**curl** [**http://testmynids.org/uid/index/html**](http://testmynids.org/uid/index/html)”.

Kết quả sẽ ra là:



Dữ liệu phản hồi ví dụ này được thiết kế để kích hoạt cảnh báo, bằng cách trả lại kết quả đầu ra của một lệnh như id vậy có thể chạy trên hệ thống từ xa bị xâm nhập thông qua web shell.

Gõ lệnh “**sudo jq 'select(.alert .signature\_id==2100498)' /var/log/suricata/eve.json**” để kiểm tra báo cáo và cảnh báo từ Suricata.



# **V. Kết luận và hướng phát triển**

## 5.1. Ưu điểm

* Áp dụng các kỹ thuật máy học để cải thiện khả năng phát hiện và phòng chống các hình thức tấn công mạng.
* Sử dụng Suricata - một hệ thống IDS/IPS phổ biến và mã nguồn mở, giúp giảm chi phí và tăng tính linh hoạt.
* Hệ thống có khả năng mở rộng, dễ dàng bảo trì và cập nhật.
* Giúp cải thiện hiệu suất của hệ thống IDS/IPS, giúp người quản trị mạng phát hiện và phòng chống các mối đe dọa mạng một cách nhanh chóng và chính xác hơn.
* kết quả của nghiên cứu này có thể được áp dụng để tăng cường an ninh mạng cho các tổ chức và doanh nghiệp, giúp bảo vệ thông tin và tài sản quan trọng của họ.

## 5.2. Nhược điểm

* Còn thiếu kiến thức chuyên môn về mạng máy tính, bảo mật mạng và kỹ thuật máy học để thực hiện được nghiên cứu này.
* Chi phí cho việc triển khai hệ thống IDS/IPS Suricata có thể khá cao, đặc biệt là đối với các tổ chức hoặc doanh nghiệp nhỏ.
* Các kỹ thuật máy học có thể đòi hỏi tài nguyên máy tính và bộ nhớ cao, do đó cần phải có hệ thống phần cứng mạnh mẽ để đảm bảo hiệu suất của hệ thống.
* Chưa thể mô phỏng được các tấn công mới và phức tạp.
* Việc triển khai và sử dụng các kỹ thuật máy học có thể cần phải tuân theo các quy định về quyền riêng tư và an ninh thông tin.

## 5.3. Hướng phát triển

Nhóm em sẽ tiếp tục trao dồi kiến thức, nghiên cứu, học tập và tìm hiểu từ thầy/cô, bạn bè, mạng xã hội để có thể nghiên cứu và hoàn thiện đồ án nghiên cứu này như tối ưu hóa hiệu suất của hệ thống IDS/IPS Suricata để đảm bảo rằng nó có thể xử lý các tấn công mạng phức tạp và đa dạng. Đánh giá hiệu quả của hệ thống IDS/IPS Suricata trên các môi trường mạng khác nhau bao gồm các môi trường có quy mô lớn và độ phức tạp cao. Kết hợp các phương pháp phát hiện mối đe dọa truyền thống với kỹ thuật máy học để tăng cường khả năng phát hiện và giảm thiểu các dấu giả mạo.

# **CÁC TÀI LIỆU THAM KHẢO**

[1] Roesch, M. (1999). Snort - lightweight intrusion detection for networks. In Proceedings of the 13th USENIX conference on System Administration (pp. 229-238). USENIX Association.

[2] Casado, M., & Wu, M. (2013). VMware network virtualization. Communications of the ACM, 56(10), 84-93.

[3] Suricata. (2021). About Suricata. <https://suricata-ids.org/about/>

[4] Garcia, S., Grill, M., Stajano, F., & Wilson, A. (2014). Evolutionary computation in the optimization of intrusion detection systems: A survey. Journal of Heuristics, 20(1), 1-24.

[5] Tsai, C. F., & Wang, Y. C. (2015). Design and implementation of a high performance intrusion detection system based on Snort. Journal of Information Science and Engineering, 31(4), 1101-1117.

[6] Network Working Group. (1999). Request for Comments: 2401. Security architecture for the Internet Protocol. <https://www.rfc-editor.org/rfc/rfc2401.html>

[7] VMware. (2021). VMware Workstation Documentation. Retrieved from <https://docs.vmware.com/en/VMware-Workstation-Pro/index.html>

[8] Vuong, T. T., & Nguyen, D. T. (2020). Deep learning for network intrusion detection: A review. International Journal of Network Security, 22(6), 1016-1035.