ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

A blue logo with a black background

Description automatically generated with low confidence

ĐỒ ÁN   
**AN TOÀN MẠNG NÂNG CAO**

**Đề tài:**  
**Security Onion**

Giảng viên hướng dẫn: ThS. Nguyễn Duy

**Nhóm sinh viên thực hiện:**  
20521317 Hoàng Đình Hiếu

20521429 Phan Quang Khải

20521439 Phan Vĩnh Khang

Tp. Hồ Chí Minh, năm 2023

**Mục lục**

**CHƯƠNG 1: GIỚI THIỆU SECURITY ONION**

* 1. Giới thiệu......................................................................................3
  2. Mô hình hoạt động.......................................................................3
  3. Cài đặt...........................................................................................4
  4. Kiến trúc........................................................................................4
     1. Evaluation……………………………………………………………………….5
     2. Standalone……………………………………………………………………..5
     3. Distributed……………………………………………………………………..5
     4. Import…………………………………………………………………………….5

**CHƯƠNG 2: CÁC CHỨC NĂNG CHÍNH**

2.1 Tổng quan giao diện...........................................................................6

2.2 Thành phần…………………………………………………………………………………….6

2.2.1 Alearts..................................................................................7

2.2.3 Hunt......................................................................................7

2.2.4 Cases.....................................................................................8

2.2.5 PCAP.....................................................................................10

**CHƯƠNG 3: DEMO**

3.1 Mô hình xây dựng………………………………………………………………………….10

3.1.1 Window……………………………………………………………………………11

3.1.2 Linux…………………………………………………………………………………12

3.2 Threat detection…………………………………………………………………………….12

3.3 Local rules……………………………………………………………………………………….14

# **CHƯƠNG 1: GIỚI THIỆU SECURITY ONION**

* 1. **Giới thiệu**

Security Onion (SO) là phần mềm mã nguồn mở, miễn phí được sử dụng rộng rãi cho Network Security Monitoring (NSM) and Enterprise Security Monitoring (ESM).

NSM: giám sát hoạt động mạng và các sự kiện bảo mật mạng liên quan.

ESM: cung cấp thông tin chi tiết về các hoạt động đó bao gồm các điểm đầu cuối, các thiết bị truy cập từ xa,…

* 1. **Mô hình hoạt động**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, thiết kế

Mô tả được tạo tự động

Quan sát sơ đồ ở trên, ta thấy SO trong môi trường mạng doanh nghiệp bao gồm firewall, workstation và server. Người quản trị mạng sử dụng SO để theo lưu lượng truy cập mạng cả ở phía bên ngoài và bên trong trung tâm dữ liệu, kết quả theo dõi ở workstation và server được lưu dưới dạng logs và gửi đến SO để phân tích.

* 1. **Cài đặt**

Yêu cầu phần cứng tối thiểu.

* Nếu chỉ sử dụng đế import và phân tích file pcap, cài đặt SO như một import note:

+ 4GB RAM.

+ CPU 2 core.

+ 100GB bộ nhớ.

* Nếu sử dụng đầy đủ chức năng, full tất cả các case:

+ 12GB RAM

+ CPU 4 core

+ 200 GB bộ nhớ.

Môi trường cài đặt:

* Cài đặt bằng file .iso tại trang chủ (<https://github.com/Security-Onion-Solutions/securityonion/blob/master/VERIFY_ISO.md>)
* Sử dụng máy ảo VMware hoặc Virtualbox, host là CentOS 7-64 bit.

Các bước cài đặt: nhóm em cài cà đặt theo các bước ở đây <https://docs.securityonion.net/en/2.3/first-time-users.html>

* 1. **Kiến trúc**

SO có nhiều cách triển khai tùy vào quy mô và mục đích sử dụng, người ta chia thành các kiến trúc sau:

* Eval
* Standalone
* Distributed
* Import

Đây là hình chọn kiểu kiến trúc trong lúc cài đặt:

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, Phông chữ

Mô tả được tạo tự động

(nhóm em cài đặt bản STANDALONE)

* **Import**

Đây là kiến trúc đơn giản nhất, chỉ có các chức năng để phân tích các **file pcap (dùng so-import-pcap)** và **evtx (so-import-evtx).**

* **Evaluation**

Đây là phiên bản đầy đủ, có giao diện web, theo dõi được các luồng mạng ở nhiều interface khác nhau và tạo logs. Dùng Filebeat (<https://docs.securityonion.net/en/2.3/filebeat.html#filebeat>)

để thu thập log và gửi đến Elasticsearch

(https://docs.securityonion.net/en/2.3/elasticsearch.html #elasticsearch)

để phân tích ,tuy nhiên kiến trúc này được dùng khi muốn cài đặt nhanh và test các chức năng của SO, nó không được thiết kế cho mục đích production.

* **Standalone**

Tương tự như Evaluation với đầy đủ chức năng nhưng thay vì dùng gửi log đến Elasticsearch nó gửi đến Logstash (https://docs.securityonion.net/en/2.3/logstash.html#logstash)

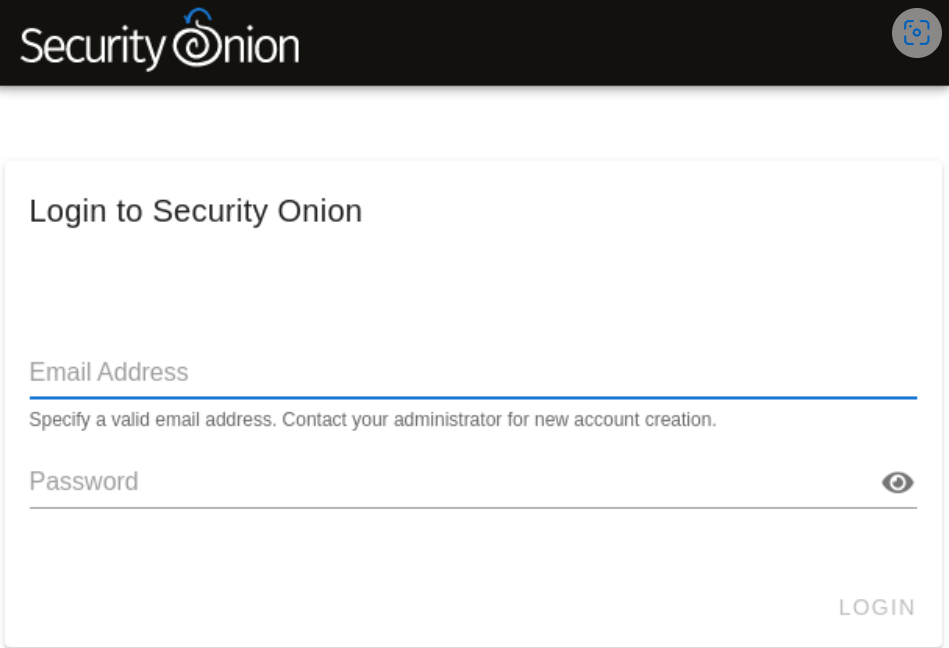
* **Distributed**

Triển khai phân tán tiêu chuẩn bao gồm một manager node, một hoặc nhiều forward nodes chạy các thành phần cảm biến mạng và một hoặc nhiều search nodes chạy các thành phần Tìm kiếm đàn hồi. Kiến trúc này có thể trả trước nhiều hơn, nhưng nó cung cấp khả năng mở rộng và hiệu suất cao hơn, vì người quản trị có thể chỉ cần thêm nhiều nút hơn để xử lý nhiều nguồn lưu lượng truy cập hoặc nhật ký hơn.

# **CHƯƠNG 2: CÁC CHỨC NĂNG CHÍNH**

**2.1 Tổng quan giao diện**

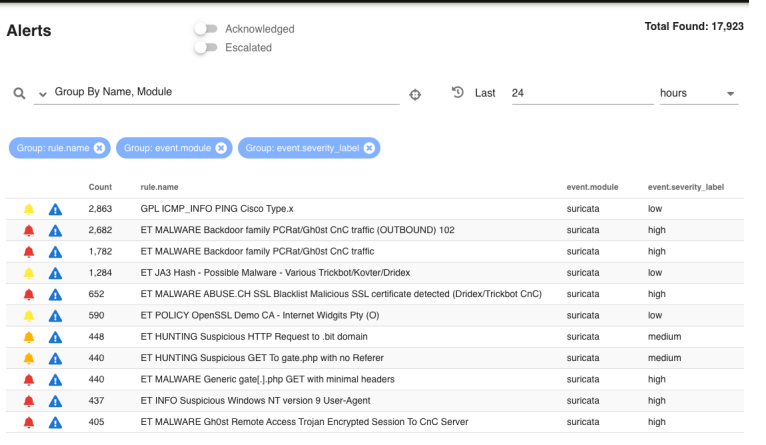
Khi cài đặt xong ta sẽ vô trang login của SO:



Sau khi login ta có giao diện web với các chức năng chính như hình dưới:

**2.2 Thành phần**

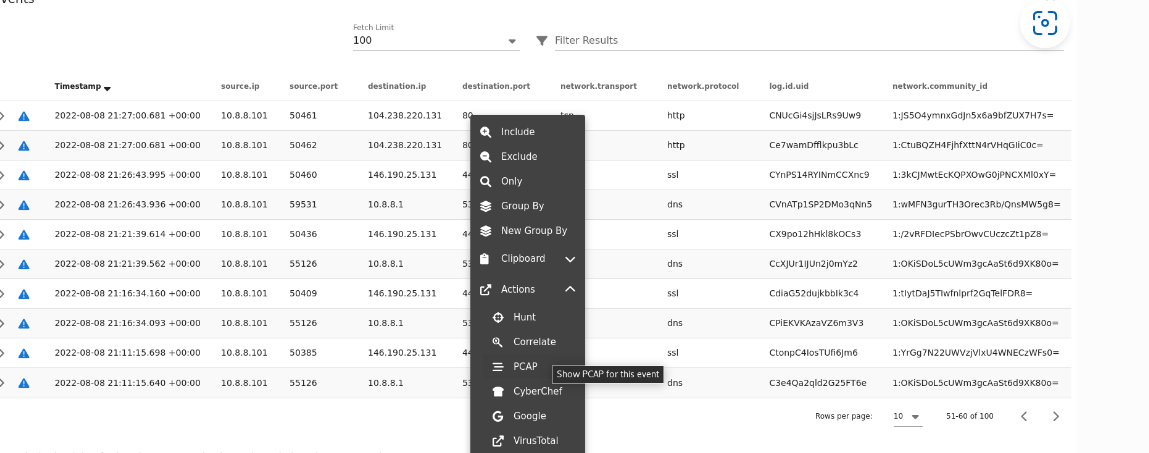
**2.2.1 Alerts**



Giao diện này cho ta xem các cảnh báo về bảo mật mà SO tạo ra và thông tin chi tiết về các cảnh báo đó đó.

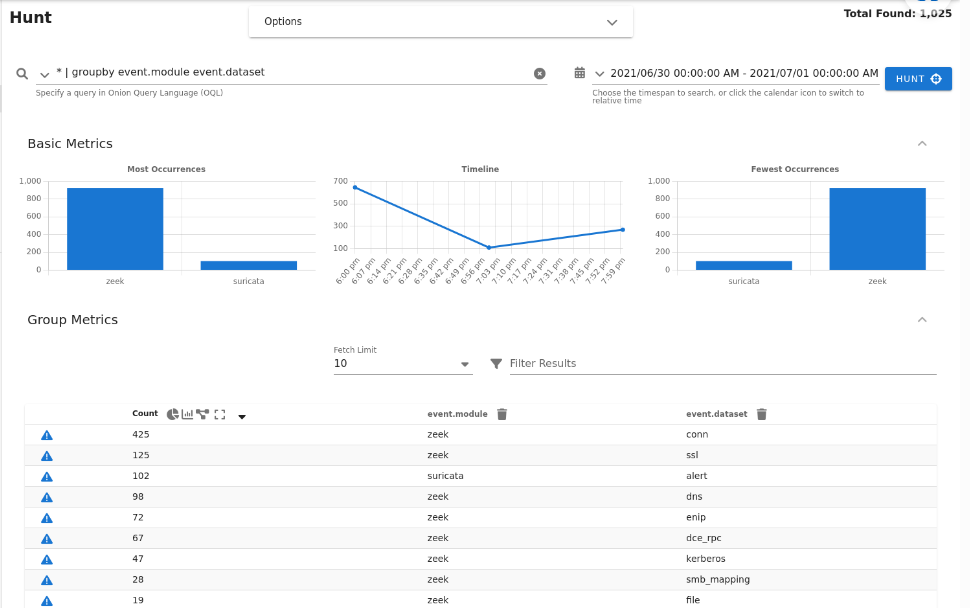
Để ý tên cảnh báo là **rule.name**, đây là tên được đặt bởi một luật (rule) mà người viết dùng để phát hiện các sự kiện bảo mật nhờ thu thập thông tin và từ các log và tổng hợp lại. Chi tiết về cách viết rule sẽ được đề cập ở phần sau, ngoài ra còn **event.module** là các loại module dùng để thu thập log và cái còn lại mức độ của sự kiện bảo mật đó.

**2.2.2 Hunt**



Hình ở trên là thông tin logs với các trường thông tin như: source.ip, source.port, network\_transport, network\_protocol,.... (tên đọc là biết nên không cần giải thích gì nhiều).

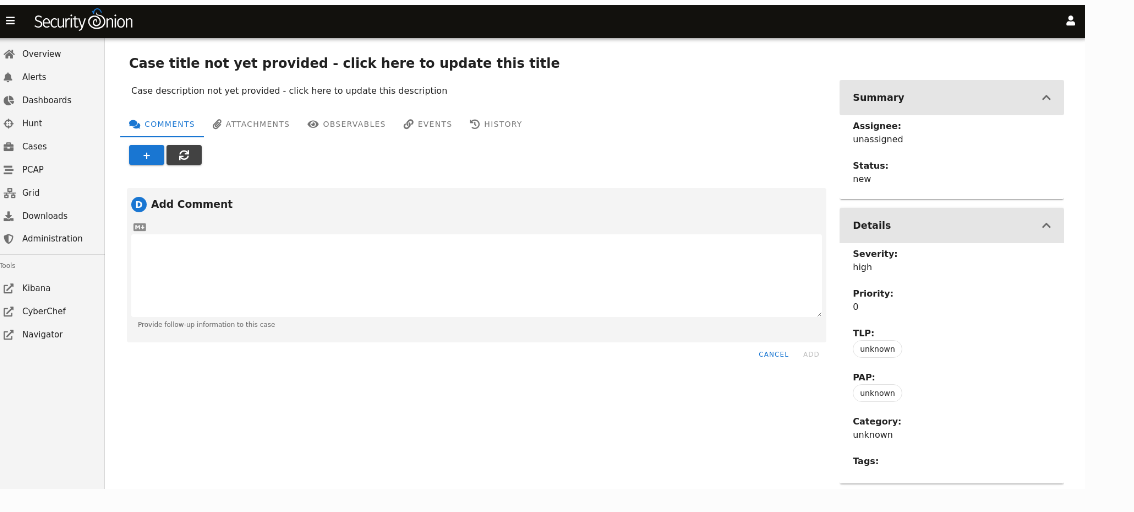
Giao diện này được dùng trong threat-hunting, sử dụng câu truy vấn **OQL(onion query language)** để lấy/lọc thông tin từ logs.



Như hình trên câu truy vấn: **\* | groupby event.module event.dataset** dùng để lọc thông tin **module** và **dataset** từ logs.

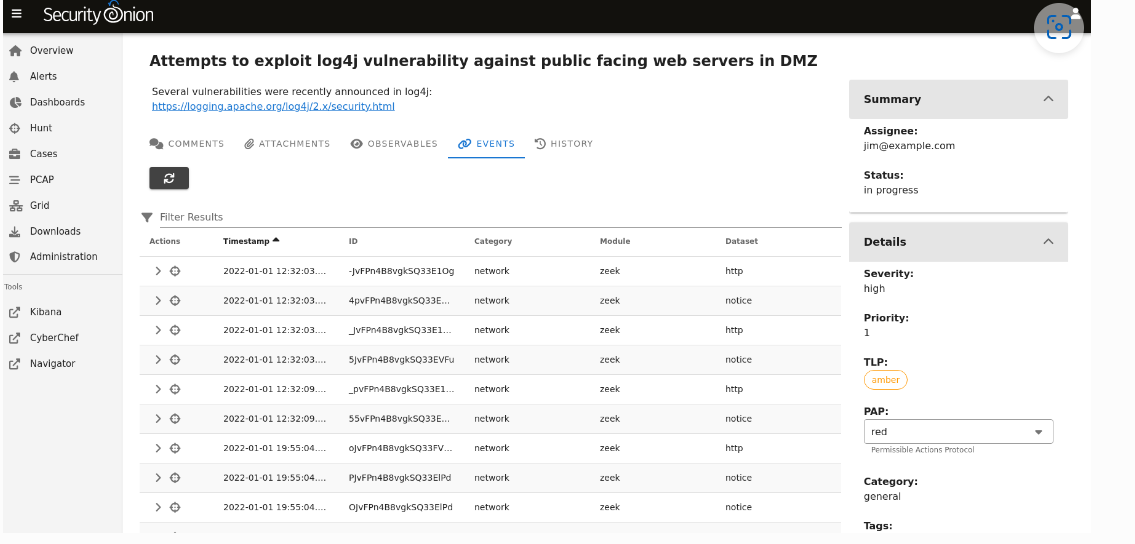
Một số từ khóa phổ biến dùng trong OQL có thể tham khảo ở đây: **https://docs.securityonion.net/en/2.3/dashboards.html#oql**

**2.2.3 Cases**



**Cases** giúp ta định nghĩa case mới dựa trên logs lấy từ **Alert** và **Hunt**, có nghĩa là khi thấy một log đặc biệt/đáng chú ý ta có thể thêm các thông tin cần thiết như tên trường hợp, mô tả, mức độ nguy hiểm, độ ưu tiên,...

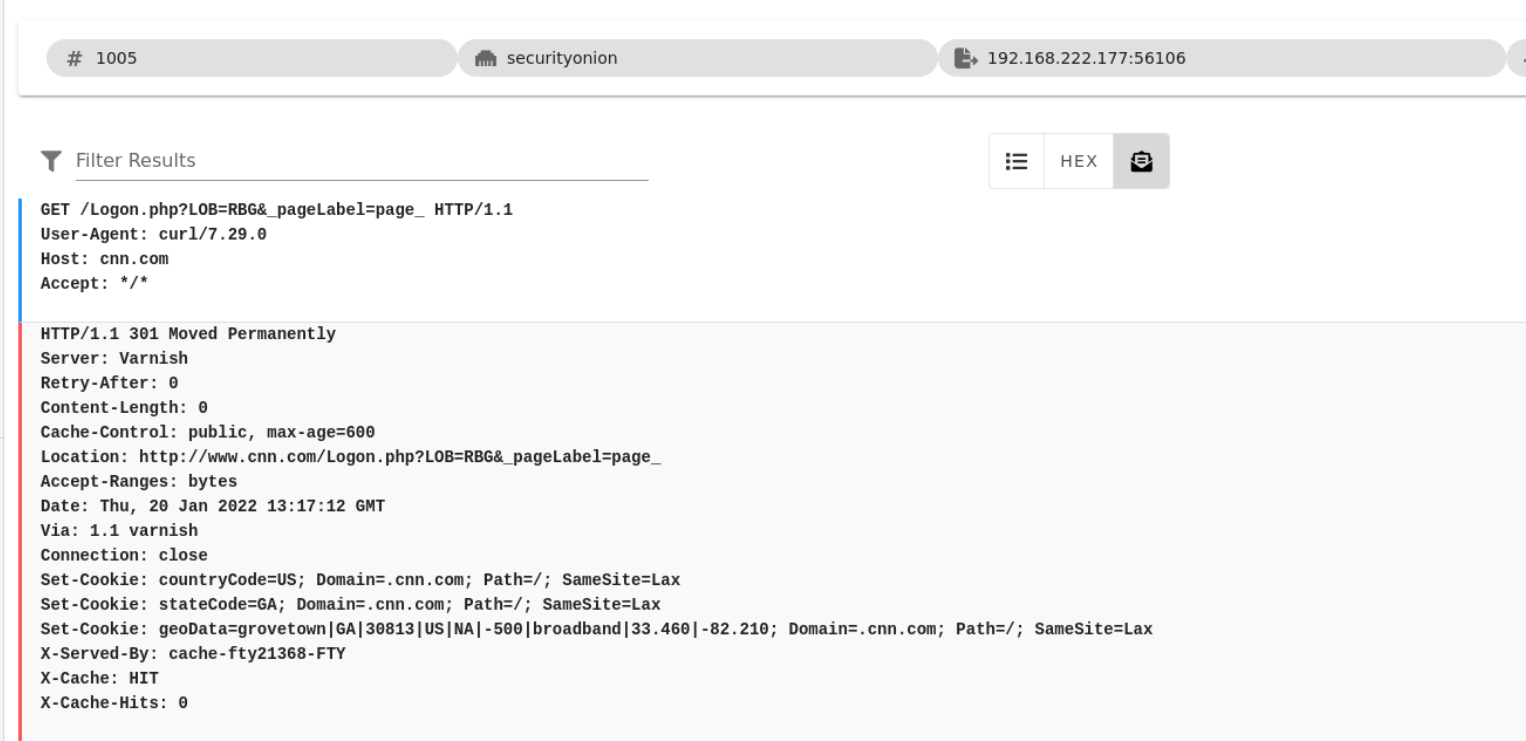
Lúc đó khi log này xuất hiện dựa vào mức độ ưu tiên và độ nguy hiểm log sẽ được đẩy lên đầu (dựa theo bộ lọc) để ta dễ quan sát, tên và mô tả giúp ta biết được log đó có liên quan đến sự kiện bảo mật nào,....



Hình trên là ví dụ ta thu thập được các log có liên quan đế sự kiện khai thác lỗi log4j ở web server.

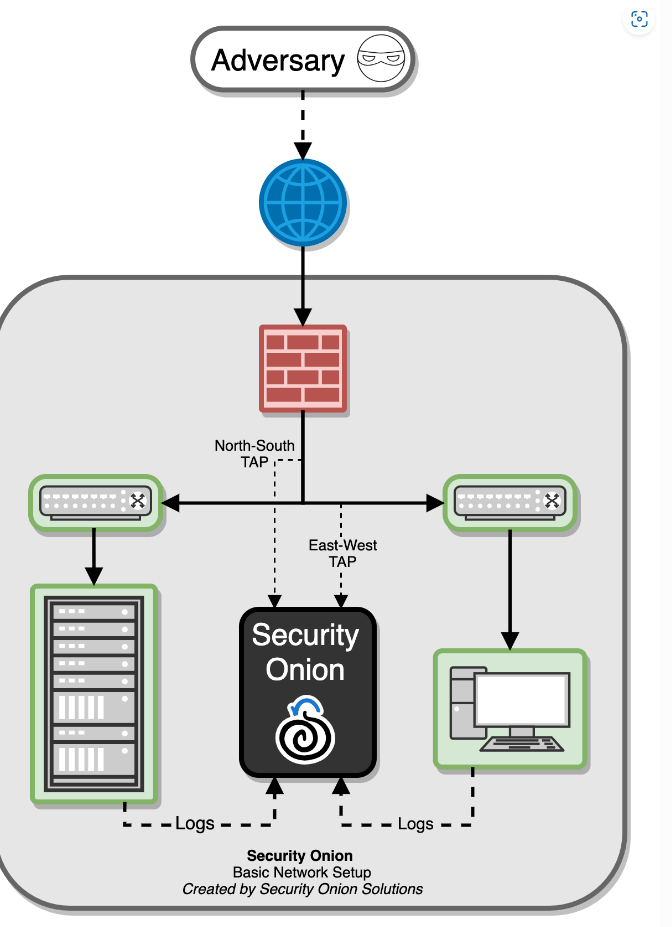
**2.2.4 PCAP**

Cho phép truy cập full packet capture từ logs thông qua **Stenographer**



# **CHƯƠNG 3: DEMO**

**3.1 Mô hình xây dựng**

****

Chúng ta xây dựng dựa trên mô hình ở trên bao gồm **Window, Linux và Firewall**. Attacker gồm hai loại là **internal và external**, trong đó internal sẽ tấn công và Window và Linux còn external thì attacker sẽ tấn công vào Firewall. Tất cả sự kiện diễn ra trong suốt cuộc tấn công sẽ được gửi về Security Onion dưới dạng **logs**.

**3.1.1 Window**

Ta dùng **Winlogbeat** (<https://docs.securityonion.net/en/2.3/beats.html#beats>) để gửi logs từ Window đến SO.

Do sử dụng **Winlogbeat** chỉ giúp thu nhập **window\_eventlog (**tham khảo **)** nên ta dùng thêm **Sysmon** (<https://docs.securityonion.net/en/2.3/sysmon.html#sysmon>) là một window service giúp thông nhập các thông tin chi tiết về process creation, network connections và file creation time.

Kết quả sau khi cài đặt thành công:

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

**3.1.2 Linux**

Cũng dùng **sysmon** như window và sử dụng thêm **suricata** logs.

Suricata: dùng để phát hiện các mối đe dọa liên quan đến mạng ([**https://docs.securityonion.net/en/2.3/suricata.html?highlight=suricata**](https://docs.securityonion.net/en/2.3/suricata.html?highlight=suricata)).

Ship logs: dùng **filebeat (**[**https://docs.securityonion.net/en/2.3/filebeat.html**](https://docs.securityonion.net/en/2.3/filebeat.html)**)**

**3.2 Threat detection**

Sau khi **setup** xong các thứ thì ta sẽ bắt đầu demo bằng cách viết **rule** để phát hiện một số trường hợp đáng nghi (có thể nguy hiểm) bằng tính năng **Playbook** của **Security Onion:**

Ảnh có chứa văn bản, ảnh chụp màn hình

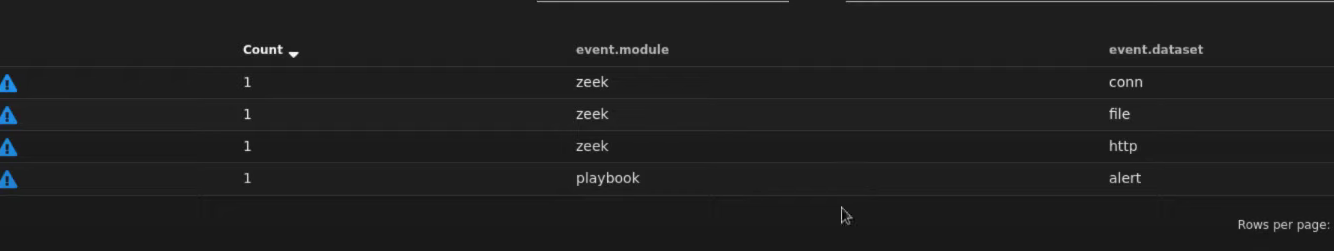
Mô tả được tạo tự động

Do rules được viết dựa trên **Sigma rules** nên có thể tham khảo ở (<https://github.com/SigmaHQ/sigma>) để học về cách viết, cách viết, các ví dụ tham khảo,…

Ở hình trên là rule để detect việc sử dụng lệnh **whoami** trên **terminal**, do quản trị viên thường không sử dụng lệnh này nên bất kì ai ngoài quản trị viên sử dụng đều đáng nghi, có thể là attacker.

Rule trên lấy tất cả các log về **process\_creation** từ **sysmon** và lọc ra trong đó cái nào có **filename=”whoami.exe”,** nếu có thì phát cảnh báo là **Whoami Execution** trên **Security Onion.**

Lúc đó tên loại log đó là Playbook luôn:



**3.3 Local rules**

Viết rule không phải mặc định là Sigma ta có thể thêm nhiều loại rule khác như **yara, suricata, snort**,… Tuy nhiên Security Onion không hỗ trợ các loại này nên ta cần phải add thủ công vào file **local.rules** của nó:

Ví dụ như đối với **suricata**:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Tới đây thì chúng ta có thể tự tạo rule của riêng mình để phát hiện và phân tích các mối đe dọa của mình, chúc mọi người thành công.