**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**CHUYÊN NGÀNH MẠNG**

**ĐỀ TÀI :**

**Ứng dụng các công cụ của Security Onion nhằm phân tích và phát hiện một cuộc tấn công (Web, Database,…)**

**SINH VIÊN : Trương Quang Hùng**

**MÃ SINH VIÊN : 102170022**

**LỚP : 17T1**

**GVHD : PGS.TS. Nguyễn Tấn Khôi**

**Đà Nẵng, 04/2021**

**TÓM TẮT**

Tên đề tài: Ứng dụng các công cụ của Security Onion nhằm phân tích và phát hiện một cuộc tấn công (Web, Database,…)

Sinh viên thực hiện: Trương Quang Hùng

Số thẻ SV: 102170022

Lớp: 17T1

1. Giới thiệu:

Dưới sự phát triển của cuộc cách mạng công nghiệp 4.0 vai trò của mạng ngày càng thể hiện rõ rệt. Tuy nhiên việc tham gia vào môi trường mạng khiến người sử dụng luôn phải đối mặt với nhiều nguy cơ khác nhau: từ việc bị thăm dò, khai thác dữ liệu đến khả năng gặp phải các cuộc tấn công về Web, Database,…

Security Onion là một bản phân phối mã nguồn mở của Linux với nhiều công cụ hữu dụng có thể phân tích, phát hiện các mối nguy hại trong môi trường mạng dựa vào việc phân tích các gói tin gửi đi trong mạng. Bằng cách cài đặt và cấu hình phù hợp, Security Onion có thể đóng vai trò như một hệ thống phát hiện xâm nhập mạng (NIDS – Network Intrusion Detection System) giúp người dùng có thể dễ dàng phát hiện các mối nguy hại đến các máy trong mạng.

1. Các bước thực hiện đề tài:

* Tìm hiểu đề tài, phân tích các yêu cầu cần thiết để xây dựng hệ thống.
* Tiến hành thiết kế mô hình mạng, cài đặt môi trường mạng trên GNS3.
* Tải, cài đặt các máy ảo cần dùng: Kali linux, Security Onion, Fedora 14.
* Mô phỏng sử dụng Security Onion để phân tích và phát hiện các hành động thăm dò, tấn công Fedora 14 của Kali linux.

1. Kết quả đạt được:

Nắm bắt được cách thực hoạt động của Security Onion khi thực hiện phân tích các gói tin để phát hiện nguy cơ tấn công trong môi trường mạng. Hiểu cấu trúc của bộ luật sử dụng trong Security Onion từ đó có thể bổ sung giúp tăng khả năng phát hiện xâm nhập của hệ thống. Nâng cao kiến thức, kinh nghiệm của bản thân khi làm việc trong môi trường mạng.

**LỜI CẢM ƠN**

Trong lời đầu tiên của bản báo cáo đồ án chuyên nhành Mạng này, em mong muốn  
gửi lời cảm ơn và biết ơn chân thành nhất của mình tới tất cả những người đã hỗ trợ,   
giúp đỡ chúng em về kiến thức và tinh thần trong quá trình thực hiện làm đồ án.

Trước hết, em xin chân thành cảm ơn thầy - PGS.TS. Nguyễn Tấn Khôi, giảng viên khoa Công nghệ thông tin, trường đại học Bách Khoa – Đại học Đà Nẵng, người đã trực tiếp hướng dẫn, nhận xét, giúp đỡ em trong suốt quá trình thực hiện đồ án này. Em xin chân thành cảm ơn Ban giám hiệu nhà trường, các thầy cô trong khoa Công nghệ thông tin và các phòng ban nhà trường đã tạo điều kiện tốt nhất cho em cũng như các bạn sinh viên khác trong suốt quá trình học tập và làm đồ án.

Do thời gian thực hiện có hạn, kiến thức còn nhiều hạn chế nên đồ án chuyên ngành Mạng chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của các thầy cô giáo và các bạn để em có thêm kinh nghiệm, hoàn thiện và phát triển đề tài hay đồ án trong tương lai.

Em xin chân thành cảm ơn!

**LỜI CAM ĐOAN**

Tôi xin cam đoan:

1. Nội dung trong đồ án này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của PGS.TS. Nguyễn Tấn Khôi.

2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.

3. Nếu có những sao chép không hợp lệ, vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

**Trương Quang Hùng – 17T1**

**MỤC LỤC**

[**CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI** 2](#_Toc73521620)

[**1.1.** **Tổng quan về Security Onion** 2](#_Toc73521621)

[***1.1.1.*** ***Giới thiệu*** 2](#_Toc73521622)

[***1.1.2.*** ***Chức năng*** 2](#_Toc73521623)

[***1.1.3.*** ***Ưu điểm*** 3](#_Toc73521624)

[***1.1.4.*** ***Nhược điểm*** 4](#_Toc73521625)

[**1.2.** **Công nghệ phát triển** 4](#_Toc73521626)

[***1.2.1.*** ***Hệ thống phát hiện xâm nhập mạng*** 4](#_Toc73521627)

[***1.2.2.*** ***Kiến trúc*** 5](#_Toc73521628)

[***1.2.3.*** ***Các công cụ NIDS*** 5](#_Toc73521629)

[***1.2.4.*** ***Công cụ Sguil*** 5](#_Toc73521630)

[***1.2.5.*** ***Công cụ Squert*** 6](#_Toc73521631)

[***1.2.6.*** ***Bộ luật của Security Onion*** 7](#_Toc73521632)

[**1.3.** **Các kiểu xâm nhập cơ bản** 9](#_Toc73521633)

[***1.3.1.*** ***Thăm dò thông tin sử dụng Nmap*** 9](#_Toc73521634)

[***1.3.2.*** ***Tấn công Cross-Site Scripting (XSS)*** 9](#_Toc73521635)

[***1.3.3.*** ***Tấn công SQL injection*** 10](#_Toc73521636)

[**1.4.** **Mô hình hệ thống phát hiện xâm nhập mạng** 11](#_Toc73521637)

[***1.4.1.*** ***Chương trình mô phỏng GNS3*** 11](#_Toc73521638)

[***1.4.2.*** ***Hệ điều hành Kali Linux*** 12](#_Toc73521639)

[***1.4.3.*** ***Ứng dụng web Damn vulnerable web application và Metasploitable*** 12](#_Toc73521640)

[**Kết chương** 14](#_Toc73521641)

[**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 15](#_Toc73521642)

[**2.1.** **Phân tích yêu cầu** 15](#_Toc73521643)

[***2.1.1.*** ***Mục tiêu đề tài*** 15](#_Toc73521644)

[**2.2.** **Thiết kế hệ thống** 15](#_Toc73521645)

[***2.2.1.*** ***Sơ đồ kiến trúc hệ thống*** 15](#_Toc73521646)

[***2.2.2.*** ***Phân tích hệ thống*** 15](#_Toc73521647)

[**Kết chương** 16](#_Toc73521648)

[**CHƯƠNG 3: XÂY DỰNG VÀ MÔ PHỎNG HỆ THỐNG** 17](#_Toc73521649)

[**3.1.** **Xây dựng sơ đồ mạng bằng GNS3** 17](#_Toc73521650)

[***3.1.1.*** ***Cài đặt GNS3 và Router Cisco*** 17](#_Toc73521651)

[***3.1.2.*** ***Cấu hình Router Cisco*** 23](#_Toc73521652)

[**3.2.** **Xây dựng mô hình hệ thống phát hiện xâm nhập mạng** 24](#_Toc73521653)

[***3.2.1.*** ***Cài đặt Security Onion*** 24](#_Toc73521654)

[***3.2.2.*** ***Cài đặt Kali Linux*** 31](#_Toc73521655)

[***3.2.3.*** ***Cài đặt Metasploitable*** 32](#_Toc73521656)

[**3.3.** **Mô phỏng hệ thống phát hiện xâm nhập** 33](#_Toc73521657)

[***3.3.1.*** ***Cấu hình Security Onion*** 33](#_Toc73521658)

[***3.3.2.*** ***Chức năng phát hiện thăm dò dữ liệu bằng nmap*** 35](#_Toc73521659)

[***3.3.3.*** ***Chức năng phát hiện tấn công XSS*** 39](#_Toc73521660)

[***3.3.4.*** ***Chức năng phát hiện SQL injection*** 41](#_Toc73521661)

[***3.3.5.*** ***Thêm luật giúp nâng cao khả năng phát hiện xâm nhập của hệ thống*** 43](#_Toc73521662)

[**Kết chương** 46](#_Toc73521663)

[**KẾT LUẬN**](#_Toc73521649) 47

[**TÀI LIỆU THAM KHẢO**](#_Toc73521649) 48

[**PHỤ LỤC**](#_Toc73521649) 49

**DANH SÁCH HÌNH ẢNH**

[Hình 1.1: Hệ thống phát hiện xâm nhập mạng (NIDS) 3](#_Toc73541424)

[Hình 1.2: Sự khác nhau giữa cách cài đặt IDS và IPS trong môi trường mạng 4](#_Toc73541425)

[Hình 1.3: Giao diện Sguil 6](#_Toc73541426)

[Hình 1.4: Giao diện Squert 7](#_Toc73541427)

[Hình 1.5: Cấu trúc cách viết luật trong Security Onion 7](#_Toc73541428)

[Hình 1.6: Mô tả quá trình thực hiện tấn công XSS 10](#_Toc73541429)

[Hình 1.7: Mô tả quá trình thực hiện tấn công SQL injection 11](#_Toc73541430)

[Hình 1.8: Giao diện ứng dụng GNS3 11](#_Toc73541431)

[Hình 1.9: Giao diện của hệ điều hành Kali Linux 12](#_Toc73541432)

[Hình 1.10: Giao diện ứng dụng web Damn Vulnerable Web Application (DVWA) 13](#_Toc73541433)

[Hình 1.11: Giao diện máy ảo Metasploitable 14](#_Toc73541434)

[Hình 2.1: Sơ đồ hệ thống tấn công 15](#_Toc73541435)

[Hình 3.1: Giao diện ứng dụng GNS3 phiên bản 1.3.13 17](#_Toc73541436)

[Hình 3.2: Tải file image của Router để cài đặt trên GNS3 18](#_Toc73541437)

[Hình 3.3: Giao diện Preferences của GNS3 18](#_Toc73541438)

[Hình 3.4: Giao diện thêm router ảo của GNS3 19](#_Toc73541439)

[Hình 3.5: Chọn đường dẫn đến file image của router ảo 19](#_Toc73541440)

[Hình 3.6: Cài tên cho router ảo 20](#_Toc73541441)

[Hình 3.7: Cài bộ nhớ RAM cho router ảo 20](#_Toc73541442)

[Hình 3.8: Thêm cổng nếu muốn cho router ảo 21](#_Toc73541443)

[Hình 3.9: Thêm cổng Serial nếu muốn cho router ảo 21](#_Toc73541444)

[Hình 3.10: Đợi máy tự cấu hình phù hợp cho router ảo 22](#_Toc73541445)

[Hình 3.11: Router Cisco c3725 ảo sau khi được cài đặt vào GNS3 22](#_Toc73541446)

[Hình 3.12: Bảng định tuyến của router 24](#_Toc73541447)

[Hình 3.13: Bảng phân chia DHCP của router 24](#_Toc73541448)

[Hình 3.14: Cửa sổ tạo máy ảo của VMware 25](#_Toc73541449)

[Hình 3.15: Chọn đường dẫn đến file iso của máy ảo 25](#_Toc73541450)

[Hình 3.16: Chọn hệ điều hành cho máy ảo 26](#_Toc73541451)

[Hình 3.17: Đặt tên cho máy ảo 26](#_Toc73541452)

[Hình 3.18: Cài đặt bộ nhớ cho máy ảo 27](#_Toc73541453)

[Hình 3.19: Mở cửa sổ cài đặt phần cứng 27](#_Toc73541454)

[Hình 3.20: Cài đặt phần cứng cho máy ảo 28](#_Toc73541455)

[Hình 3.21: Giao diện máy ảo mới thêm vào VMware 28](#_Toc73541456)

[Hình 3.22: Giao diện chọn ngôn ngữ cho Security Onion 29](#_Toc73541457)

[Hình 3.23: Chọn cách thức cài đặt Security Onion 29](#_Toc73541458)

[Hình 3.24: Cài đặt tài khoản sử dụng 30](#_Toc73541459)

[Hình 3.25: Khởi động lại máy ảo 30](#_Toc73541460)

[Hình 3.26: Máy ảo Security Onion với địa chỉ được cấp phát tự động từ DHCP\_Router 31](#_Toc73541461)

[Hình 3.27: Máy ảo Kali Linux với địa chỉ được cấp phát tự động từ DHCP\_Router 32](#_Toc73541462)

[Hình 3.28: Máy ảo Metasploitable với địa chỉ được cấp phát tự động từ DHCP\_Router 32](#_Toc73541463)

[Hình 3.29: Nhập mật khẩu để bắt đầu cài đặt kiến trúc 33](#_Toc73541464)

[Hình 3.30: Không cần cài đặt mạng vì đã được cài từ trước 33](#_Toc73541465)

[Hình 3.31: Giao diện lựa chọn kiến trúc hệ thống 34](#_Toc73541466)

[Hình 3.32: Cài đặt tài khoản sử dụng cho hệ thống phát hiện xâm nhập 34](#_Toc73541467)

[Hình 3.33: Xác nhận hoàn thiện cấu hình hệ thống 35](#_Toc73541468)

[Hình 3.34: Xác nhận các thông báo hoàn thiện cấu hình hệ thống 35](#_Toc73541469)

[Hình 3.35: Các công cụ của hệ thống phát hiện xâm nhập mạng 35](#_Toc73541470)

[Hình 3.36: Quét tìm các máy trạm của mạng con 36](#_Toc73541471)

[Hình 3.37: Quét tìm các cổng và dịch vụ trên cổng đang mở 36](#_Toc73541472)

[Hình 3.38: Giao diện ứng dụng web tại cổng 80 của mục tiêu 37](#_Toc73541473)

[Hình 3.39: Chọn interface quản lý phát hiện xâm nhập 38](#_Toc73541474)

[Hình 3.40: Các cảnh báo về các hành vi xâm nhập được phát hiện theo thời gian thực 38](#_Toc73541475)

[Hình 3.41: Phát hiện hành vi thăm dò bằng công cụ nmap 39](#_Toc73541476)

[Hình 3.42: Ứng dụng Web Squert giúp phân tích, thống kê hỗ trợ phát hiện xâm nhập 39](#_Toc73541477)

[Hình 3.43: Đăng nhập vào ứng dụng DVWA bằng tài khoản admin 39](#_Toc73541478)

[Hình 3.44: Cài đặt mức bảo mật về low để dễ dàng cho kiểm thử 40](#_Toc73541479)

[Hình 3.45: Ứng dụng web này dễ dàng bị tấn công bởi XSS 40](#_Toc73541480)

[Hình 3.46: Phát hiện tấn công XSS và luật giúp phát hiện 41](#_Toc73541481)

[Hình 3.47: Thực hiện tấn công SQL injection để lấy thông tin từ cơ sở dữ liệu 42](#_Toc73541482)

[Hình 3.48: Tài khoản và mật khẩu thu được có thể mã hóa một cách dễ dàng 42](#_Toc73541483)

[Hình 3.49: Phát hiện tấn công SQL injection 43](#_Toc73541484)

[Hình 3.50: Sử dụng Terminal truy cập quyền root 43](#_Toc73541485)

[Hình 3.51: Chuyển đến thư mục chứa luật 43](#_Toc73541486)

[Hình 3.52: Mở file để thêm luật của bản thân 44](#_Toc73541487)

[Hình 3.53: Luật được cập nhật thành công vào cơ sở dữ liệu 44](#_Toc73541488)

[Hình 3.54: Luật hoạt động tốt khi có tấn công thử nghiệm 45](#_Toc73541489)

[Hình 3.55: Công cụ hỗ trợ tạo luật đơn giản 46](#_Toc73541490)

**DANH SÁCH TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
| NIDS | Network Intrusion Detection System |
| SO | Security Onion |
| IDS | Intrusion Detection System |
| IPS | Intrusion Prevention System |
| IP | Internet Protocol |
| ICMP | Internet Control Message Protocol |
| TCP | Tranmission Control Protocol |
| UDP | User Datagram Protocol |
| URI | Uniform Resource Identifier |
| Nmap | Network Mapper |
| XSS | Cross-Site Scripting |
| HTML | Hyper Text Markup Language |
| SQLi | SQL injection |
| DVWA | Damn Vulnerable Web Application |

**MỞ ĐẦU**

1. **Tổng quan về đề tài**

Security Onion là một bản phân phối mã nguồn mở của Linux với nhiều công cụ hữu dụng có thể phân tích, phát hiện các mối nguy hại trong môi trường mạng dựa vào việc phân tích các gói tin gửi đi trong mạng. Bằng cách cài đặt và cấu hình phù hợp, Security Onion có thể đóng vai trò như một hệ thống phát hiện xâm nhập mạng (NIDS – Network Intrusion Detection System).

1. **Mục đích và ý nghĩa của đề tài**
   1. ***Mục đích***

Môi trường mạng luôn tồn lại nhiều nguy cơ bị tấn công, xâm nhập vì vậy việc cài đặt một hệ thống phát hiện các nguy cơ đó một cách nhanh chóng, dễ dàng và chính xác là điều vô cùng cần thiết.

* 1. ***Ý nghĩa***

Thực hiện các kế hoạch để phát hiện sớm những rủi ro, xâm nhập nhằm bảo vệ mạng lưới mạng khỏi Hacker.

1. **Phương pháp thực hiện**

Thực hiện theo các bước sau:

* Tìm hiểu đề tài và phân tích yêu cầu.
* Xây dựng mô hình hệ thống.
* Triển khai phân tích và phát hiện các cuộc tấn công.

1. **Bố cục của đồ án**

Đồ án bao gồm các nội dung sau:

*Mở đầu*

*Chương 1: Trình bày về cơ sở lý thuyết*

*Chương 2: Trình bày về phân tích và thiết kế hệ thống*

*Chương 3: Trình bày về triển khai phát hiện xâm nhập hệ thống*

*Kết luận và hướng phát triển*

**CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

* 1. **Tổng quan về Security Onion [1]**
     1. ***Giới thiệu***

Security Onion (SO) là một phiên bản của Linux được thiêt kế để phát hiện xâm nhập và giám sát an toàn mạng. Bộ công cụ này dựa trên Xubuntu 10.04, gồm Snort, Suricta, Sguil, Squert, Snorby và nhiều công cụ an toàn khác. Đây là bộ công cụ rất hữu dụng trong giảng dạy và học tập, ngoài ra Security Onion còn được sử dụng cho các văn phòng và mạng lưới cá nhân.

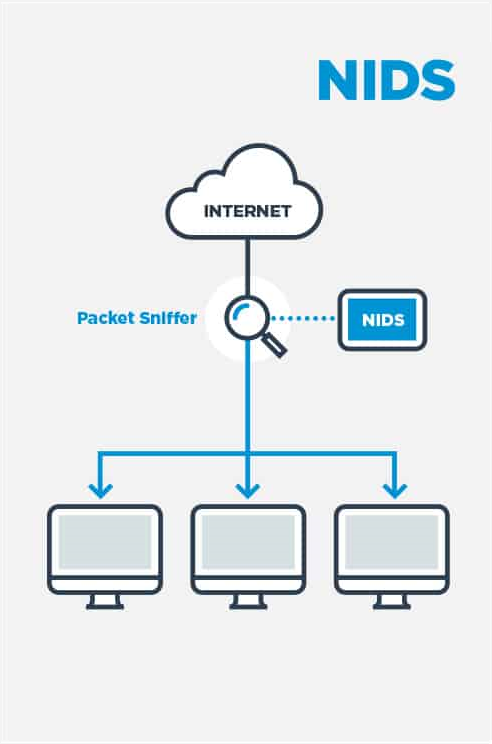
* + 1. ***Chức năng***

Người quản trị có thể sử dụng Security Onion để theo dõi lưu lượng truy cập từ phía bên ngoài trung tâm dữ liệu để phát hiện kẻ xâm nhập một môi trường mạng, thiết lập lệnh và kiểm soát hoặc có thể là xâm nhập dữ liệu. Người quản trị cũng có thể theo dõi lưu lượng truy cập trong một trung tâm dữ liệu để phát hiện các động thái có nguy cơ từ bên trong.

Security Onion có hai chức năng cốt lõi chính:

* Bắt tất cả gói tin (Full Packet Capture): Được thực hiện thông qua công cụ netsniff-ng. Đây là công cụ hỗ trợ phân tích, bắt các gói tin trên môi trường mạng mà không thông qua quá trình sao chép về bộ nhớ người dùng (Zero-copy). Công cụ hỗ trợ định dạng file pcap.
* Phát hiện mạng (Network detection): Phân tích lưu lượng gói tin trong mạng hoặc hệ thống máy chủ, đồng thời cung cấp dữ liệu nhật ký (các bản ghi log) và cảnh báo cho các sự kiện và hoạt động xâm nhập được phát hiện. Security Onion cung cấp hệ thống phát hiện xâm nhập mạng hướng theo quy tắc (Rule-driven NIDS), để phát hiện xâm nhập mạng theo quy tắc, Security Onion sử dụng Snort hoặc Suricata. Hệ thống dựa trên quy tắc phân tích, xem xét lưu lượng mạng để tìm dấu hiệu nhận dạng khớp với lưu lượng độc hại, bất thường hoặc đáng ngờ.

Security Onion được tích hợp sẵn nhiều công cụ giúp hỗ trợ phân tích các gói tin bắt được, tổng hợp, thống kê dữ liệu với giao diện đơn giản, dễ sử dụng.

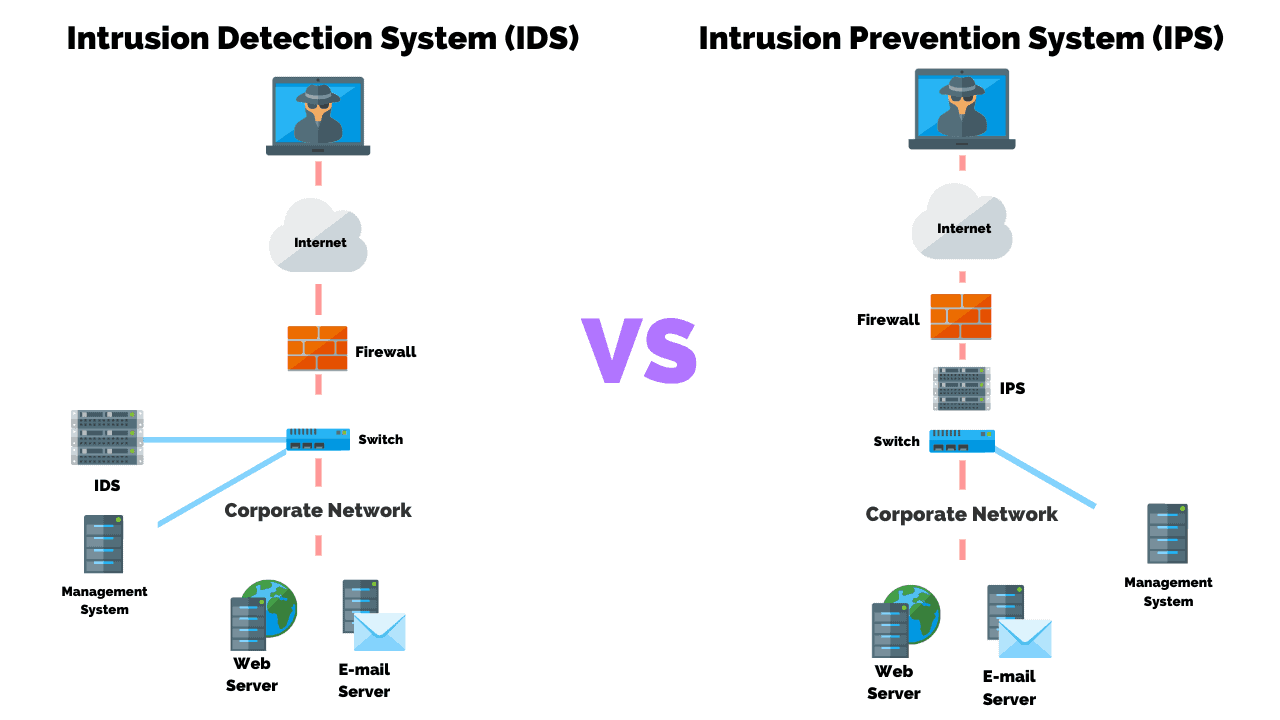


Hình 1.1: Hệ thống phát hiện xâm nhập mạng (NIDS)

* + 1. ***Ưu điểm***
* Là một phiên bản của Linux nên dễ dàng trong việc cài đặt, cấu hình cũng như không yêu cầu nhiều về khả năng phần cứng của máy khi cài đặt.
* Thuận tiện trong cài đặt như là một hệ thống phát hiện xâm nhập trong môi trường mạng.
* Giao diện đơn giản, dễ nhìn hỗ trợ phát hiện thời gian thực cũng như thống kê dữ liệu theo khoảng thời gian cụ thể.
* Bộ luật luôn được cập nhật thường xuyên bởi cộng đồng chia sẻ, bên cạnh đó rất dễ dàng trong việc thiết kế, tạo bộ luật riêng phù hợp với mục đích cũng như môi trường sử dụng.
  + 1. ***Nhược điểm***
* Security Onion chỉ hỗ trợ chức năng hoạt động như một hệ thống phát hiện xâm nhập, không thể ngăn chặn khi có xâm nhập xảy ra.
* Vì phát hiện xâm nhập dựa trên bộ luật có sẵn, phân tích gói tin bằng cách bắt toàn bộ gói khiến thời gian phát hiện đôi khi có độ trễ nhất định.
* Tuy luôn được cộng đồng chia sẻ bổ sung thêm luật mới nhưng vẫn thể phát hiện hoàn toàn mọi loại tấn công, xâm nhập đặc biệt là các hình thức mới chưa có luật phù hợp.
  1. **Công nghệ phát triển**
     1. ***Hệ thống phát hiện xâm nhập mạng***

Hệ thống phát hiện xâm nhập mạng (Network Intrusion Detection System - NIDS) phân tích lưu lượng mạng để phát hiện tấn công, xâm nhập cho cả mạng hoặc một phần của mạng. Trong một sơ đồ mạng, trong đó các NIDS thường được bố trí để giám sát phát hiện xâm nhập tại cổng vào và cho từng phân đoạn của mạng.

Khác với hệ thống ngăn chặn xâm nhập (Intrusion Prevention System - IPS), một hệ thống phát hiện chỉ có thể phát hiện được các hành vi tấn công, xâm nhập bằng cách phân tích các gói tin bắt được trong môi trường mạng, không có chức năng ngăn chặn các cuộc xâm nhập đấy. Chính vì khả năng này khi cài đặt một NIDS trong mạng cá nhân chỉ cần nối máy có NIDS với mạng như một Client thông thường không cần phải cấu hình như một tường lửa đóng vai trò ngăn chặn xâm nhập.



Hình 1.2: Sự khác nhau giữa cách cài đặt IDS và IPS trong môi trường mạng

* + 1. ***Kiến trúc***

Để triển khai Security Onion có rất nhiều kiểu triển khai tùy thuộc vào mục đích sử dụng từ một máy ảo nhỏ hay đến mộ hệ thống doanh nghiệp lớn hơn. Dựa trên mục đích sử dụng người ta chia Security Onion thành các kiểu kiến trúc sau:

* Import: Đây là kiểu kiến trúc đơn giản nhất, Security Onion sẽ được thiết lập chạy vừa đủ các công cụ cần thiết để có thể phân tích gói tin lưu dưới định dạng file pcap.
* Evaluation: Đây là kiểu kiến trúc thích hợp cho thực hiện trên một máy ảo, cấu hình dùng cho các bài lad để giảng dạy hay mô hình mạng cá nhân vừa và nhỏ. Security Onion sẽ tự động cài đặt các công cụ cũng như cấu hình cần thiết giúp dễ dàng cho người mới sư dụng hay với nhu cầu đơn giản.
* Production: Đây là kiến trúc triển khai hoạt động như một sản phẩm dùng trong sản xuất. Security Onion cho phép tùy biến cài đặt cấu hình. Tuy quá trình cài đặt có phần phức tạp hơn kiến trúc Evaluation tuy nhiên nó cho phép người quản trị có thể dễ dàng trong việc quản lý cũng như mở rộng chức năng. Đây là loại kiến trúc thường dùng trong các mạng lớn như công ty, doanh nghiệp.
  + 1. ***Các công cụ NIDS***

Các công cụ NIDS là các công cụ giám sát, phát hiện xâm nhập hệ thống mạng, bắt được các hành động khả nghi cụ thể và đưa ra các cảnh báo dựa vào luật đã được cài đặt sẵn.

Security Onion có thể sử dụng Snort hoặc Suricata làm NIDS engine. Trong quá trình cài đặt, nếu chọn “Evaluation Mode”, Security Onion sẽ mặc định NIDS emgine là Snort. Ngược lại nếu chọn “Produciton Mode” sẽ được chọn tùy biến giữa Snort hoặc Suricata làm NIDS engine.

Snort là một hệ thống phát hiện và ngăn chặn xâm nhập mạng miễn phí và nguồn mở. Nó sử dụng ngôn ngữ dựa trên quy tắc được cài sẵn, thực hiện phân tích giao thức, tìm kiếm kết hợp nội dung và có thể được sử dụng để phát hiện nhiều loại tấn công và thăm dò khác nhau.

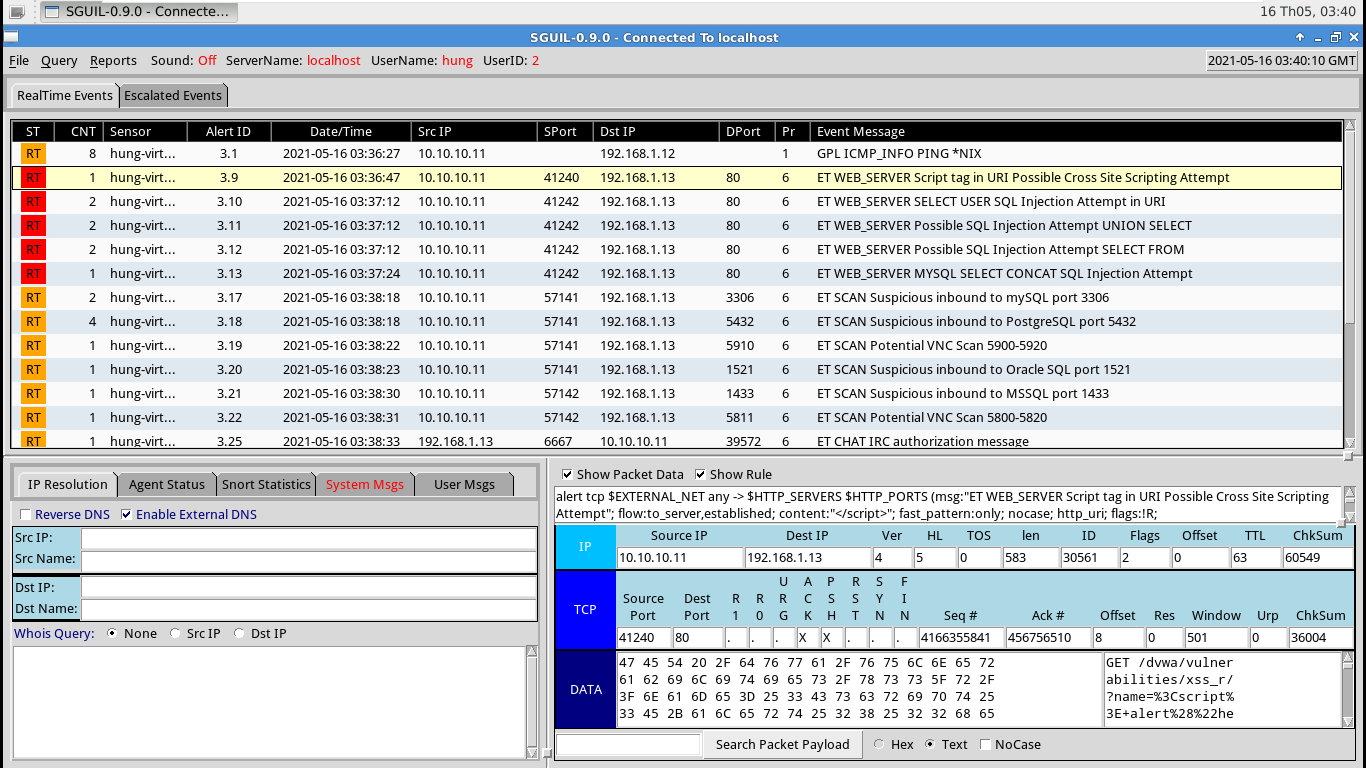
Suricata cũng là một hệ thống phát hiện xâm nhập mạng miễn phí mã nguồn mở như Snort. Cả hai đều có thể sự dụng chung bộ luật nhưng cách cài đặt cấu hình có phần khác nhau. Suricata thường phức tạp hơn bởi được tích hợp thêm nhiều tính năng như: đa luồng, trích xuất file, tăng tốc phần cứng,…

Các cảnh báo của NIDS có thể được phân tích kĩ hơn bằng các công cụ giao diện trực quan như Sguil, Squert.

* + 1. ***Công cụ Sguil***

Sguil là một ứng dụng phân tích/giám sát mạng, có giao diện đơn giản, cung cấp khả năng bắt và truy cập các sự kiện thời gian thực, bắt toàn bộ gói tin và các dữ liệu phiên (session data). Đây là một công cụ quản lý hệ thống mạng và cũng là một công cụ phân tích hướng sự kiện (event driven).

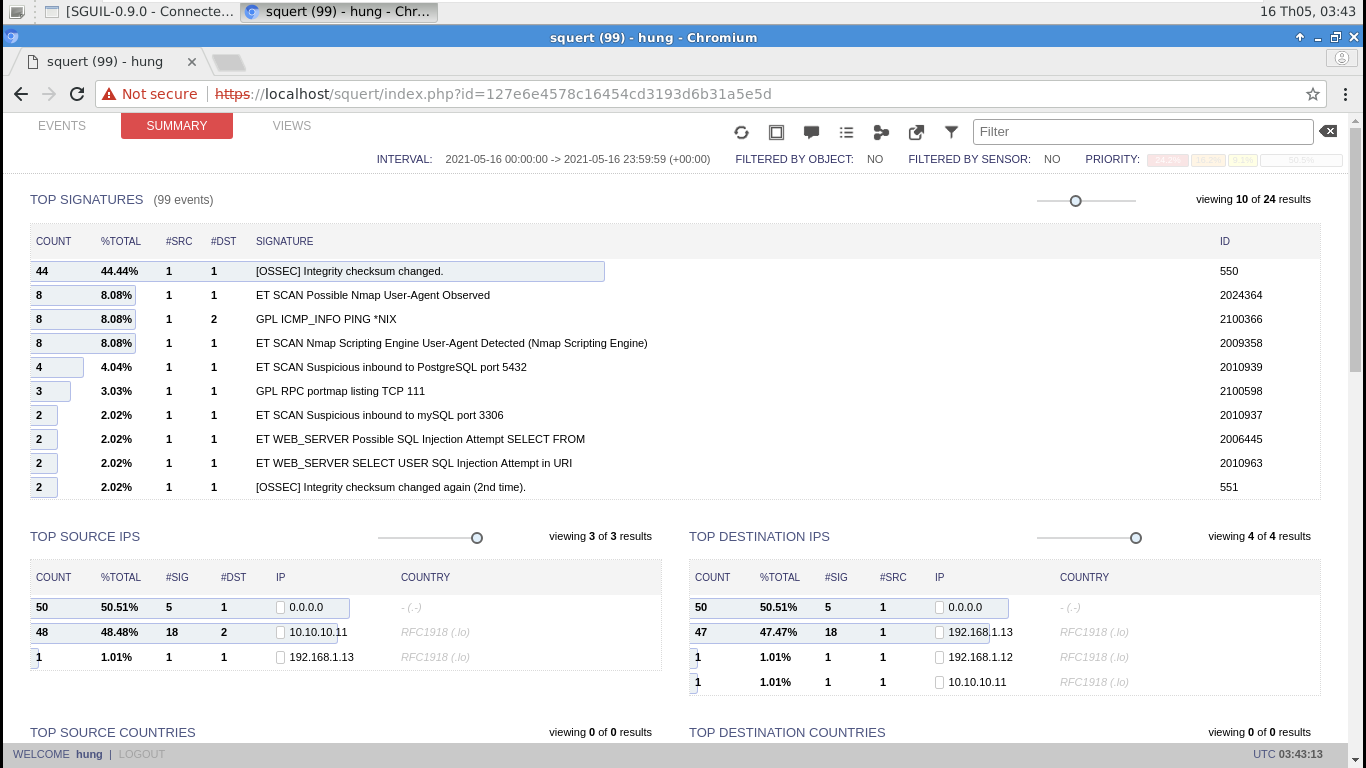
Sguil cho phép hiển thị thông tích phân tích gói tin từ các công cụ NIDS theo thời gian thực, tùy chỉnh truy vấn theo thời gian, mục tiêu hay loại tấn công. Ngoài ra Sguil có tích hợp nối trực tiếp đến công cụ Wireshark khi muốn phân tích kĩ hơn một gói tin bắt được.



Hình 1.3: Giao diện Sguil

* + 1. ***Công cụ Squert***

Squert là một ứng dụng web được sử dụng để thực hiện các lệnh truy vấn và xem các sự kiện từ log thu thập được trong cơ sở dữ liệu của Sguil (các cảnh báo nhận được từ NIDS), dữ liệu sẽ được thống kê theo biểu đồ trực quan giúp dễ dàng trong việc theo dõi xâm nhập.

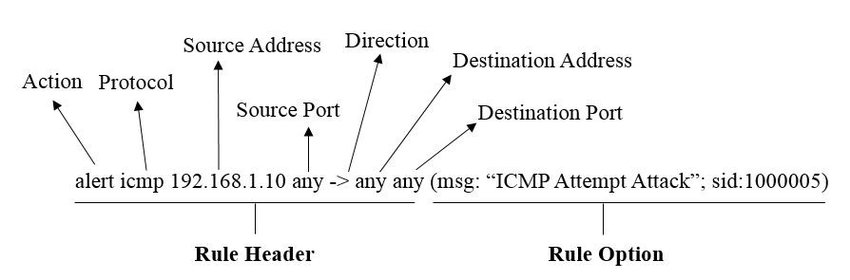


Hình 1.4: Giao diện Squert

* + 1. ***Bộ luật của Security Onion***

Trong quá trình cài đặt Security Onion có thể lựa chọn Snort hoặc Suricata làm hệ thống phát hiện xâm nhập mạng (NIDS). Tuy mỗi công cụ có cách cấu hình cũng như vận hành khác nhau nhưng may mắn thay cả hai đều có thể dùng chung cách cài đặt luật cơ bản dùng cho Security Onion.

Bộ luật cơ bản của Security Onion có thể dùng chung cho cả Snort hay Suricata bao gồm 2 phần chính: Rule header và rule Options



Hình 1.5: Cấu trúc cách viết luật trong Security Onion

* Rule Header: luôn là phần đầu tiên của rule bao gồm các thành phần:
* Actions: Việc xử lý đối với dữ liệu được phân tích, vì Security Onion chỉ tích hợp IDS nên actions sử dụng chủ yếu là **alert**.
* Protocol: Giao thức sử dụng của gói tin bao gồm: **tcp, udp, icmp** và **ip**.
* IP address: IP đích và nguồn của gói tin, nếu không biết rõ mục tiêu tấn công hay cần bảo vệ có thể để **any**.
* Port: Cổng địa chỉ nguồn và đích của gói tin, tương tự như địa chỉ IP khi viết luật ta có thể để **any** để thông báo áp dụng với mọi cổng.
* Direction: Toán tử “**->”** chỉ ra đâu là hướng nguồn, đâu là hướng đích. Ngoài ra còn có toán tử “<>” sẽ xem cặp địa chỉ IP/port nguồn đích là như nhau.
* Rule Options: Đây chính là trọng tâm của việc phát hiện xâm nhập. Nội dung chứa các dấu hiệu để xác định xâm nhập. Nó nằm sau phần Rule Header và được bọc bởi cặp dấu ngoặc đơn “()”, tất cả các rule options được phân cách nhau bởi dấu chấm phẩy “;”, phần đối số sẽ được tách bởi dấu hai chấm “:”. Rule options có nhiều lựa chọn và không bắt buộc phải đủ các thành phần mới hoạt động được. Bao gồm các thành phần cơ bản sau:
* Message (msg): Được dùng để cho biết thông tin về từng signature và các cảnh báo tương ứng. Đinh dạng của msg như sau: **msg:“………”;**
* Signature ID (sid): Cho ta biết định danh riêng của mỗi signature. Định danh được bắt đầu với số, đối với luật tự tạo local số được đếm từ 1000000 trở đi. Định dạng của sid như sau: **sid:1234567;**
* Revision (rev): Mỗi sid thường đi kèm với một rev đại diện cho các phiên bản của signature. Mỗi khi signature được sửa đổi thì số rev sẽ tăng lên bởi người tạo luật. Định dạng của rev như sau: **rev:1;**
* Classtype: Cung cấp thông tin về việc phân loại các lớp quy tắc và cảnh báo. Mỗi lớp là một tên ngắn gọn, có định dạng như sau: **classtype:test;**
* Priority: Chỉ ra mức độ ưu tiên của signature. Các giá trị ưu tiên dao động từ 1->255 nhưng thường sử dụng các giá trị từ 1->5. Mức ưu tiên cac nhất là 1, những mức ưu tiên cao hơn sẽ được kiểm tra trước. Định dạng của mức chỉ ưu tiên như sau: **priority:1;**
* 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, bao gồm các ký tự thường, hoa, đặc biệt hay các mã hexa tương ứng. Định dạng của nội dung như sau: **content:“…..”;** (Lưu ý các kí tự đặc biệt như nháy đơn, nháy kép, hai chấm,… phải chuyển về mã hexa)
* Offset: Là một thuộc tính của Content, chỉ độ lệch byte trong quá trình kiểm tra. Ví dụ **offset=3** thì sẽ kiểm tra từ byte thứ 4 trong gói tin. Vì là thuộc tính của Content nên bắt buôc phải đặt sau **content**, có định dạng như sau: **content:“hello world!”; offset:5;**
* Depth: Là một thuộc tính của Content, chỉ ra số lượng byte tối đa cần được kiểm tra. Ví dụ **depth=3** thì sẽ kiểm tra 3 byte đầu của gói tin. Vì là thuộc tính của Content nên bắt buôc phải đặt sau **content**, có định dạng như sau: **content:“hello world!”; depth:12;**
* Nocase: Là một thuộc tính của Content, thông báo kiểm tra gói tin chứa nội dung không phân biệt chữ hoa, chữ thường. Vì là thuộc tính của Content nên bắt buôc phải đặt sau **content**, có định dạng như sau: **content:“Hello World!”; nocase;**
* URI: Là một thuộc tính của Content, thông báo chuẩn hóa các gói tin chứa đường dẫn http đã được mã hóa. Vì là thuộc tính của Content nên bắt buôc phải đặt sau **content**, có định dạng như sau: **content:“/test page”; http\_uri;**
* Not: Là một thuộc tính của Content, thông báo khi phát hiện các gói tin khi không chứa nột dung trong phần Content. Thuộc tính not được biểu hiện bởi dấu chấm than trước mục nội dung của Content, có định dạng như sau: **content:!“hello”;**
  1. **Các kiểu xâm nhập cơ bản**
     1. ***Thăm dò thông tin sử dụng Nmap [2]***

Nmap (tên đầy đủ Network Mapper) là một công cụ bảo mật được phát triển bởi Floydor Vaskovitch. Nmap có mã nguồn mở, miễn phí, dùng để quét cổng và lỗ hổng bảo mật. Các chuyên gia quản trị mạng sử dụng Nmap để xác định xem thiết bị nào đang chạy trên hệ thống của họ, cũng như tìm kiếm ra các máy chủ có sẵn và các dịch vụ mà các máy chủ này cung cấp, đồng thời dò tìm các cổng mở và phát hiện các nguy cơ về bảo mật.

Nmap có thể được sử dụng để giám sát các máy chủ đơn lẻ cũng như các cụm mạng lớn bao gồm hàng trăm nghìn thiết bị và nhiều mạng con hợp thành.

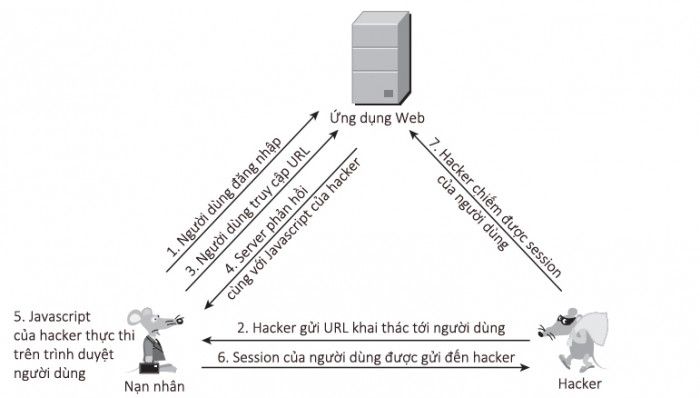
Một trong những lý do dẫn đến sự phổ biến rộng rãi của Nmap là nó có thể được sử dụng được trên rất nhiều hệ điều hành khác nhau. Nó chạy được trên Windows và macOS cũng như được hỗ trợ trên các bản phân phối của Linux bao gồm Red Hat, Mandrake, SUSE và Fedora. Nó cũng hoạt động tốt trên các hệ điều hành khác bao gồm BSD, Solaris, AIX và AmigaOS.

* + 1. ***Tấn công Cross-Site Scripting (XSS)***

Cross Site Scripting (XSS) là một trong những tấn công phổ biến và dễ bị tấn công nhất mà tất cả các Tester có kinh nghiệm đều biết đến. Nó được coi là một trong những tấn công nguy hiểm nhất đối với các ứng dụng web và có thể mang lại những hậu quả nghiêm trọng. Để khai thác một lỗ hổng XSS, hacker sẽ chèn mã độc thông qua các đoạn script để thực thi chúng ở phía Client. Thông thường, các cuộc tấn công XSS được sử dụng để vượt qua truy cập và mạo danh người dùng.

Mục đích chính của cuộc tấn công này là ăn cắp dữ liệu nhận dạng của người dùng như: cookies, session tokens và các thông tin khác. Trong hầu hết các trường hợp, cuộc tấn công này đang được sử dụng để ăn cắp cookie của người khác. Như chúng ta biết, cookie giúp người dùng đăng nhập tự động. Do đó với cookie bị đánh cắp, hacker có thể đăng nhập bằng các thông tin nhận dạng khác. Và đây là một trong những lý do, tại sao cuộc tấn công này được coi là một trong những cuộc tấn công nguy hiểm nhất.

Tấn công XSS đang được thực hiện ở phía client. Nó có thể được thực hiện với các ngôn ngữ lập trình phía client khác nhau. Tuy nhiên, thường xuyên nhất cuộc tấn công này được thực hiện với Javascript và HTML.

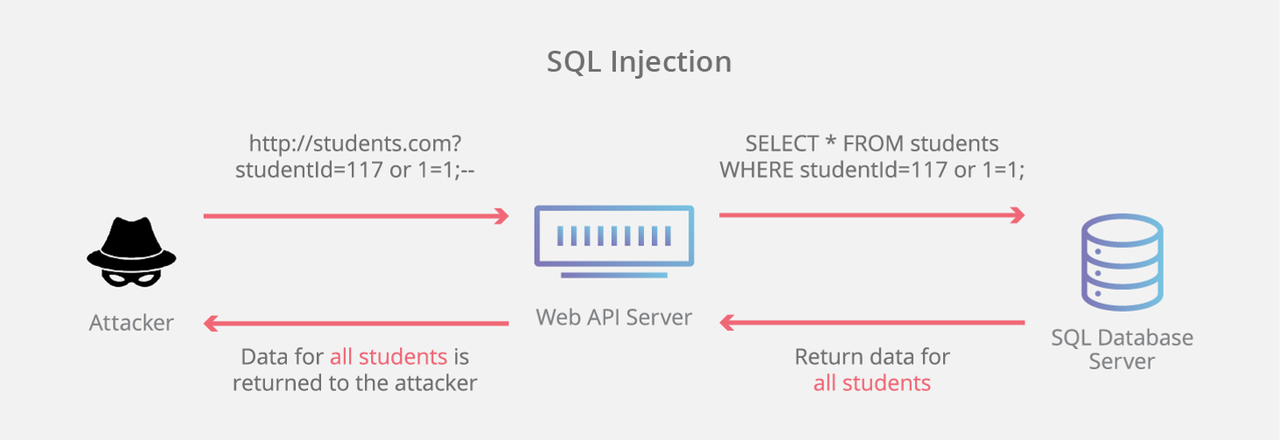


Hình 1.6: Mô tả quá trình thực hiện tấn công XSS

* + 1. ***Tấn công SQL injection***

SQL injection – còn được gọi là SQLi – sử dụng những lỗ hổng trong các kênh đầu vào (input) của website để nhắm mục tiêu vào cơ sở dữ liệu nằm trong phần phụ trợ của ứng dụng web, nơi lưu giữ những thông tin nhạy cảm và có giá trị nhất. Chúng có thể được kẻ tấn công sử dụng để ăn cắp hoặc xáo trộn dữ liệu, cản trở sự hoạt động của các ứng dụng, và, trong trường hợp xấu nhất, nó có thể chiếm được quyền truy cập quản trị vào máy chủ cơ sở dữ liệu.

Các cuộc tấn công SQL Injection được thực hiện bằng cách gửi lệnh SQL độc hại đến các máy chủ cơ sở dữ liệu thông qua các yêu cầu của người dùng mà website cho phép. Bất kỳ kênh input nào cũng có thể được sử dụng để gửi các lệnh độc hại, bao gồm các thẻ <input>, chuỗi truy vấn (query strings), cookie và tệp tin.

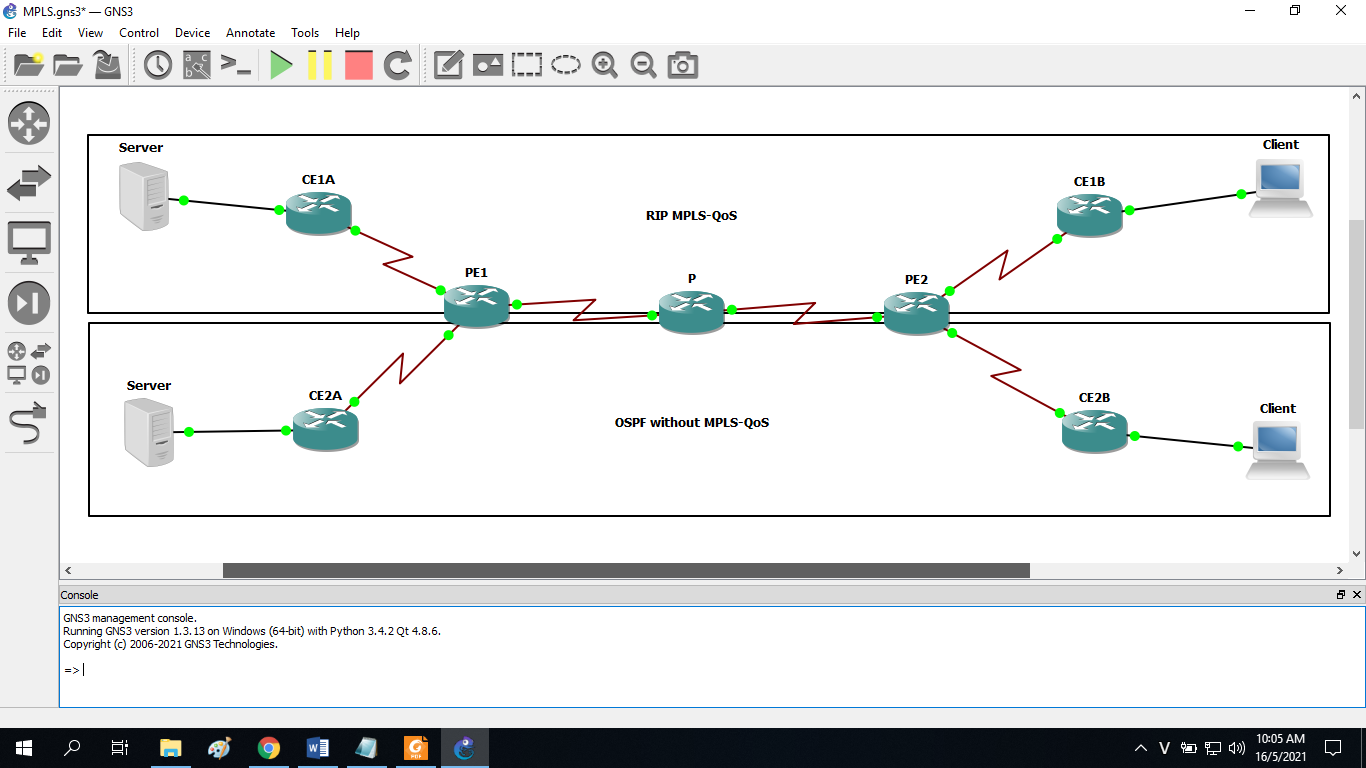


Hình 1.7: Mô tả quá trình thực hiện tấn công SQL injection

* 1. **Mô hình hệ thống phát hiện xâm nhập mạng**
     1. ***Chương trình mô phỏng GNS3 [3]***

GNS3 là 1 chương trình giả lập mạng có giao diện đồ họa cho phép bạn có thể giả lập các loại router Cisco sử dụng IOS (hệ điều hành của router) thật ,ngoài ra còn có thể giả lập các thiết bị mạng khác như ATM, Frame Relay, Ethernet Switch ,Pix Firewall… và đặt biệt có thể kết nối vào hệ thống mạng thật và sử dụng như thiết bị thật.

GNS3 giúp xây dựng, thiết kế và kiểm tra mạng của bạn trong một môi trường ảo không có rủi ro thật và tiếp cận các cộng đồng mạng lớn nhất để trao đổi giúp đỡ cùng nhau. GNS3 là một giải pháp dễ dàng để thiết kế và xây dựng mạng lưới của bất kỳ kích thước nào mà không cần lo ngại về phần cứng và chi phí. Và đặc biệt là nó hoàn toàn miễn phí.



Hình 1.8: Giao diện ứng dụng GNS3

* + 1. ***Hệ điều hành Kali Linux [4]***

Kali Linux được phát triển và duy trì bởi Offensive Security. Họ được biết đến như một ông lớn trong lĩnh vực và cực kỳ đáng tin cậy trong thế giới bảo mật. Thậm chí họ còn là đơn vị chứng nhận cho các chuyên gia bảo mật với một số chứng chỉ cao cấp nhất hiện nay như OSCP, OSCE, OSWP, OSEE.

Kali Linux được xem như là một giải pháp cực kỳ tiện lợi cho những người đang học tập, nghiên cứu và làm việc trong lĩnh vực bảo mật. Tất cả các phần mềm hữu ích mà bạn cần đều có sẵn trên Kali, nên bạn hoàn toàn có thể sử dụng để kiểm thử bất cứ hệ thống nào.

Kali là một bản phân phối Linux. Cũng giống như các bản phân phối khác, bạn có thể cài đặt vĩnh viễn trên một ổ đĩa cứng. Cài đặt được trên bất kỳ dòng máy máy laptop, pc, máy ảo,… Và với giao diện đồ họa GNOME được tích hợp sẵn, cùng hiệu suất ổn định nên được rất nhiều người sử dụng ưa chuộng.

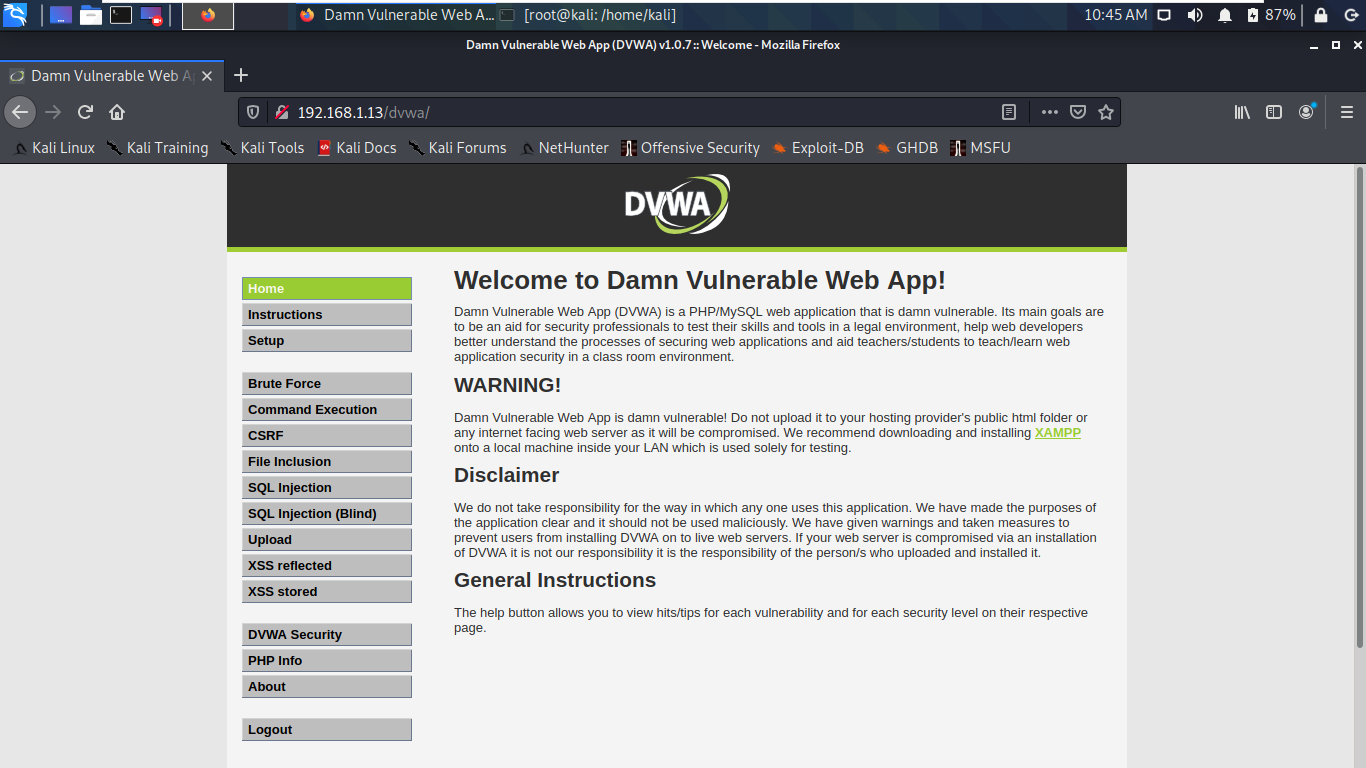


Hình 1.9: Giao diện của hệ điều hành Kali Linux

* + 1. ***Ứng dụng web Damn vulnerable web application và Metasploitable***

Damn Vulnerable Web Application (DVWA) là một ứng dụng mã nguồn PHP/MySQL tập hợp sẵn các lỗi logic về bảo mật ứng dụng web trong mã nguồn PHP. Lỗi logic khi lập trình có thể áp dụng đối với các loại ngôn ngữ lập trình nhằm giảm thiểu khả năng tạo ra lổ hổng bảo mật từ tư duy lập trình chưa cẩn thận [5].

Mục tiêu chính của DVWA đó là tạo ra một môi trường thực hành hacking/pentest hợp pháp. Giúp cho các nhà phát triển ứng dụng web hiểu hơn về hoạt động lập trình an toàn và bảo mật hơn. Bên cạnh đó DVWA cũng cung cấp cho các thầy cô/học sinh phương pháp học và thực hành tấn công khai thác lỗi bảo mật ứng dụng web ở mức cơ bản và nâng cao.



Hình 1.10: Giao diện ứng dụng web Damn Vulnerable Web Application (DVWA)

Metasploitable là một hệ điều hành được thiết kế với mục tiêu đơn giản là bị vi phạm và nhờ vào việc này có thể thực hiện hàng trăm thử nghiệm thâm nhập để cải thiện bảo mật và ngăn chặn các cuộc tấn công trong ngắn hạn, trung hạn hoặc dài hạn.

Đây là một hệ thống Linux được phát triển với các lỗ hổng bảo mật cao nhất, trong đó chúng ta có thể thực hiện tất cả các thử nghiệm cần thiết để cải thiện các kỹ thuật bảo mật hoặc biết cách tận dụng tối đa các ứng dụng được thiết kế cho mục đích này.

Metasploitable tồn tại nhiều lỗ hổng bảo mật cũng như các công cụ hỗ trợ cho quá trình kiểm thử. Ứng dụng web DVWA là một trong những công cụ được tích hợp sẵn bên trong Metasploitable khi cài đặt giúp dễ dàng trong quá trình kiểm thử các lỗ hổng bảo mật về web, cơ sở dữ liệu trên ứng dụng này.



Hình 1.11: Giao diện máy ảo Metasploitable

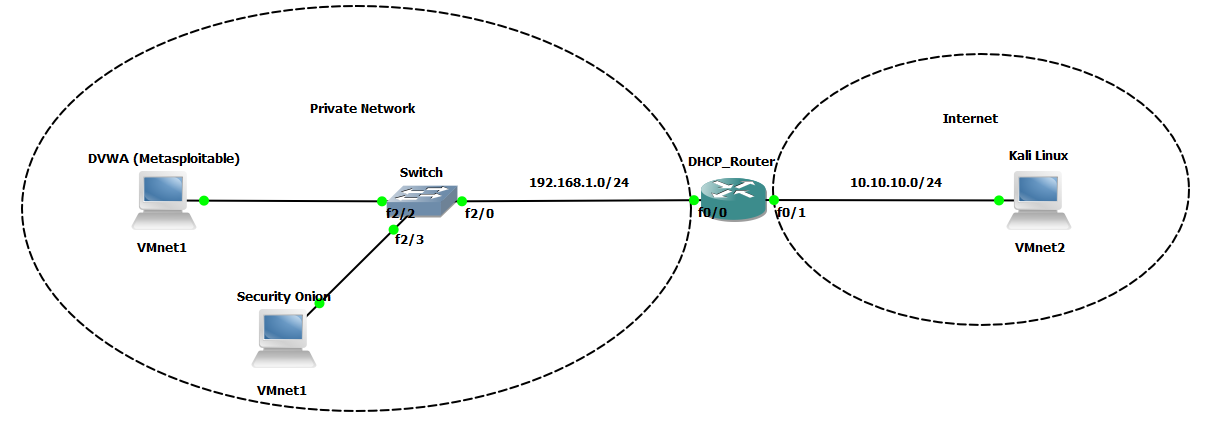
**Kết chương**

Chương này em đã tìm hiểu được về cơ sở lý thuyết hệ thống phát hiện xâm nhập mạng, các công cụ phát triển cũng như cách thức hoạt động, bộ luật của Security Onion và một vài hình thức thăm dò, tấn công cơ bản cùng với các máy ảo hỗ trợ kiểm thử.

**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

* 1. **Phân tích yêu cầu**
     1. ***Mục tiêu đề tài***
* Nghiên cứu hệ thống phát hiện xâm nhập mạng Security Onion.
* Thiết kế hệ thống sử dụng Kali Linux tấn công Metasploitable.
* Nghiên cứu phát hiện xâm nhập với các công cụ của Security Onion.
* Nghiên cứu cách thiết kế luật, có thể tự viết luật hoặc tạo công cụ hỗ trợ tạo luật một cách đơn giản dễ dàng hơn.
  1. **Thiết kế hệ thống**
     1. ***Sơ đồ kiến trúc hệ thống***

Xây dựng hệ thống mạng trên GNS3



Hình 2.1: Sơ đồ hệ thống tấn công

* + 1. ***Phân tích hệ thống***

Yêu cầu đặt ra: Xây dựng hệ thống tấn công bao gồm hai vùng là Internet và Private Network:

* Internet: Mô phỏng môi trường hoạt động của kẻ tấn công (Kali Linux). Có địa chỉ mạng con sẽ được cài đặt là 10.10.10.0/24.
* Private Network: Mô phỏng một mạng cá nhân, là môi trường chứa các máy cần bảo vệ (Metasploitable) của người dùng cũng như chứa hệ thống phát hiện xâm nhập mạng Security Onion. Có địa chỉ mạng con sẽ được cài đặt là 192.168.1.0/24.

Việc cài đặt DHCP service cho Router Cisco vừa giúp tiết kiệm phần cứng khi không phải cài đặt máy ảo riêng làm nhiệm vụ cấp phát IP động, vừa phù hợp cho việc mô phỏng một mạng cá nhân khi các máy Client trong mạng được cấp phát IP động thay cho việc cài IP tĩnh.

Từ bên ngoài Internet, Kali linux sẽ thực hiện các hoạt động thăm dò cũng như xâm nhập đến các mục tiêu chứa nhiều lỗ hổng như Metasploitable phía bên trong mạng cá nhân Private Network.

Security Onion sẽ được cài đặt trong mạng cá nhân với vai trò như một hệ thống giúp phát hiện các các hoạt động xâm nhập mạng có thể xảy ra ngay bên trong mạng con. Các phát hiện xâm nhập sẽ được gửi về cảnh báo theo thời gian thực ngay trên giao diện công cụ Sguil, được tổng hợp và thống kê bằng công cụ Squert.

Để tăng tính hiệu quả của việc phát hiện xâm nhập, chúng ta có thể tự tạo riêng bộ luật giúp hỗ trợ phát hiện nhiều loại tấn công khác nhau. Việc thiết kế luật có thể làm thủ công như viết theo format của bộ luật hay sử dụng công cụ hỗ trợ tạo luật em tự viết sẽ được giới thiệu ở chương tiếp theo.

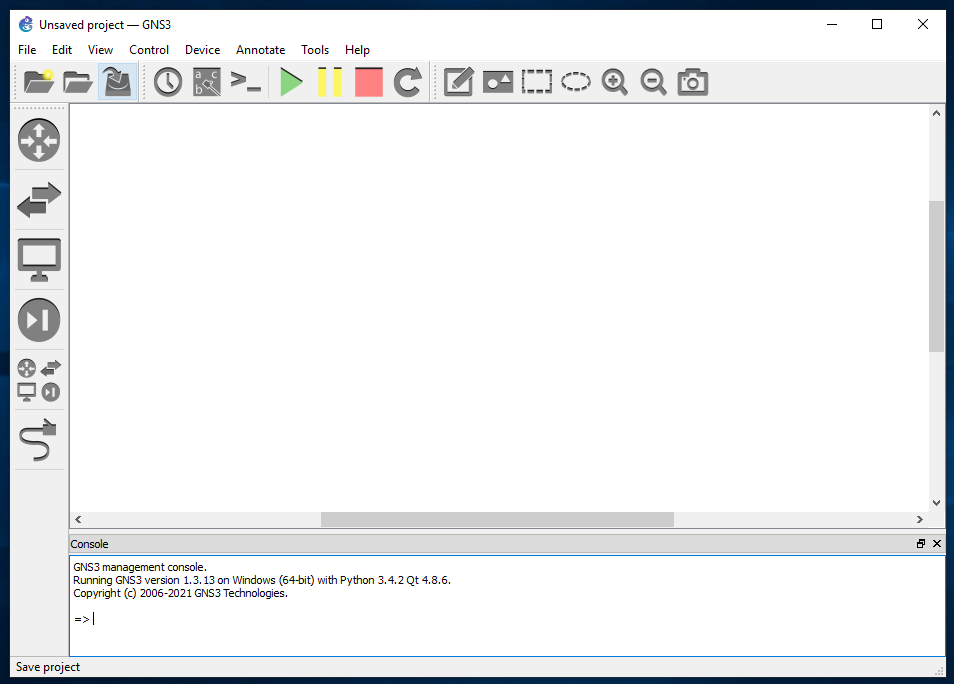
**Kết chương**

Trong chương này em đã tiến hành phân tích các yêu cầu của hệ thống cũng như thiết kế các chức năng của một hệ thống tấn công. Từ đó có thể giúp triển khai xây dựng và cài đặt một hệ thống mô phỏng tấn công và phát hiện xâm nhập mạng trong chương tiếp theo.

**CHƯƠNG 3: XÂY DỰNG VÀ MÔ PHỎNG HỆ THỐNG**

* 1. **Xây dựng sơ đồ mạng bằng GNS3**
     1. ***Cài đặt GNS3 và Router Cisco***

Hiện nay có nhiều phiên bản GNS3 từ cũ đến mới nhất, thực hiện trong đồ án này em lựa chọn phiên bản 1.3.13 vì dễ dàng cài đặt, thân thiện với người sử dụng cũng như ổn định với nhiều loại máy khác nhau.

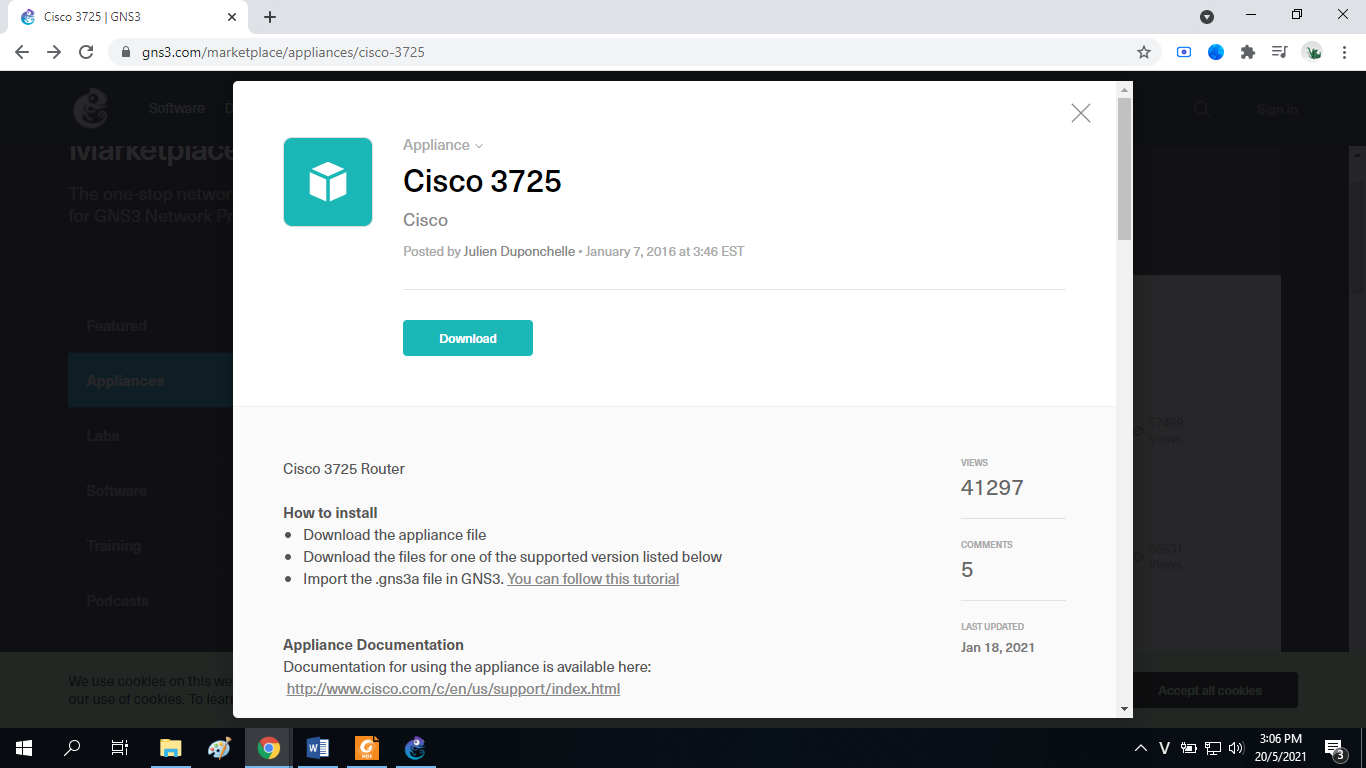


Hình 3.1: Giao diện ứng dụng GNS3 phiên bản 1.3.13

Tương sự, em sử dụng router ảo của Cisco phiên bản C3725, dưới đây là cách cài đặt router ảo trên GNS3 sử dụng file image của Cisco cung cấp:

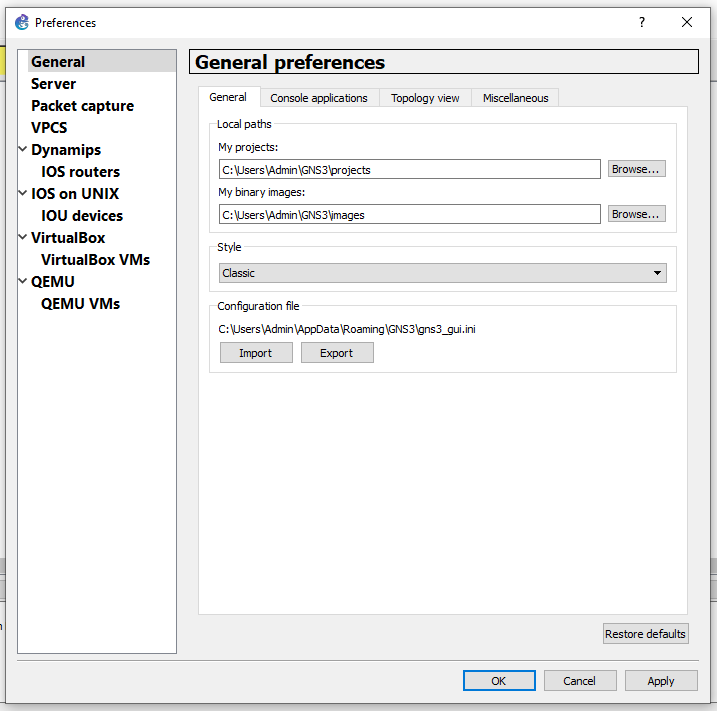
* Bước 1: Tải file image của router Cisco dành cho GNS3 lại địa chỉ sau:

[*https://gns3.com/marketplace/appliances/cisco-3725*](https://gns3.com/marketplace/appliances/cisco-3725%20)



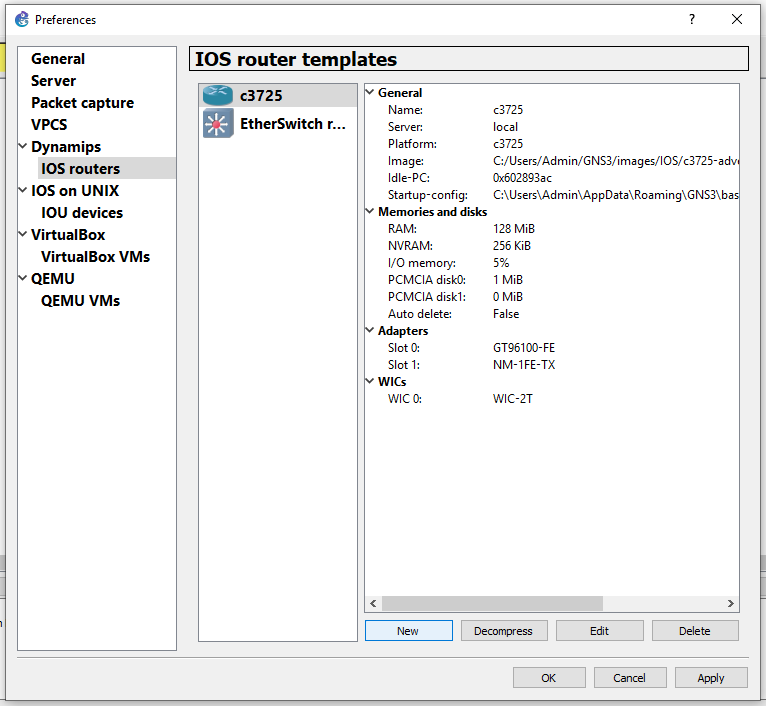
Hình 3.2: Tải file image của Router để cài đặt trên GNS3

* Bước 2: Mở GNS3, vào Edit -> Preferences (Ctrl+Shift+P) để mở cửa sổ cài đặt router ảo.



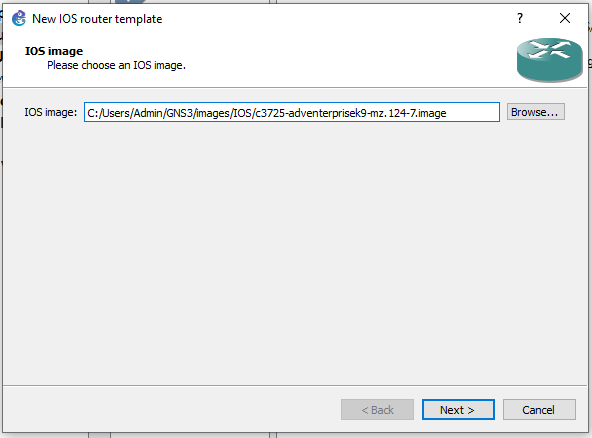
Hình 3.3: Giao diện Preferences của GNS3

* Bước 3: Chọn IOS routers -> New để bắt đầu thêm router ảo vào GNS3.



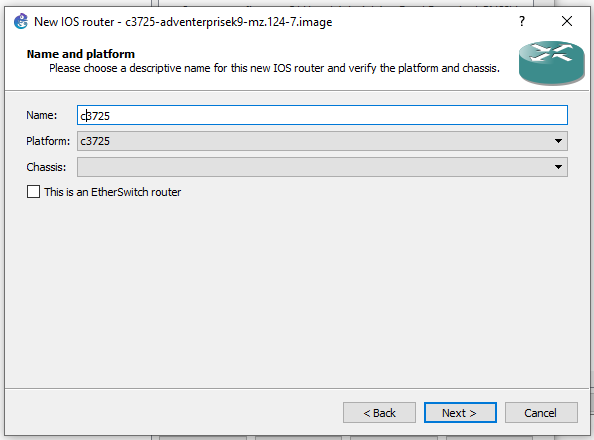
Hình 3.4: Giao diện thêm router ảo của GNS3

* Bước 4: Chọn Browse rồi tìm đường dẫn đến file image vừa tải, bấm Next để tiếp tục.



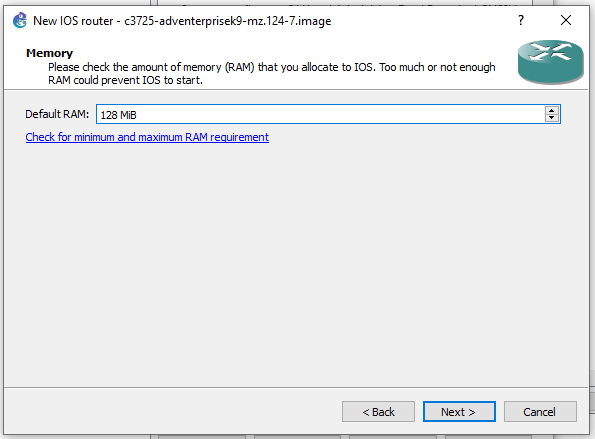
Hình 3.5: Chọn đường dẫn đến file image của router ảo

* Bước 5: Đặt tên cho router theo mong muốn, có thể giữ nguyên không đổi, bấm Next để tiếp tục.



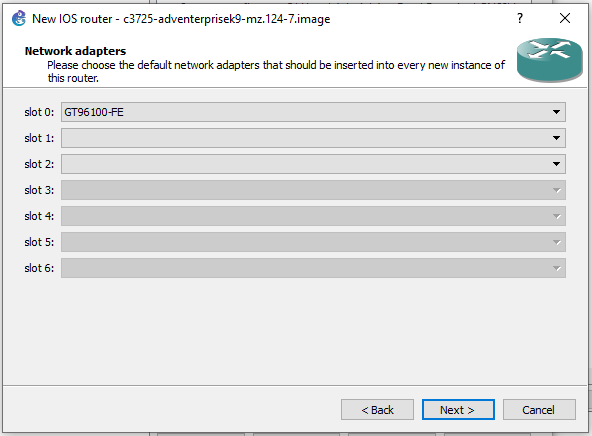
Hình 3.6: Cài tên cho router ảo

* Bước 6: Để mặc định số lượng bộ nhớ RAM cho router, bấm Next để tiếp tục.



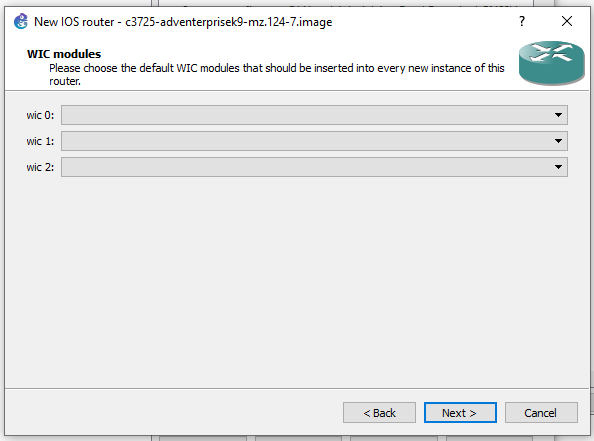
Hình 3.7: Cài bộ nhớ RAM cho router ảo

* Bước 7: Thêm cổng cho router, với sơ đồ đã thiết kế cho hệ thống mô phỏng này ta không cần phải thêm cổng nữa, bấm Next để tiếp tục.



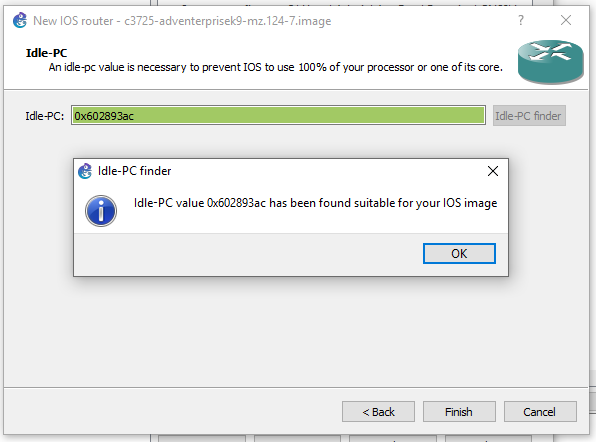
Hình 3.8: Thêm cổng nếu muốn cho router ảo

* Bước 8: Thêm cổng Serial cho router, tương tự ta không cần thêm và bấm next để tiếp tục.



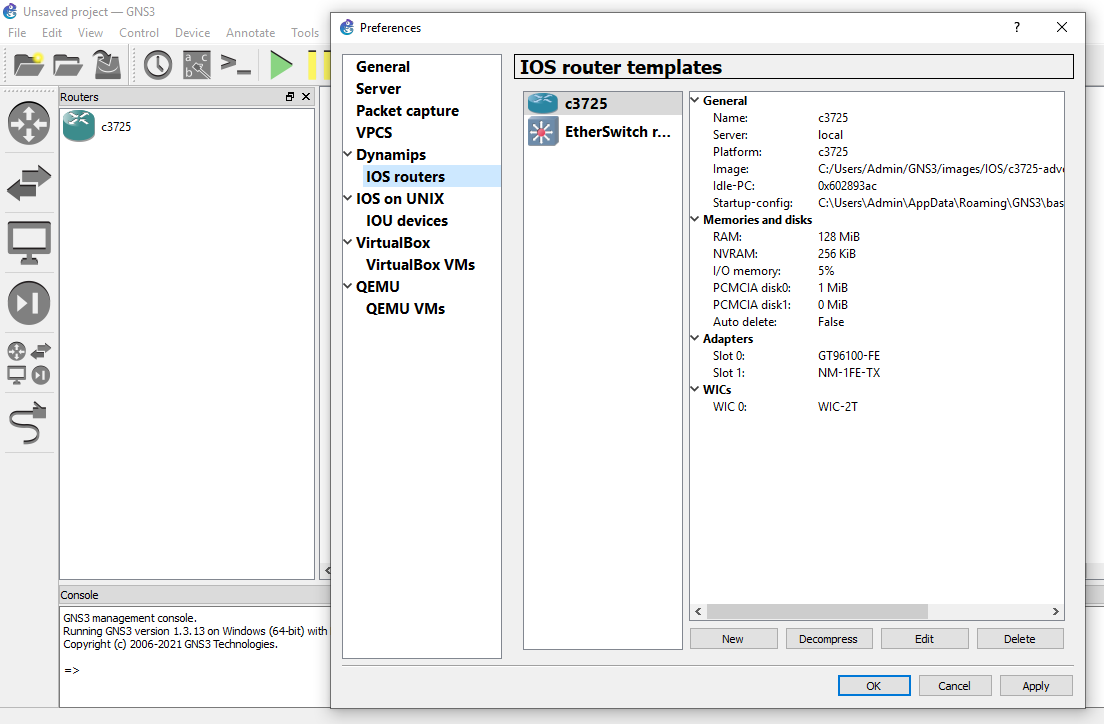
Hình 3.9: Thêm cổng Serial nếu muốn cho router ảo

* Bước 9: Bấm chọn Idle-PC-finder để máy tự định cấu hình phù hợp cho router, sẽ tốn một khoản thời gian nhất định và khi hoàn thành nhấn Finish để kết thúc thêm router ảo Cisco c3725 vào GNS3.



Hình 3.10: Đợi máy tự cấu hình phù hợp cho router ảo

Giờ đây ta đã có một router ảo hoạt động như một router Cisco thật trong môi trường mạng ảo của GNS3.



Hình 3.11: Router Cisco c3725 ảo sau khi được cài đặt vào GNS3

* + 1. ***Cấu hình Router Cisco***

Sau khi cài đặt router ảo trên GNS3, ta cài đặt mô hình sơ đồ mạng như đã phân tích trong Chương 2. Mạng Internet chứa Kali Linux sử dụng mạng ảo VMnet2 của VMware, mạng Private Network sử dụng mạng ảo VMnet1 của VMware cho cả Metasploitable cũng như Security Onion.

Sau khi cài đặt mô hình xong ta thực hiện cấu hình router DHCP làm nhiệm vụ cấp phát IP động cho các máy trạm cũng như lưu thông mạng giữa mạng cá nhân Private Network và Internet.

* Bước 1: Cài đặt địa chỉ IP cho hai cổng default gateway của lần lượt hai mạng nối với router.

***conf t***

***in f0/0***

***no shut***

***ip address 192.168.1.1 255.255.255.0***

***duplex full***

***exit***

***in f0/1***

***no shut***

***ip address 10.10.10.1 255.255.255.0***

***duplex full***

***exit***

* Bước 2: Cài đặt dịch vụ DHCP cho mạng bên trong cũng như ngoài Internet.

***service dhcp***

***ip dhcp excluded-address 192.168.1.1 192.168.1.10***

***ip dhcp pool DHCP***

***network 192.168.1.0 255.255.255.0***

***default-router 192.168.1.1***

***dns-server 192.168.1.1***

***exit***

***ip dhcp excluded-address 10.10.10.1 10.10.10.10***

***ip dhcp pool OUTSIDE***

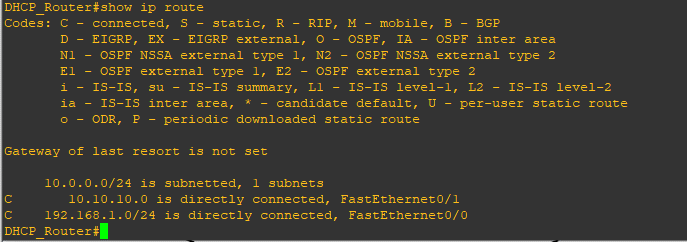
***network 10.10.10.0 255.255.255.0***

***default-router 10.10.10.1***

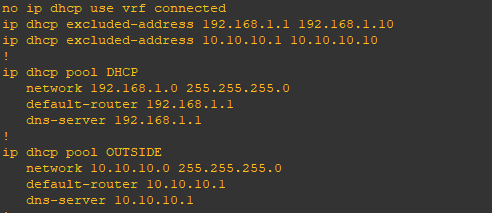
***dns-server 10.10.10.1***

***end***

Sử dụng các lệnh ***show ip route*** và ***show run*** để kiểm tra kết quả cài đặt router.



Hình 3.12: Bảng định tuyến của router



Hình 3.13: Bảng phân chia DHCP của router

* 1. **Xây dựng mô hình hệ thống phát hiện xâm nhập mạng**
     1. ***Cài đặt Security Onion***

Dễ dàng tải các file .iso của Security Onion với nhiều phiên bản phù hợp tại địa chỉ: [*https://github.com/Security-Onion-Solutions/security-onion*](https://github.com/Security-Onion-Solutions/security-onion)

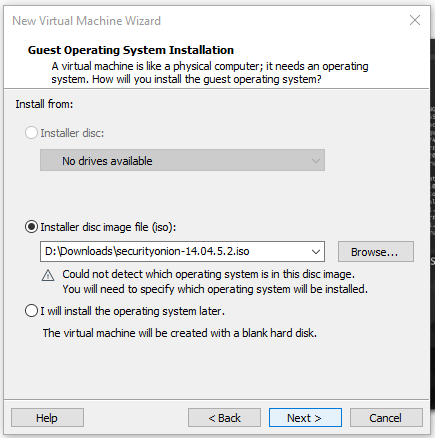
Sau khi tải về máy ta tiến hành cài đặt Security Onion trên VMware (phiên bản sử dụng là bản v15.5 Pro) theo các bước sau:

* Bước 1: Mở VMware, chọn File -> New Virtual Machine để mở cửa sổ tạo mới máy ảo.
* Bước 2: Chọn Typical installation -> Next để chọn chế độ cài đặt dễ dàng.



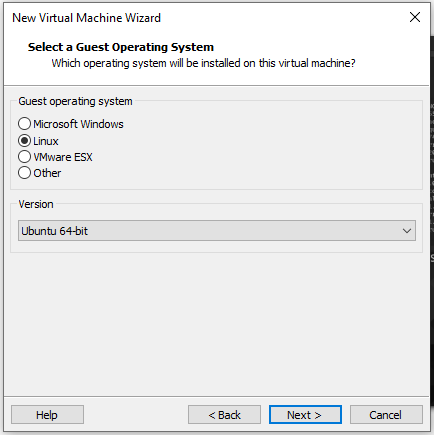
Hình 3.14: Cửa sổ tạo máy ảo của VMware

* Bước 3: Chọn đường dẫn đến file .iso vừa tải được trong mục Installer disc image file, bấm Next để tiếp tục.



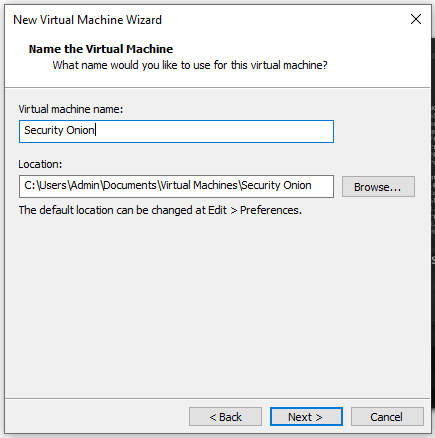
Hình 3.15: Chọn đường dẫn đến file iso của máy ảo

* Bước 4: Chọn Linux, Ubuntu 64-Bit rồi bấm Next để tiếp tục.



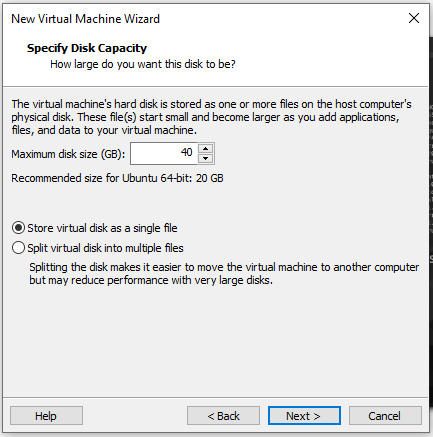
Hình 3.16: Chọn hệ điều hành cho máy ảo

* Bước 5: Cài đặt tên cho máy ảo rồi bấm Next để tiếp tục.



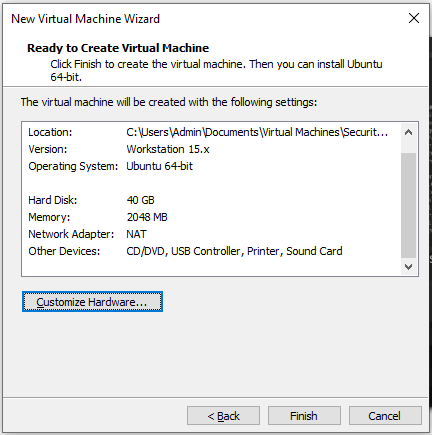
Hình 3.17: Đặt tên cho máy ảo

* Bước 6: Cài đặt bộ nhớ cho máy ảo (tối thiểu 40GB), chọn Store virtual disk as a single file rồi bấm Next để tiếp tục.



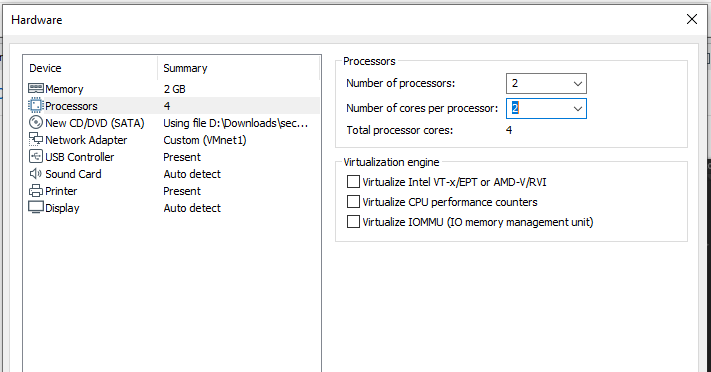
Hình 3.18: Cài đặt bộ nhớ cho máy ảo

* Bước 7: Nhấn Customize Hardware để mở cửa sổ chỉnh phần cứng



Hình 3.19: Mở cửa sổ cài đặt phần cứng

* Bước 8: Chọn bộ nhớ RAM cho máy ảo (tối thiểu 2GB), số bộ xử lý (2 bộ xử lý, 2 lõi mỗi bộ giúp xử lý tốt hơn), chọn VMnet1 làm Network Adapter theo như sơ đồ hệ thống đã trình bày trong chương 2.



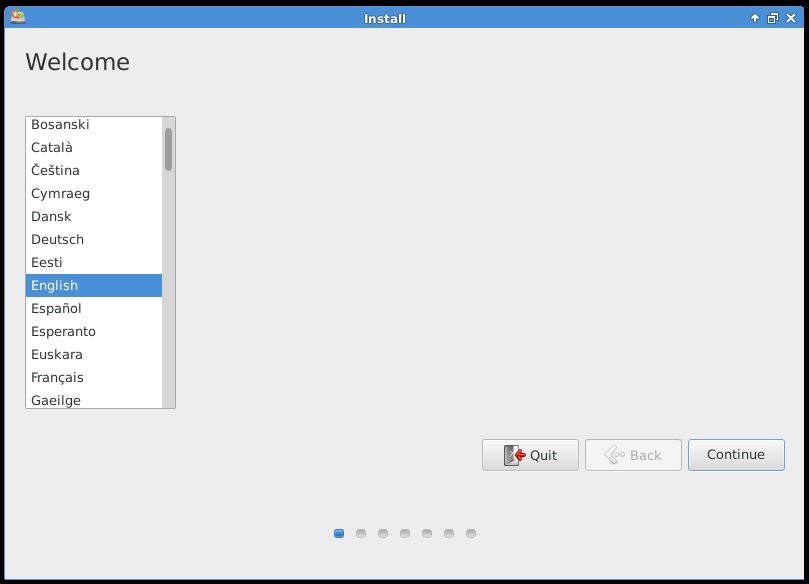
Hình 3.20: Cài đặt phần cứng cho máy ảo

* Bước 9: Nhấn Close rồi Finish để tiếp tục thêm máy ảo vào máy.
* Bước 10: Mở máy ảo vừa thêm để cài đặt hệ thống.



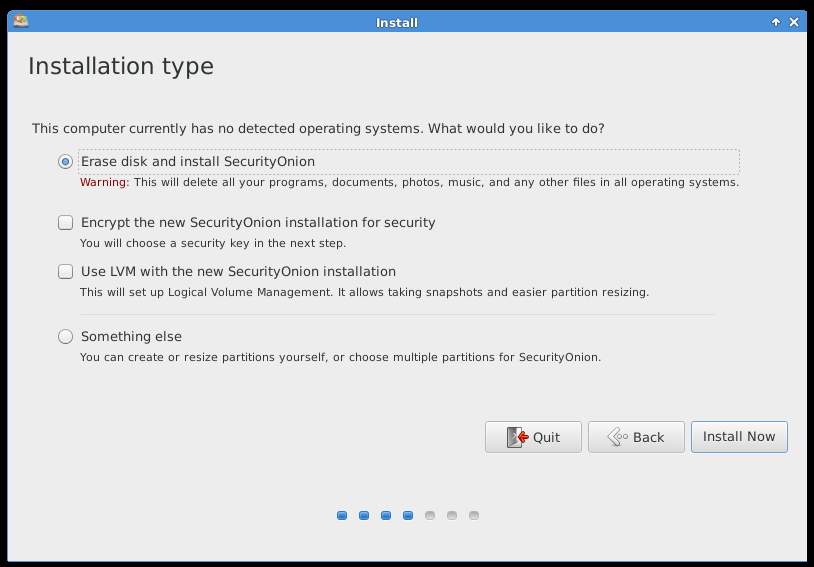
Hình 3.21: Giao diện máy ảo mới thêm vào VMware

* Bước 11: Chọn ngôn ngữ rồi nhấn Continue để tiếp tục.



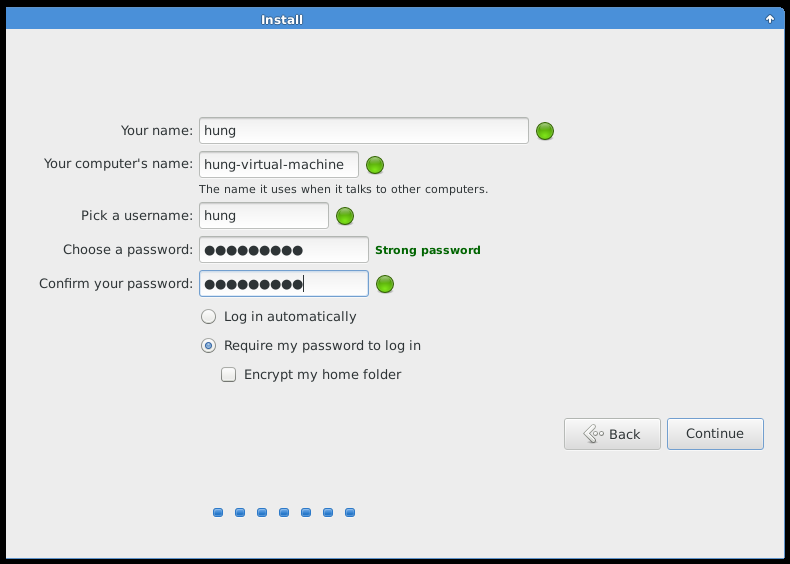
Hình 3.22: Giao diện chọn ngôn ngữ cho Security Onion

* Bước 12: Nhấn Continue, chọn Erase disk and install SecurityOnion rồi nhấn Install Now, Continue để cài đặt.



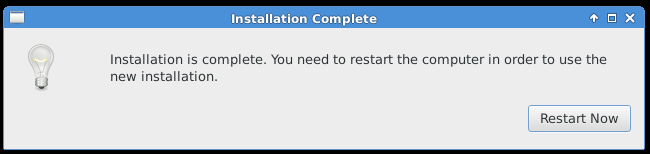
Hình 3.23: Chọn cách thức cài đặt Security Onion

* Bước 13: Nhấn Continue liên tục đến màn hình cài đặt tài khoản, cài đặt tài khoản sử dụng. Trong bài này tài khoản sẽ tên là **hung** với mật khẩu là **123qwe!@#**. Nhấn Continue để hoàn thiện cài đặt.



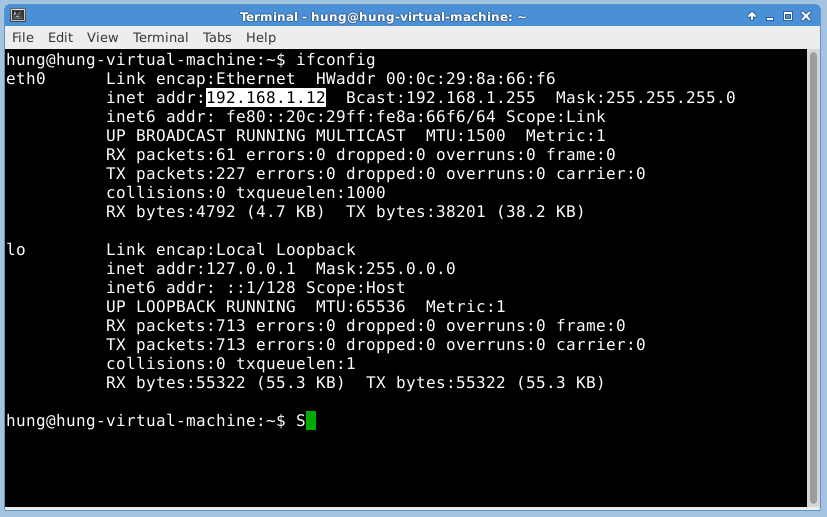
Hình 3.24: Cài đặt tài khoản sử dụng

* Bước 14: Nhấn Restart Now để khởi động lại máy ảo sau cài đặt



Hình 3.25: Khởi động lại máy ảo

Giờ đây ta đã có một máy ảo Security Onion sẵn sàng cài đặt các công cụ, chức năng hoạt động như một hệ thống phát hiện xâm nhập mạng với tài khoản đăng nhập hung/123qwe!@#. Nhập lệnh ***ifconfig*** trong Terminal để kiểm tra địa chỉ IP được cấp phát tự động từ DHCP\_Router.



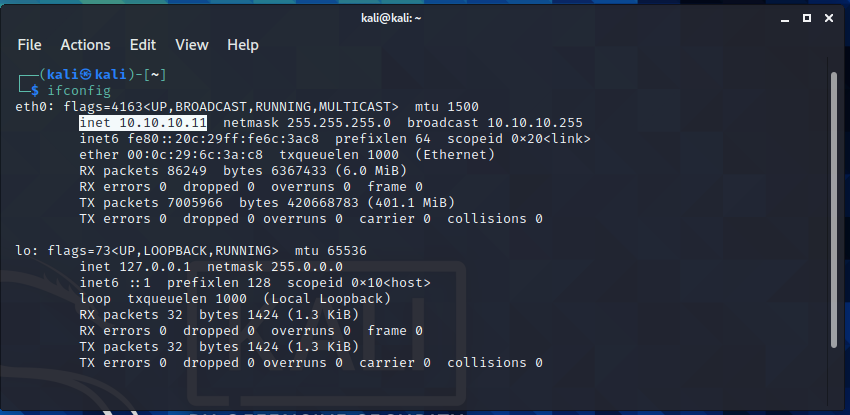
Hình 3.26: Máy ảo Security Onion với địa chỉ được cấp phát tự động từ DHCP\_Router

* + 1. ***Cài đặt Kali Linux***

Dễ dàng tải phiên bản Kali Linux dành riêng cho VMware đã được cấu hình sẵn đầy đủ cả về phần cứng cũng như các phần mềm cần thiết, chỉ cần giải nén rồi thêm vào VMware tại đại chỉ: [*https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/*](https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/)

Sau thi thêm vào VMware ta chỉ cần nhấn đúp vào mục Network Adapter để chỉnh thành VMnet2 như sơ đồ hệ thống trong chương 2.

Khởi động máy ảo, đăng nhập bằng tài khoản Kali/Kali nhập lệnh ***ifconfig*** trong Terminal để kiểm tra địa chỉ IP được cấp phát tự động từ DHCP\_Router.



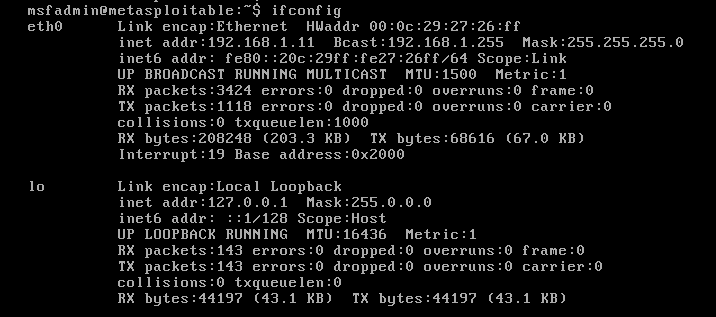
Hình 3.27: Máy ảo Kali Linux với địa chỉ được cấp phát tự động từ DHCP\_Router

* + 1. ***Cài đặt Metasploitable***

Tương tự Kali Linux, Metasploitable cũng có phiên bản dành riêng cho VMware chỉ cần tải về rồi thêm vào tại địa chỉ: [*https://sourceforge.net/projects/metasploitable/*](https://sourceforge.net/projects/metasploitable/)

Sau thi thêm vào VMware ta chỉ cần nhấn đúp vào mục Network Adapter để chỉnh thành VMnet1 như sơ đồ hệ thống trong chương 2.

Khởi động máy ảo, đăng nhập bằng tài khoản msfadmin/msfadmin nhập lệnh ***ifconfig*** ngay tại giao diện Terminal để kiểm tra địa chỉ IP được cấp phát tự động từ DHCP\_Router.

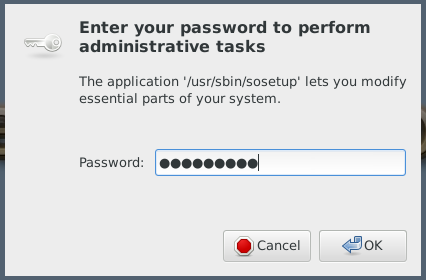


Hình 3.28: Máy ảo Metasploitable với địa chỉ được cấp phát tự động từ DHCP\_Router

* 1. **Mô phỏng hệ thống phát hiện xâm nhập**
     1. ***Cấu hình Security Onion***

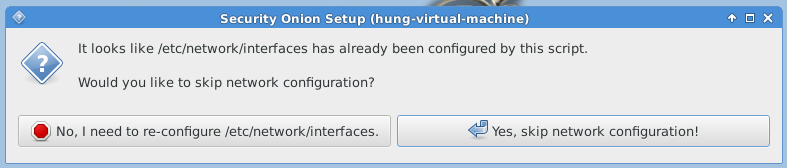
Thực hiện cấu hình Security Onion để cài đặt các ứng dụng cần thiết cho chức năng như một hệ thống phát hiện xâm nhập mạng.

* Bước 1: Nhấn đúp vào biểu tượng Setup ở màn hình chính để bắt đầu cài đặt. nhập mật khẩu đã đăng kí lúc cài máy: **123qwe!@#**. Nhấn OK để tiếp tục.



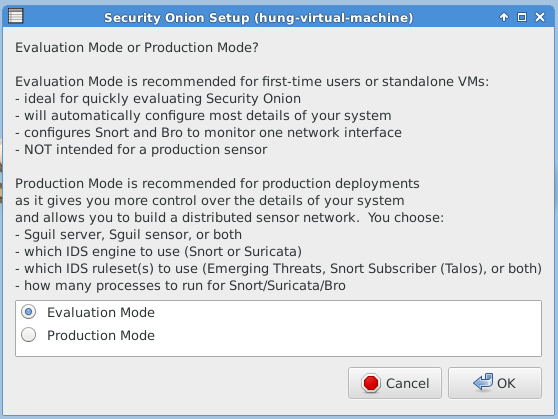
Hình 3.29: Nhập mật khẩu để bắt đầu cài đặt kiến trúc

* Bước 2: Nhấn Yes, continue rồi Yes, skip network configuration vì đã cài đặt và kiểm tra địa chỉ IP máy khi mới cài.



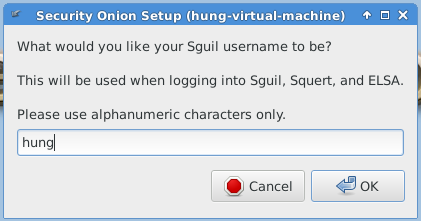
Hình 3.30: Không cần cài đặt mạng vì đã được cài từ trước

* Bước 3: Chọn kiến trúc Evaluation rồi nhấn OK để tiếp tục.



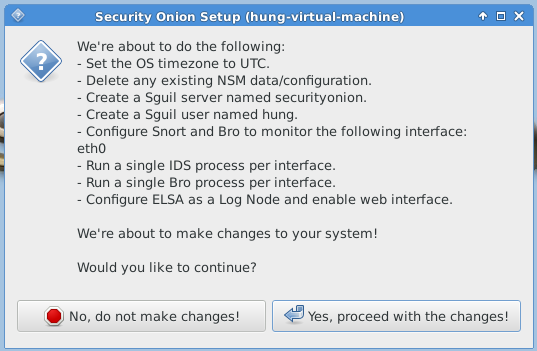
Hình 3.31: Giao diện lựa chọn kiến trúc hệ thống

* Bước 4: Nhấn Yes, continue rồi thêm tài khoản và mật khẩu khi sử dụng hệ thống phát hiện xâm nhập mạng. Ở đây ta sử dụng lại tài khoản **hung/123qwe!@#**.



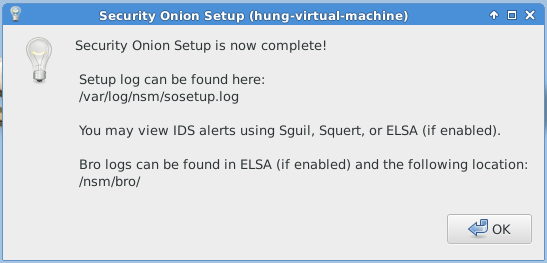
Hình 3.32: Cài đặt tài khoản sử dụng cho hệ thống phát hiện xâm nhập

* Bước 5: Chọn Yes, proceed with the changes! để hoàn thiện cấu hình và thêm các ứng dụng cần thiết cho hệ thống phát hiện xâm nhập mạng.



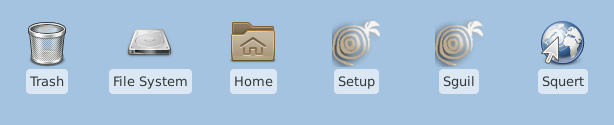
Hình 3.33: Xác nhận hoàn thiện cấu hình hệ thống

* Bước 6: Nhấn OK để xác nhận các thông báo hoàn thiện cài đặt kiến trúc.



Hình 3.34: Xác nhận các thông báo hoàn thiện cấu hình hệ thống

Giờ đây ta đã có một máy ảo Security Onion cài đặt kiến trúc Evaluation tích hợp với công cụ cần thiết sẵn sàng hoạt động như một hệ thống phát hiện xâm nhập mạng.

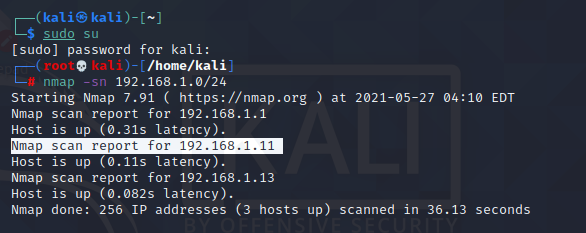


Hình 3.35: Các công cụ của hệ thống phát hiện xâm nhập mạng

* + 1. ***Chức năng phát hiện thăm dò dữ liệu bằng nmap***

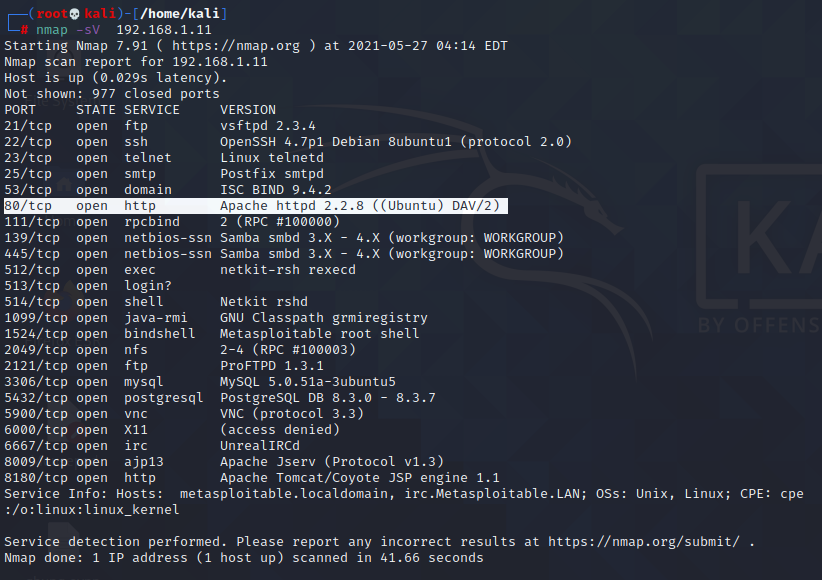
Thực hiện thăm dò dữ liệu sử dụng công cụ nmap của Kali Linux, nâng quyền root bằng lệnh ***sudo su*** sau đó nhập mật khẩu **kali** của tài khoản, lần lượt thực hiện các giai đoạn trong quá trình thăm dò dữ liệu sau:

* Tìm kiếm địa chỉ IP các máy client trong mạng con 192.168.1.0/24: Sử dụng lệnh ***nmap –sn 192.168.1.0/24***. Kết quả trả về là địa chỉ IP của các máy trạm trong mạng con nhắm tới. Mục tiêu của chúng ta là máy **192.168.1.11**.



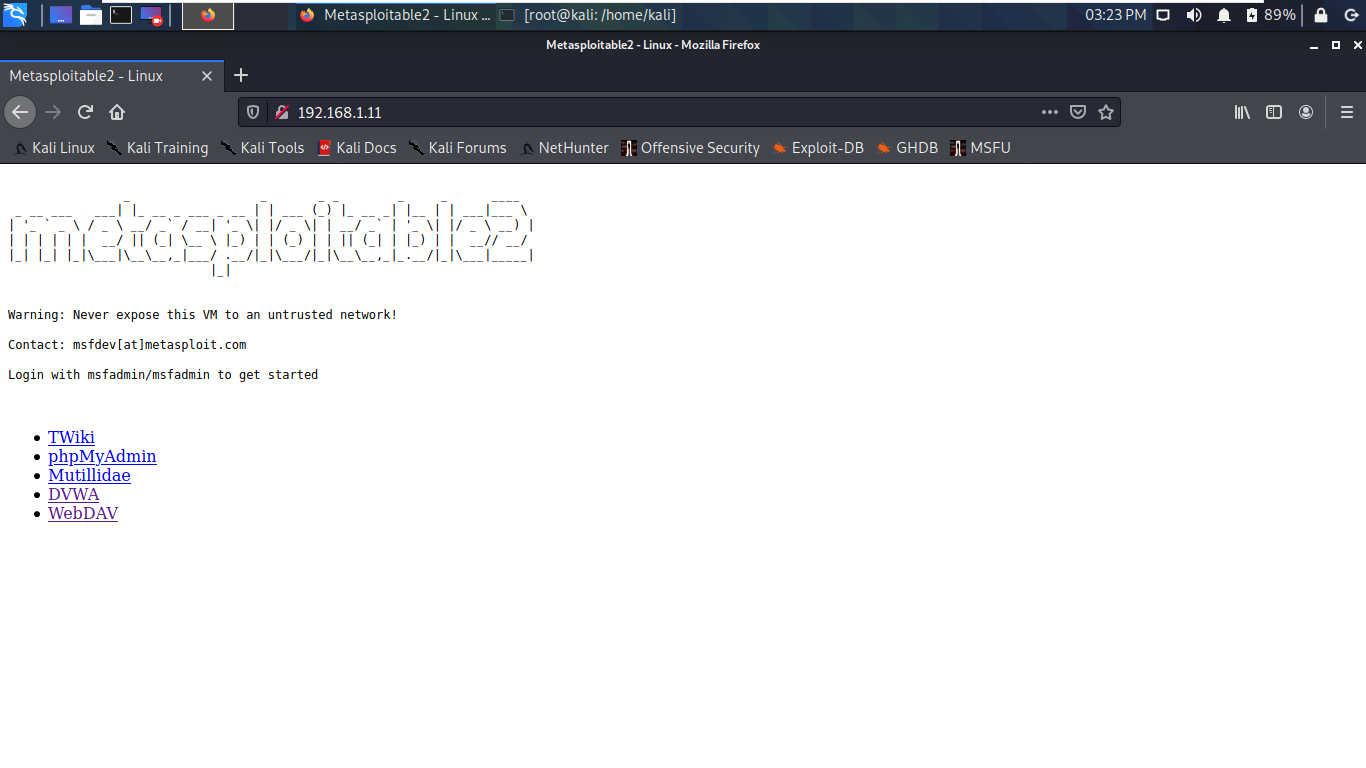
Hình 3.36: Quét tìm các máy trạm của mạng con

* Tìm kiếm thông tin về các port đang mở của mục tiêu 192.168.1.11: Sử dụng lệnh  ***nmap –sV 192.168.1.11***. Kết quả trả về là thông tin về các port đăng mở cũng như dịch vụ được cài đặt trên port đó. Mục tiêu của chúng ta là port **80**  với dịch vụ http Apache từ đó có thể thực hiện các cuộc tấn công mạng vào ứng dụng Web và Database.



Hình 3.37: Quét tìm các cổng và dịch vụ trên cổng đang mở

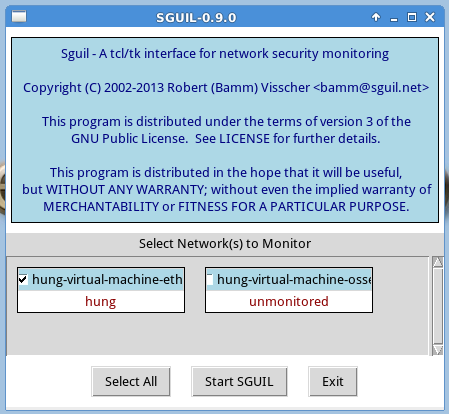
Giờ đây kẻ tấn công có thể tận dụng ứng dụng web được cài ở port 80 để khai thác dần các loại lỗ hổng bảo mật thường gặp về Web, Database như XSS, SQL injection,…



Hình 3.38: Giao diện ứng dụng web tại cổng 80 của mục tiêu

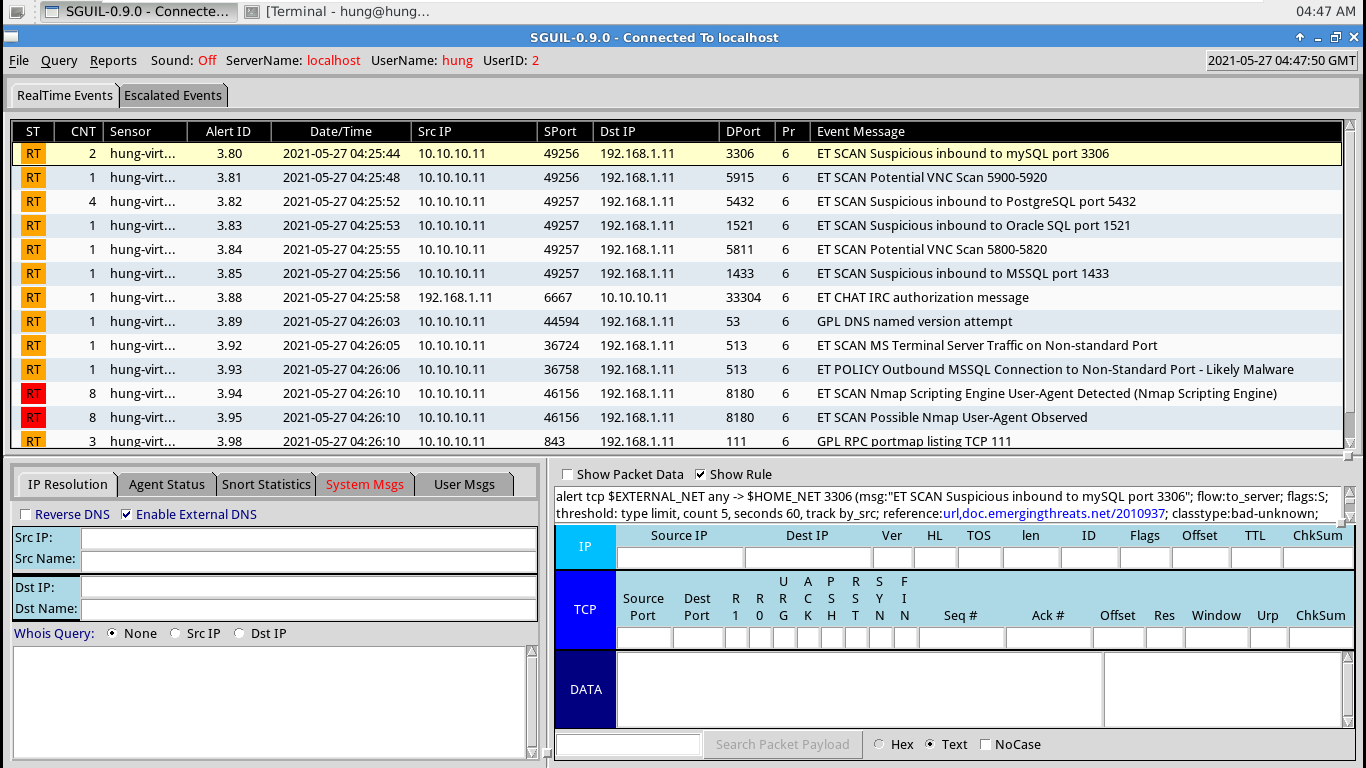
Security Onion sau khi cài đặt như một hệ thống phát hiện xâm nhập mạng sẽ luôn đánh hơi và phát hiện các mối xâm nhập mạng theo thời gian thực vào mạng con. Hệ thống được tích hợp các công cụ hỗ trợ giao diện trực quan hơn cho người dùng như Sguil, Squert.

Bật ứng dụng Sguil sau khi đăng nhập bằng tài khoản đã đăng kí sử dụng hệ thống, trường hợp này là tài khoản **hung/123qwe!@#**. Ta chọn interface muốn đánh hơi, phát hiện xâm nhập là hung-virtual-machine-eth0 rồi chọn Start Sguil để khởi chạy.



Hình 3.39: Chọn interface quản lý phát hiện xâm nhập

Sau khi khởi động ứng dụng sẽ tự động thu thập lại các thông tin phát hiện xâm nhập trong quá khứ cũng như ngay tại thời điểm hiện tại theo thời gian thực để thể hiện các thống báo về mối nguy hại xâm nhập.



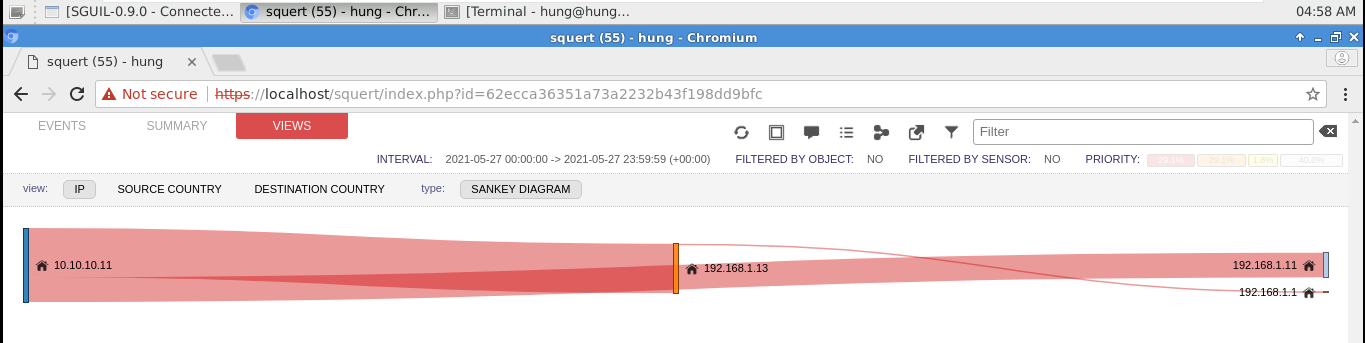
Hình 3.40: Các cảnh báo về các hành vi xâm nhập được phát hiện theo thời gian thực

Dễ dàng nhận thấy rằng bên cạch việc phát hiện việc quét các cổng một cách đáng nghi hệ thống còn phát hiện việc sử dụng công cụ Nmap của máy tại địa chỉ ip 10.10.10.11 chính là Kali Linux lên mục tiêu 192.168.1.11.



Hình 3.41: Phát hiện hành vi thăm dò bằng công cụ nmap

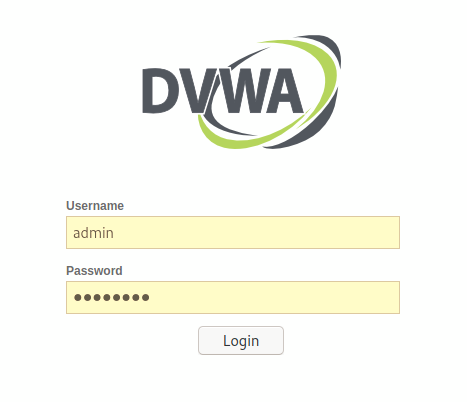
Công cụ Squert với việc thống kê các luồng dữ liệu của các hành vi xâm nhập hỗ trợ đắc lực trong việc đễ dàng phát hiện kẻ tấn công cũng như mục tiêu đang bị nhắm đến. Dễ dàng nhận ra kẻ tấn công tại địa chỉ 10.10.10.11 và đang muốn nhắm đến máy trạm tại địa chỉ 192.168.1.11.



Hình 3.42: Ứng dụng Web Squert giúp phân tích, thống kê hỗ trợ phát hiện xâm nhập

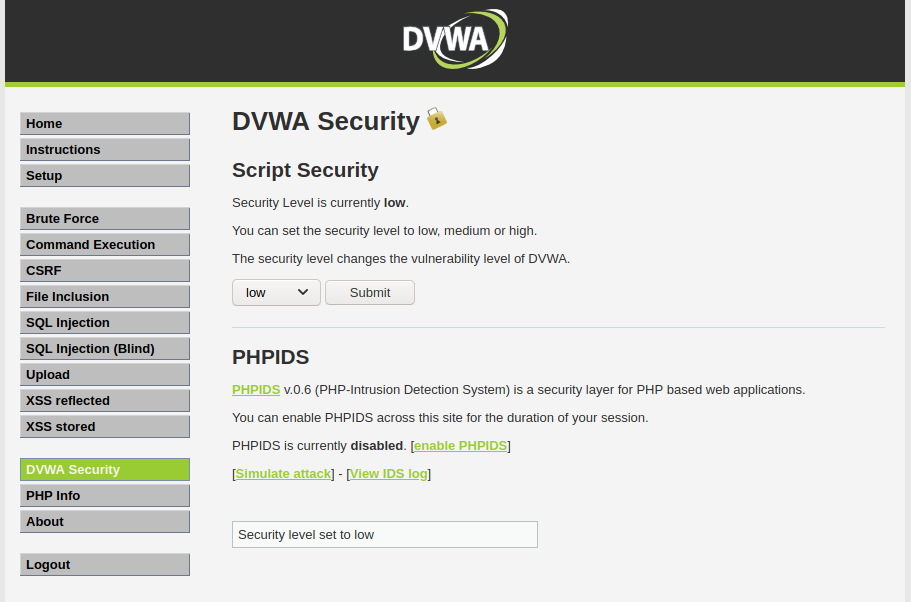
* + 1. ***Chức năng phát hiện tấn công XSS***

Sau khi truy cập vào ứng dụng web của máy trạm tại địa chỉ 192.168.1.11, ta sẽ sử dụng Damn vulnareble web application (DVWA) để thực hiện tấn công XSS. Ta sẽ đăng nhập bằng tài khoản admin/password từ đó thực hiện các chức năng mà ứng dụng hỗ trợ để kiểm thử các loại tấn công.



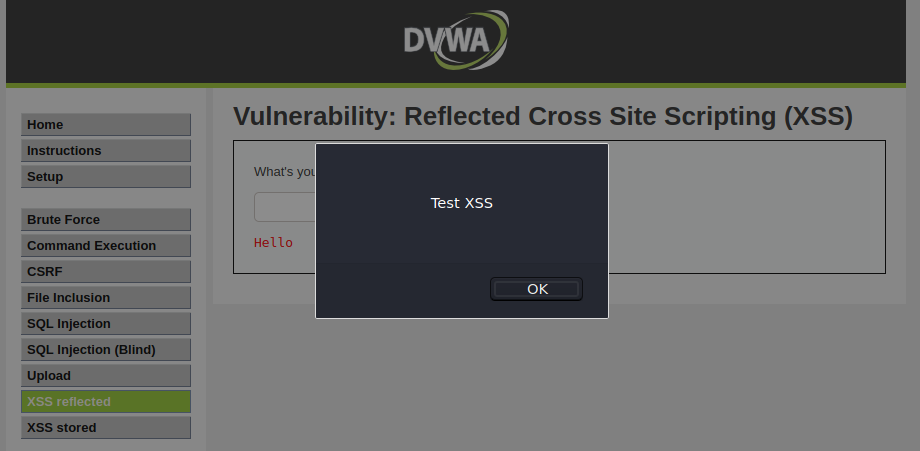
Hình 3.43: Đăng nhập vào ứng dụng DVWA bằng tài khoản admin

Vào mục DVWA Security sau đó cài về low để dễ dàng cho việc kiểm thử các cuộc tấn công trên ứng dụng web.



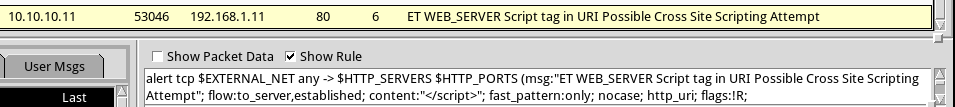
Hình 3.44: Cài đặt mức bảo mật về low để dễ dàng cho kiểm thử

Nhấn chọn XSS Reflected để sử dụng chức năng tấn công XSS. Tại form What’s your name ta sẽ nhập một đoạn Script đơn giản ***<script> alert(“Test XSS”) </script>*** để chạy trên Server nếu trả về thông báo có nghĩa là ứng dụng web này có thể bị tấn công bởi XSS.



Hình 3.45: Ứng dụng web này dễ dàng bị tấn công bởi XSS

Ngay tại thời điểm xảy ra tấn công XSS Security Onion sẽ chặn bắt các gói tin trong môi trường mạng, phân tích để phát hiện các dấu hiệu xâm nhập mạng. Các thông báo sẽ được lưu lại cơ sở dữ liệu với dạng log và thể hiện trực tiếp theo thời gian thực trên công cụ Sguil. Công cụ không chỉ hỗ trợ làm giao diện thể hiện các xâm nhập mà còn đi kèm theo bộ luật cách phát hiện xâm nhập giúp dễ dàng trong việc tìm hiểu loại tấn công đang thực hiện. Trong trường hợp tấn công XSS thì gói tin bắt được có dấu hiệu của một đoạn Script đính kèm vào url. Việc có thể xem được luật cũng như có thể xem toàn bộ gói tin hỗ trợ đắc lực trong việc thiết kế bộ luật riêng sau này.



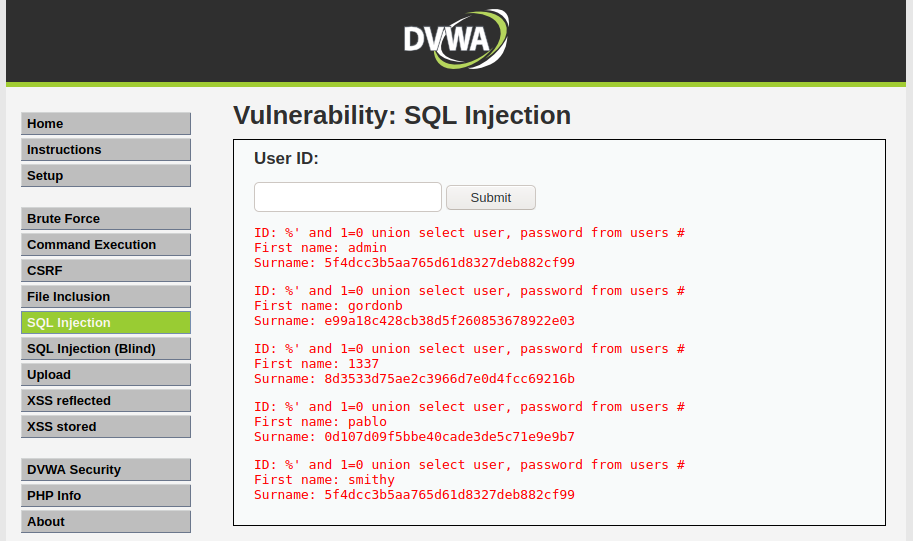
Hình 3.46: Phát hiện tấn công XSS và luật giúp phát hiện

* + 1. ***Chức năng phát hiện SQL injection***

Bên cạnh việc cho phép kiểm thử một cuộc tấn công XSS cơ bản lên ứng dụng Web, DVWA còn hỗ trợ thực hiện mô phỏng tấn SQL injection lên cơ sở dữ liệu cụ thể ở đây là thực hiện các lệnh truy xuất không được phép lên MySQL.

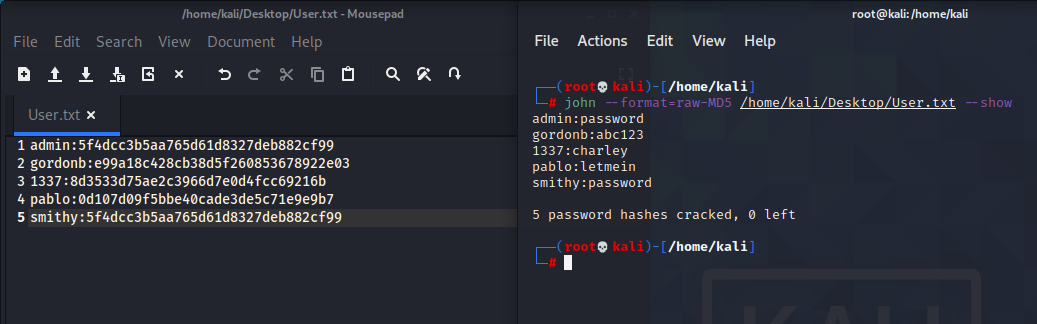
Nhấn chọn SQL Injection để sử dụng chức năng tấn công SQL injection vào cơ sở dữ liệu của trang web. Như thường lệ khi nhập số id của tài khoản bất kì tại Form User ID web sẽ trả về cho ta thông tin Họ và tên của tài khoản nhưng giờ đây khi hệ thống có lỗ hổng về SQL injection ta có thể dễ dàng truy xuất username và password của tài khoản được lưu trong cơ sở dữ liệu với lệnh ***%' and 1=0 union select user, password from users #***. Giải thích về lệnh trên thì khi có ID lấy từ Form sẽ được thêm trực tiếp vòa một lệnh truy xuất có sẵn, việc cho ***ID=% and 1=0*** sẽ luôn trả về false tức bảng rỗng, bảng này sẽ được hợp với một bảng chứa user và password chính là mục tiêu nhắm đến trong cuộc tấn công này.

Kết quả trả về giờ đây thay vì họ và tên của một tài khoản với ID nhập tại Form thì là một bảng chứa username và pasword dạng mã hash MD5 của tất cả các tài khoản trong cơ sở dữ liệu.

******

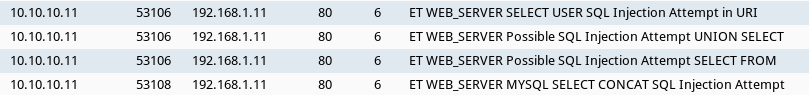
Hình 3.47: Thực hiện tấn công SQL injection để lấy thông tin từ cơ sở dữ liệu

Từ đây ta có thể lưu các cặp username và password này lại một file txt với tên User.txt bất kì rồi dùng các công cụ giải mã MD5 để tìm mật khẩu của từng tài khoản. Kali Linux được cài sẵn công cụ John the Ripper cho giải mã mật khẩu MD5 chỉ với một lệnh đơn giản là: ***john --format=raw-MD5 [đường dẫn đến file chứa mật khẩu] --show***. Sau khi giải mã xong mật khẩu sẽ được hiện lên màn hình và giờ đây ta đã có tất cả tài khoản cũng như mật khẩu trong cơ sở dữ liệu của trang web này.

******

Hình 3.48: Tài khoản và mật khẩu thu được có thể mã hóa một cách dễ dàng

Tương tự như tấn công XSS, khi xảy ra một cuộc tấn công SQl injection Security Onion dễ dàng phân tích các gói tin bắt được để phát hiện xâm nhập theo thời gian thực một cách nhanh nhất có thể. Sau khi phát hiện hành vi xâm nhập có thể thực hiện ngăn chặn bằng nhiều cách khác nhau như chặn kết nối, nâng cấp cơ sở dữ liệu,…

******

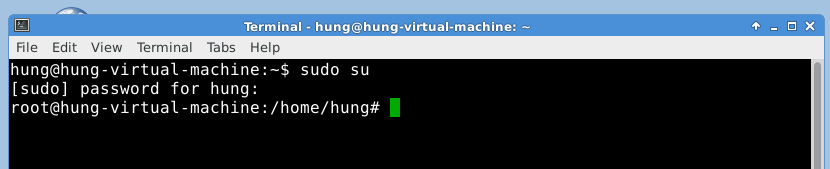
Hình 3.49: Phát hiện tấn công SQL injection

* + 1. ***Thêm luật giúp nâng cao khả năng phát hiện xâm nhập của hệ thống***

Bên cạnh việc sử dụng bộ luật có sẵn khi cài đặt kiến trúc cho Security Onion ta có thể tải luật từ các cộng đồng chuyên tạo luật cho Snort/Suricata hay đặc biệt hơn ta có thể tự thiết kế riêng một bộ luật của bản thân. Việc viết luật cũng như sử dụng công cụ tích hợp sẵn để cập nhật các luật đã viết lên cơ sở dữ liệu khá đơn giản và dễ dàng. Có thể viết luật thủ công theo chuẩn cấu trúc như đã trình bày ở Chương 1 hay sử dụng sẵn công cụ hỗ trợ tạo luật do em tự thiết kế.

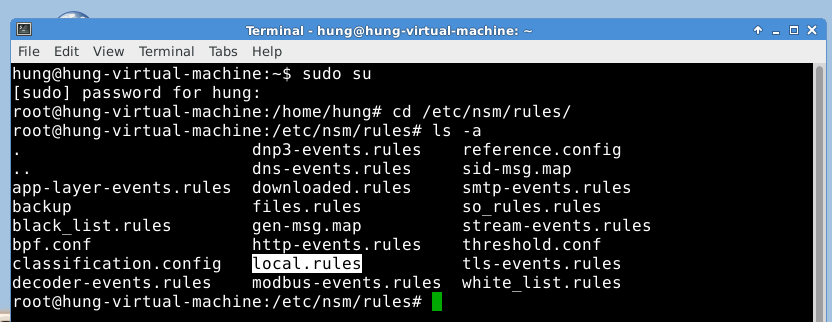
Cách 1: Viết và cập nhật luật thủ công

* Bước 1: Mở Terminal bằng tổ hợp phím ***Ctrl+Alt+T***, chuyển về tài khoản root bằng cách nhập lệnh ***sudo su*** rồi gõ mật khẩu là ***123qwe!@#***.



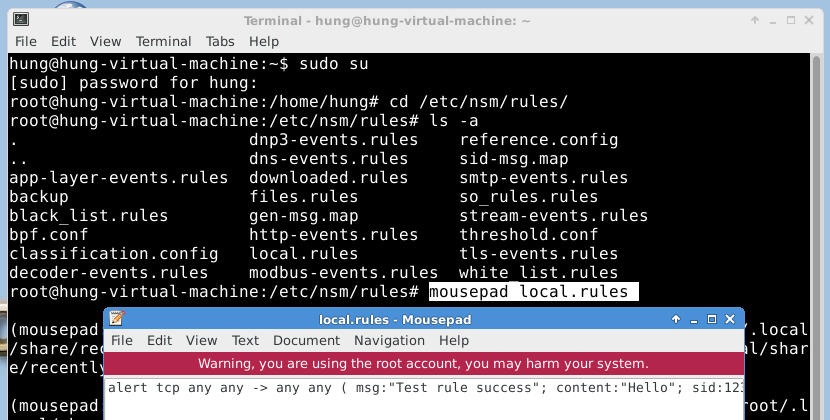
Hình 3.50: Sử dụng Terminal truy cập quyền root

* Bước 2: Chuyển địa chỉ đến thư mục chứa file ghi luật bằng câu lệnh ***cd /etc/nsm/rules/***, sử dụng lệnh ***ls –a*** để thấy rằng thư mục này chứa file ***local.rules*** chính là file ta dùng để viết luật do bản thân tự thiết kế.



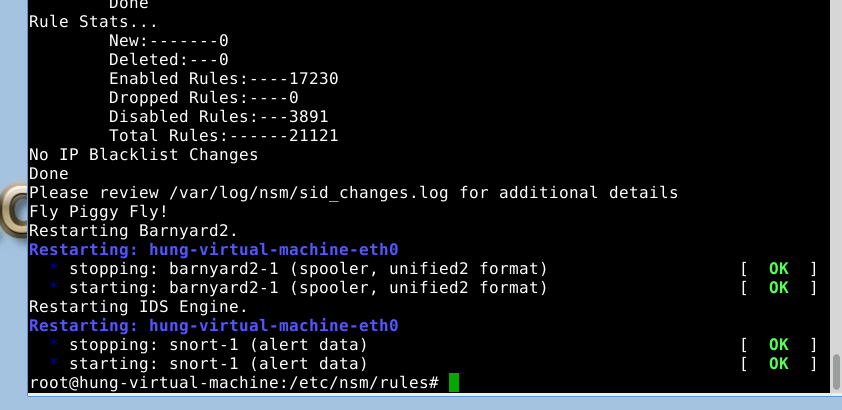
Hình 3.51: Chuyển đến thư mục chứa luật

* Bước 3: Mở file ***local.rules*** bằng công cụ mousepad để có thể thêm luật bằng cách sử dụng lệnh  ***mousepad local.rules***.
* Bước 4: Thêm luật của bản thân vào file ***local.rules***, ví dụ ta muốn cảnh báo “Test rule success” khi có khi có một gói tin TCP chứa nội dung “Hello” gửi đến mạng con cần bảo vệ ta thêm luật  ***alert tcp any any -> any any ( msg:"Test rule success"; content:"Hello"; sid:1234567; )*** vào file rồi lưu lại.



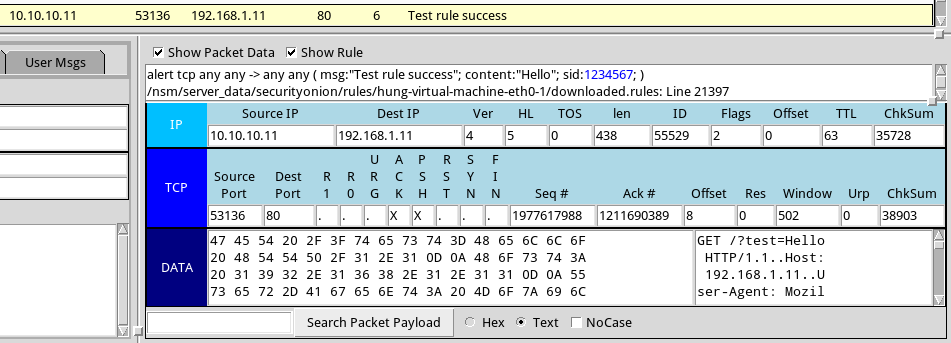
Hình 3.52: Mở file để thêm luật của bản thân

* Bước 5: Đóng file đã lưu, sử dụng lệnh ***rule-update*** để Security Onion tự động kiểm tra luật đúng cấu trúc, không trùng lặp rồi cập nhật vào cơ sở dữ liệu. Lưu ý luật ghi trong file  ***local.rules*** phải hoàn toàn đúng cú pháp, cấu trúc nếu không khi cập nhật sẽ bị kiểm tra bỏ qua nếu chỉ cần sai một lỗi dù là nhỏ nhất.



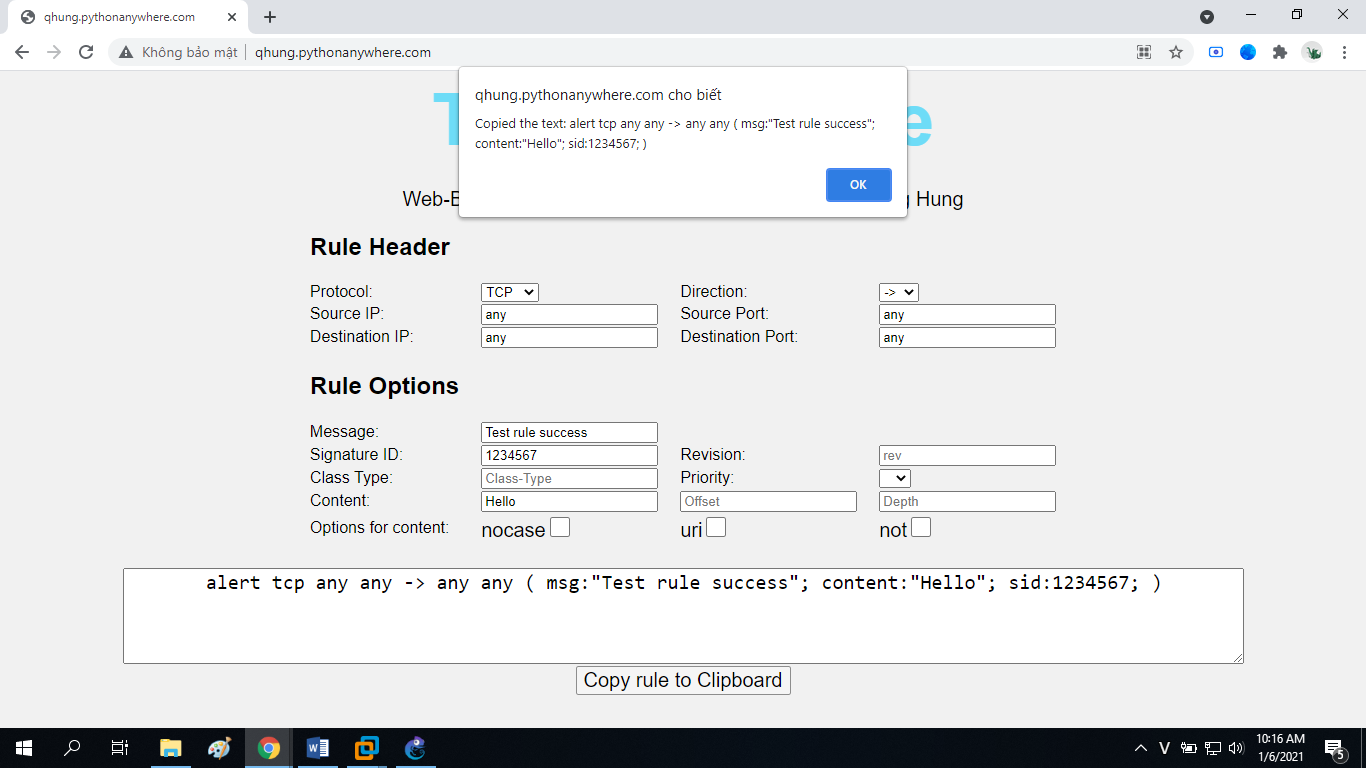
Hình 3.53: Luật được cập nhật thành công vào cơ sở dữ liệu

Nếu thành công luật sẽ được tự động thêm vào cuối file ***downloaded.rules*** và được sử dụng cùng với các luật có sẵn của hệ thống. Kiểm tra luật bằng cách tại máy Kali Linux truy cập vào địa chỉ của máy Metasploitable, thêm tham số bất kì chứa “Hello” như ví dụ sau ***http://192.168.1.11/?test=Hello*** nếu bên công cụ Sguil của Security Onion có thông báo tức là luật đã được thêm thành công và hoạt động ổn định.



Hình 3.54: Luật hoạt động tốt khi có tấn công thử nghiệm

Cách 2: Sử dụng công cụ hỗ trợ tạo luật do em tự viết. Công cụ này có 2 phiên bản: một là bản html hoạt động offline nhưng phải có file ***Tool.html*** được cài đặt trong máy và một bản được viết bằng Django-Python được đẩy lên mạng tại địa chỉ [*https://qhung.pythonanywhere.com/*](https://qhung.pythonanywhere.com/). Cả hai phiên bản này đều hoạt động giống nhau, giao diện bao gồm các Form nhập dữ liệu. Các form này khi nhập dữ liệu sẽ tự động kiểm tra lỗi cú pháp cũng như các lỗi về nghĩa của thành phần tương ứng trong cấu trúc luật. Nếu sai sẽ thông báo để nhập lại, đúng sẽ thêm thành phần đó vào luật. Khi thêm đủ các thành phần có thể dễ dàng sao chép luật hoàn thiện vào bộ nhớ đệm chỉ với một nút Copy Rule to Clipboard sau đó dễ dàng dán vào file ***local.rules*** như trình bày ở trên để cập nhật luật.



Hình 3.55: Công cụ hỗ trợ tạo luật đơn giản

**Kết chương**

Trong chương này em đã tiến hành xây dựng mô hình mô phỏng phát hiện tấn công, xâm nhập của Kali Linux tới máy nạn nhân Metasploitable bởi hệ thống phát hiện xâm nhập mạng Security Onion. Bên cạnh cách cấu hình, cài đặt chi tiết hệ thống còn có cách thức tạo và cập nhật luật bổ sung cho cơ sở dữ liệu.

**KẾT LUẬN**

1. **Kết quả đạt được**

Về mặt lý thuyết: Nâng cao khả năng phân tích hệ thống, nâng cao các kiến thức, kinh nghiệm làm việc với các loại máy ảo cũng như lập trình web tạo công cụ bằng ngôn ngữ html, python.

Về mặt nghiên cứu công nghệ: Sau khi hoàn thành đồ án chuyên ngành mạng em đã nắm bắt được các đặc điểm của công nghệ mình tìm hiểu và sử dụng trong đề tài, biết ứng dụng nó thực tế trên chính đồ án này và nắm rõ được các thao tác phát hiện xâm nhập mạng cơ bản. Tuy nhiên, thời gian tìm hiểu có hạn và khả năng học hỏi của bản thân còn hạn chế nên còn nhiều tính năng cũng như kỹ thuật xử lý, phân tích em còn chưa nắm bắt hết.

Về mặt xây dựng: ứng dụng đã có một số kết quả nhất định. Hệ thống còn một số chưa tối ưu. Em hi vọng trong thời gian sắp tới, hệ thống sẽ được xây dựng hoàn thiện và tối ưu hơn để đưa vào sử dụng thực tế.

Tuy nhiên vẫn có những mặt hạn chế:

* Mô hình chỉ phù hợp cho các mạng vừa và nhỏ.
* Chỉ có tính năng phân tích phát hiện các xâm nhập nên vẫn cẩn phải có tường lửa hay hệ thống ngăn chặn xâm nhập để tăng tính hiệu quả.
* Công cụ hỗ trợ tạo luật còn khá đơn giản, giao diện thô sơ không bắt mắt.

1. **Hướng phát triển**

Trong tương lai, có thời gian nhiều hơn, em muốn hoàn thiện hệ thống cũng như công cụ của mình theo hướng sau:

* Nâng cao khả năng phân tích và phát hiện xâm nhập của hệ thống bằng cách nghiên cứu thêm nhiều loại tấn công mới để bổ sung vào cơ sở dữ liệu luật của hệ thống.
* Nâng cấp về phần cứng cũng như nghiên cứu thêm về các công cụ hỗ trợ của Security Onion để phát hiện đạt năng suất cũng như hiệu quả cao hơn.
* Nâng cấp công cụ hỗ trợ tạo luật với nhiều tính năng cũng như giao diện thẩm mỹ, dễ sử dụng hơn.

**TÀI LIỆU THAM KHẢO**

[1] Security Onion Solutions – LLC, “Security Onion Documentation”, 2021.

[Online]. Xem tại: <https://docs.securityonion.net/en/16.04/>. [Ngày truy cập: 2/2021].

[2] [Online]. Xem tại: <https://nmap.org/>. [Ngày truy cập: 3/2021].

[3] Galaxy Technologies LLC, “GNS3 Documentation”, 2021.

[Online]. Xem tại: <https://docs.gns3.com/docs/>. [Ngày truy cập 3/2021].

[4] OffSec Services, “Kali Linux Documentation”, 2021.

[Online]. Xem tại: <https://www.kali.org/docs/>. [Ngày truy cập: 3/2021].

[5] [Online]. Xem tại: <https://dvwa.co.uk/>. [Ngày truy cập: 3/2021].

[6] [Online]. Xem tại: <https://docs.rapid7.com/metasploit/metasploitable-2/>. [Ngày truy cập: 4/2021].

**PHỤ LỤC**

Code Tool tạo luật cho Security Onion bằng HTML:

File Tool.html

<head>

<script>

***function*** ValidateIPaddress**(**ipaddress**)** **{**

***var*** ipformat **=** /^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$/**;**

***if*** **(**ipaddress**==**'any' **||** ipaddress**==**""**)** ***return*** ***true*;**

***if*** **(**ipaddress.match**(**ipformat**))** ***return*** ***true*;**

***return*** ***false*;**

**}**

***function*** ValidatePort**(**port**)** **{**

***if*** **(**port**==**"any" **||** port**==**""**)** ***return*** ***true*;**

***if*** **(!(**/^[1-9]\d\*$/**.**test**(**port**)** **&&** 1 **<=** 1 **\*** port **&&** 1 **\*** port **<=** 65535**))** ***return*** ***false***

***return*** ***true*;**

**}**

***function*** ValidateSID**(**num**)** **{**

***if*** **(!(**/^[1-9]\d\*$/**.**test**(**num**)** **&&** 1000000 **<=** 1 **\*** num **&&** 1 **\*** num **<=** 999999999**))** ***return*** ***false***

***return*** ***true*;**

**}**

***function*** ValidateRange**(**num**)** **{**

***if*** **(!(**/^[1-9]\d\*$/**.**test**(**num**)** **&&** 1 **<=** 1 **\*** num **&&** 1 **\*** num **<=** 999999999**))** ***return*** ***false***

***return*** ***true*;**

**}**

***function*** ValidateOffset**(**offset**)** **{**

***return*** Number.isInteger**(**1 **\*** offset**);**

**}**

***function*** ValidateDepth**(**num**)** **{**

***if*** **(!(**/^[1-9]\d\*$/**.**test**(**num**)** **&&** 1 **<=** 1 **\*** num **&&** 1 **\*** num **<=** 65535**))** ***return*** ***false***

***return*** ***true*;**

**}**

***function*** ValidateString**(**s**)** **{**

***var*** a **=** s.split**(**' '**);**

***if*** **(**a.length**>**1**)** ***return*** ***false*;**

***return*** ***true*;**

**}**

***function*** FormatContent**(**content**)** **{**

***var*** s **=** content**+**''**;**

s**=**s.replace**(**/\"/g**,**'|22|'**);**

s**=**s.replace**(**/\'/g**,**'|27|'**);**

s**=**s.replace**(**/\;/g**,**'|3b|'**);**

s**=**s.replace**(**/\:/g**,**'|3a|'**);**

s**=**s.replace**(**/\\/g**,**'|5c|'**);**

s**=**s.split**(**'||'**).**join**(**' '**);**

***return*** s**;**

**}**

***function*** fun**()** **{**

***var*** rule **=** "alert "**;**

***var*** protocol **=** document.getElementById**(**"protocol"**).**value**;**

***var*** sip **=** document.getElementById**(**"sip"**).**value**;**

***if*** **(!**ValidateIPaddress**(**sip**))** **{**

sip**=**""**;**

document.getElementById**(**"sip"**).**value**=**"Invalid"**;**

**}**

***var*** sport **=** document.getElementById**(**"sport"**).**value**;**

***if*** **(!**ValidatePort**(**sport**))** **{**

sport**=**""**;**

document.getElementById**(**"sport"**).**value**=**"Invalid"**;**

**}**

***var*** dip **=** document.getElementById**(**"dip"**).**value**;**

***if*** **(!**ValidateIPaddress**(**dip**))** **{**

dip**=**""**;**

document.getElementById**(**"dip"**).**value**=**"Invalid"**;**

**}**

***var*** dport **=** document.getElementById**(**"dport"**).**value**;**

***if*** **(!**ValidatePort**(**dport**))** **{**

dport**=**""**;**

document.getElementById**(**"dport"**).**value**=**"Invalid"**;**

**}**

rule **=** rule**+**protocol**+**" "**+**sip**+**" "**+**sport**+**" "**+**document.getElementById**(**"direction"**).**value**+**" "**+**dip**+**" "**+**dport**;**

***var*** option **=** ""**;**

***var*** msg **=** document.getElementById**(**"msg"**).**value.replace**(**/\"/g**,**"'"**);;**

***if*** **(**msg**!=**""**)** option **+=** 'msg:"'**+**msg**+**'"; '**;**

***var*** content **=** document.getElementById**(**"content"**).**value**;**

***if*** **(**content**!=**""**)** **{**

***if*** **(**document.getElementById**(**"not"**).**checked**)** option **+=** 'content:!"'**;**

***else*** option **+=** 'content:"'**;**

content **=** FormatContent**(**content**);**

option **+=** content**+**'"; '**;**

***var*** offset **=** document.getElementById**(**"offset"**).**value**;**

***if*** **(**offset**!=**""**)**

***if*** **(**ValidateOffset**(**offset**))** option **+=** 'offset:'**+**offset**+**'; '**;**

***else*** document.getElementById**(**"offset"**).**value**=**"Invalid"**;**

***var*** depth **=** document.getElementById**(**"depth"**).**value**;**

***if*** **(**depth**!=**""**)**

***if*** **(**ValidateDepth**(**depth**))** option **+=** 'depth:'**+**depth**+**'; '**;**

***else*** document.getElementById**(**"depth"**).**value**=**"Invalid"**;**

***if*** **(**document.getElementById**(**"nocase"**).**checked**)** option**+=**'nocase; '**;**

***if*** **(**document.getElementById**(**"uri"**).**checked**)** option**+=**'http\_uri; '**;**

**}**

***var*** classtype **=** document.getElementById**(**"classtype"**).**value**;**

ValidateString**(**classtype**);**

***if*** **(**classtype**!=**""**)**

***if*** **(**ValidateString**(**classtype**))** option**+=**'classtype:'**+**classtype**+**'; '**;**

***else*** document.getElementById**(**"classtype"**).**value**=**"Invalid"**;**

***var*** pri **=** document.getElementById**(**"pri"**).**value**;**

***if*** **(**pri**!=**""**)** option**+=**'priority:'**+**pri**+**'; '**;**

***var*** sid **=** document.getElementById**(**"sid"**).**value**;**

***if*** **(**sid**!=**""**)**

***if*** **(**ValidateSID**(**sid**))** option **+=** 'sid:'**+**sid**+**'; '**;**

***else*** document.getElementById**(**"sid"**).**value**=**"Invalid"**;**

***var*** rev **=** document.getElementById**(**"rev"**).**value**;**

***if*** **(**rev**!=**""**)**

***if*** **(**ValidateRange**(**rev**))** option **+=** 'rev:'**+**rev**+**'; '**;**

***else*** document.getElementById**(**"rev"**).**value**=**"Invalid"**;**

***if*** **(**option**!=**""**)** rule **=** rule**+**" ( "**+**option**+**")"**;**

document.getElementById**(**"view"**).**value **=** rule**;**

**}**

***function*** save**()** **{**

***var*** copyText **=** document.getElementById**(**"view"**);**

copyText.select**();**

document.execCommand**(**"copy"**);**

alert**(**"Copied the text: " **+** copyText.value**);**

**}**

</script>

<style type=**"text/css"**>

**body**

**{**

**background-color: #f1f1f1;**

**white-space: nowrap;**

**overflow: auto;**

**text-align: center;**

**font-family: Helvetica;**

**vertical-align: middle;**

**font-size: 0;**

**}**

**.TOOL {**

**color: #6edcf7;**

**font-size: 72px;**

**font-weight: bold;**

**font-family: Helvetica;**

**}**

**.header2{**

**font-size: 20px;**

**}**

**td {**

**padding:0 10px 0 10px;**

**}**

**input.largerCheckbox {**

**width: 20px;**

**height: 20px;**

**}**

</style>

</head>

<body>

<div class=**"header2"**>

<h1><span class=**"TOOL"**>**Tool make rule**</span></h1>

<p style=**"text-align: center;"**><span style=**"font-size: 20px;"**>**Web-Based Security Onion rule creator by Truong Quang Hung**</span></p>

</div>

<form >

<table align=**"center"**>

<tr>

<td><h2>**Rule Header**</h2></td>

</tr>

<tr>

<td>**Protocol:** </td>

<td>

<select id=**"protocol"** onblur=fun() onchange=fun()>

<option value=**"tcp"**>**TCP**</option>

<option value=**"udp"**>**UDP**</option>

<option value=**"icmp"**>**ICMP**</option>

<option value=**"ip"**>**IP**</option>

</select>

</td>

<td>**Direction:** </td>

<td>

<select id=**"direction"** onblur=fun() onchange=fun()>

<option value=**"->"**>**->**</option></option>

<option value=**"<>"**><></option>

</select>

</td>

</tr>

<tr>

<td>**Source IP:** </td>

<td><input type=**"text"** id=**"sip"** onblur=fun() placeholder=**"any"**></td>

<td>**Source Port:** </td>

<td><input type=**"text"** id=**"sport"** onblur=fun() placeholder=**"any"**></td>

</tr>

<tr>

<td>**Destination IP:** </td>

<td><input type=**"text"** id=**"dip"** onblur=fun() placeholder=**"any"**></td>

<td>**Destination Port:** </td>

<td><input type=**"text"** id=**"dport"** onblur=fun() placeholder=**"any"**></td>

</tr>

<tr height=**"20"**><td></td></tr>

<tr>

<td><h2>**Rule Options**</h2></td>

</tr>

<tr>

<td>**Message:**</td>

<td><input type=**"text"** id=**"msg"** onblur=fun() placeholder=**"Rule message"**></td>

</tr>

<tr>

<td>**Signature ID:** </td>

<td><input type=**"text"** id=**"sid"** onblur=fun() placeholder=**"sid"**></td>

<td>**Revision:** </td>

<td><input type=**"text"** id=**"rev"** onblur=fun() placeholder=**"rev"**></td>

</tr>

<tr>

<td>**Class Type:** </td>

<td><input type=**"text"** id=**"classtype"** onblur=fun() placeholder=**"Class-Type"**></td>

<td>**Priority:** </td>

<td>

<select id=**"pri"** onblur=fun() onchange=fun()>

<option value=**""**></option>

<option value=**"1"**>**1**</option>

<option value=**"2"**>**2**</option>

<option value=**"3"**>**3**</option>

<option value=**"4"**>**4**</option>

<option value=**"5"**>**5**</option>

</select>

</td>

</tr>

<tr>

<td>**Content:** </td>

<td><input type=**"text"** id=**"content"** onblur=fun() placeholder=**"Content match"**></td>

<td><input type=**"text"** id=**"offset"** onblur=fun() placeholder=**"Offset"**></td>

<td><input type=**"text"** id=**"depth"** onblur=fun() placeholder=**"Depth"**></td>

</tr>

<tr>

<td>**Options for content:** </td>

<td><font style=**"font-size:20px;"**>**nocase**<input type=**"checkbox"** id=**"nocase"** onchange=fun() class=**"largerCheckbox"**></td>

<td><font style=**"font-size:20px;"**>**uri**<input type=**"checkbox"** id=**"uri"** onchange=fun() class=**"largerCheckbox"**></td>

<td><font style=**"font-size:20px;"**>**not**<input type=**"checkbox"** id=**"not"** onchange=fun() class=**"largerCheckbox"**></td>

</tr>

</table>

<table align=**"center"**>

<tr height=**"20"**><td></td></tr>

<tr>

<td><textarea id=**"view"** name=**"view"** style=**"font-size: 20px; text-align:center;"** rows=**"4"** cols=**"100"**></textarea></td>

</tr>

<tr align=**"center"**>

<td><input type=**"button"** style=**"font-size: 20px;"** value=**"Copy rule to Clipboard"** onclick=save()></td>

</tr>

</table>

</form>

</body>