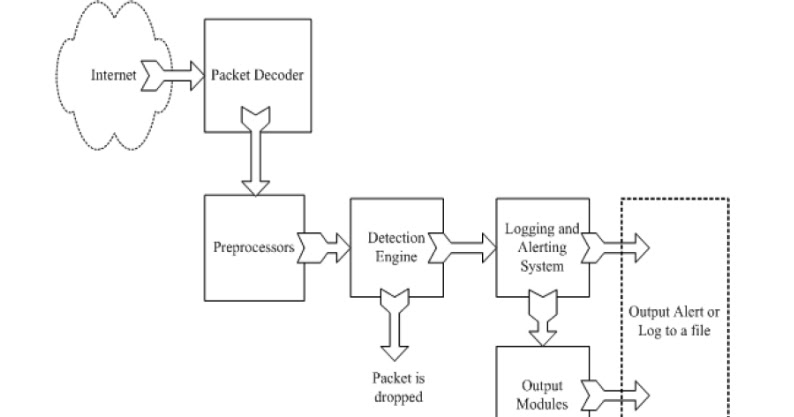
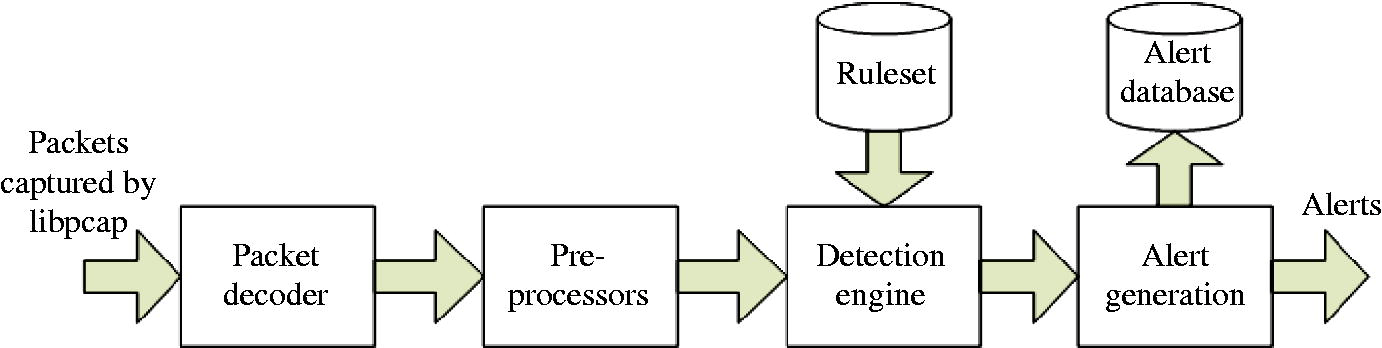
**Snort là gì?**

Snort là một kiểu IDS/IPS, thực hiện giám sát các gói tin ra vào hệ thống.

* Snort là một mã nguồn mở miễn phí với nhiều tính năng trong việc bảo vệ hệ thống bên trong, phát hiện sự tấn công từ bên ngoài vào hệ thống.
* Snort được viết bởi Martin Roesch vào năm 1998. Hiện tại, Snort được phát triển bởi Sourcefire, nơi mà Roesch đang là người sáng lập và CTO, và được sở hữu bởi Cisco từ năm 2013.

**Mô hình kiến trúc của Snort**

[](https://github.com/hocchudong/ghichep-IDS-IPS-SIEM/blob/master/Images/kien-truc-snort.jpg)

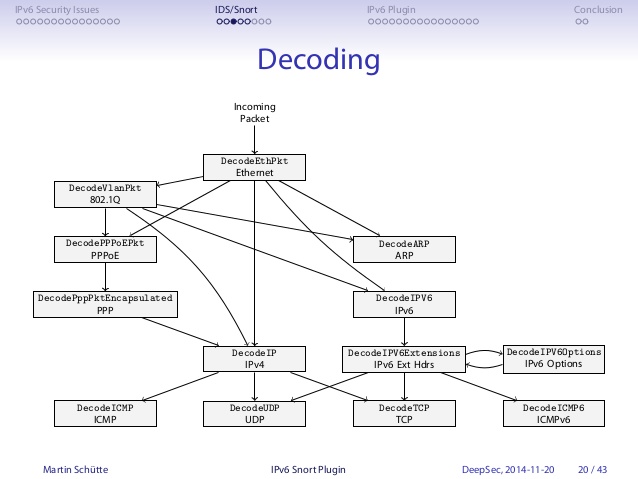
[](https://github.com/hocchudong/ghichep-IDS-IPS-SIEM/blob/master/Images/snort-architecture.png)

Trong mô hình kiến trúc trên, hệ thống Snort được chia thành 4 phần:

* Module Decoder: Xử lý giải mã các gói tin
* Module Preprocessors: Tiền xử lý
* Module Detection Engine: Phát hiện
* Module Logging and Alerting System: Lưu log và cảnh báo

**Module Decoder**

Snort sử dụng thư viện pcap để bắt mọi gói tin trên mạng lưu thông qua hệ thống. Gói tin sau khi được giải mã sẽ đưa vào Module tiền xử lý.

[](https://github.com/hocchudong/ghichep-IDS-IPS-SIEM/blob/master/Images/the-ipv6-snort-plugin-at-deepsec-2014-20-638.jpg)

**Module Preprocessors**

Module này rất quan trọng đối với bất kỳ hệ thống nào để có thể chuẩn bị gói dữ liệu đưa vào cho Module phát hiện phân tích. 03 nhiệm vụ chính

* Kết hợp lại các gói tin: Khi một dữ liệu lớn được gửi đi, thông tin sẽ không đóng gói toàn bộ vào một gói tin mà thực hiện phân mảnh, chia thành nhiều gói tin rồi mới gửi đi. Khi Snort nhận được các gói tin này, nó phải thực hiện kết nối lại để có gói tin ban đầu. Module tiền xử lý giúp Snort có thể hiểu được các phiên làm việc khác nhau.
* Giải mã và chuẩn hóa giao thức (decode/normalize): cCông việc phát hiện xâm nhập dựa trên dấu hiệu nhận dạng nhiều khi thất bại khi kiểm tra các giao thức có dữ liệu có thể được biểu diễn dưới nhiều dạng khác nhau. Ví dụ: một Web server có thể nhận nhiều dạng URL: URL viết dưới dạng hexa/unicode hay URL chấp nhận dấu / hay . Nếu Snort chỉ thực hiện đơn thuần việc so sánh dữ liệu với dấu hiệu nhận dạng sẽ xảy ra tình trạng bỏ sót hành vi xâm nhập. Do vậy, 1 số Module tiền xử lý của Snort phải có nhiệm vụ giải mã và chỉnh sửa, sắp xếp lại các thông tin đầu vào.
* Phát hiện các xâm nhập bất thường (nonrule/anormal):các plugin dạng này thường để xử lý với các xâm nhập không thể hoặc rất khó phát hiện bằng các luật thông thường. Phiển bản hiện tại của Snort có đi kèm 2 plugin giúp phát hiện xâm nhập bất thường đó là portscan và bo (backoffice). Portscan dùng để đưa ra cảnh báo khi kẻ tấn công thực hiện quét cổng để tìm lỗ hổng. Bo dùng để đưa ra cảnh báo khi hệ thống nhiễm trojan backoffice.

**Module Detection Engine**

Đây là module quan trọng nhất của Snort. Nó chịu trách nhiệm phát hiện các dấu hiệu xâm nhập. Module sử dụng các luật (rule) được định nghĩa từ trước để so sánh với dữ liệu thu thập được, từ đó xác định xem có xâm nhập xảy ra hay không.

Một vấn đề quan trọng đối với module detection engine là thời gian xử lý gói tin: một IDS thường nhận rất nhiều gói tin và bản thân nó cũng có rất nhiều luật xử lý. Khi lưu lượng mạng quá lớn thì có thể xảy ra việc bỏ sót hoặc không phản hồi đúng lúc. Khả năng xử lý của module phát hiện phụ thuộc vào nhiều yếu tố: số lượng các luật, tốc độ hệ thống, băng thông mạng

Module detection engine có khả năng tách các phần của gói tin ra và áp dụng luật lên tưng phần của gói tin:

* IP header
* Header ở tầng transport: TCP, UDP
* Header ở tầng application: DNS, HTTP, FTP,...
* Phần tải (payload) của gói tin

Do các luật trong Snort được đánh số thứ tự ưu tiên nên một gói tin khi bị phát hiện bởi nhiều luật khác nhau, cảnh báo được đưa ra theo luật có mức ưu tiên cao nhất.

**Module log và cảnh báo**

Tùy thuộc vào module phát hiện (detection engine) có nhận dạng được xâm nhập hay ko mà gói tin có thể bị ghi log hay đưa ra cảnh báo. Các file log là các file dữ liệu có thể ghi dưới nhiều định dạng khác nhau như tcpdump, xml, syslog, log file.

**Bộ luật của Snort**

![rule\_snort]../Images/rule\_snort.png)

Module detection engine sử dụng các bộ luật để nhận dạng dữ liệu. ví dụ một luật

alert tcp 192.168.0.0/22 23 -> any any (content:”confidential”; msg: “Detected confidential”)

Cấu trúc của một rule được chia thành 02 phần: |Rule header|Rule Option|

* Phần Header: Chứa thông tin về hành động mà luật đó sẽ thực hiện khi phát hiện ra có xâm nhập nằm trong gói tin và nó cũng chứa tiêu chuẩn để áp dụng luật với gói tin đó.
* Phần Option: Chứa thông điệp cảnh báo và các thông tin về các phần của gói tin dùng để tạo nên cảnh báo. Phần Option này chưa các tiêu chuẩn phụ thêm để đối sánh với gói tin

**Cấu trúc phần header**

|Action|Protocol|Address|port|Direction|Address|Port|

Ví dụ:

|Alert| TCP| 192.168.0.0/22| 23| ->| Any| Any|

Giải thích:

* Action: là phần quy định loại hành động nào được thực thi. Thông thường các hành động tạo ra một cảnh báo hoặc log thông điệp, hay kích hoạt một luật khác
* Protocol: một giao thức cụ thể như tcp, udp, icmp, ip
* Address: địa chỉ nguồn hoặc địa chỉ đích (tùy thuộc vào hướng của gói tin: direction)
* Port: Xác định các cổng nguồn hoặc cổng đích của một gói tin
* Direction: Phần này sẽ chỉ ra địa chỉ nguồn và địa chỉ đích

**Action**

Trong Snort, có 05 action được định nghĩa:

* Pass: Cho phép Snort bỏ qua gói tin này
* Log: dùng để log gói tin. Có thể log vào file hay CSDL
* Alert: Gửi thông điệp cảnh báo khi có dấu hiệu xâm nhập được phát hiện
* Activate: Tạo ra cảnh báo và kích hoạt thêm các luật khác để kiểm tra thêm điều kiện của gói tin
* Dynamic: Đây là luật được gọi bởi các luật khác có Action khai báo là Activate

**Protocol**

Chỉ ra các loại gói tin mà luật được áp dụng:

* IP
* ICMP
* TCP
* UDP

Nếu là IP thì Snort sẽ kiểm tra header của lớp liên kết để xác định loại gói tin. Nếu bất kỳ giao thức nào khác, Snort sẽ sử dụng header là IP để xác định loại giao thức.

**Address**

Có 02 phần là địa chỉ đích và địa chỉ nguồn. Nó có thể là một IP đơn hoặc 1 dải mạng. Nếu là "any" thì áp dụng cho tất cả các địa chỉ trong mạng. Chú ý: Nếu là một host thì có dạng: ipaddress/32. Ví dụ: 192.168.0.100/32

Snort cung cấp phương pháp để loại trừ địa chỉ IP bằng cách sử dụng dấu "!". Ví dụ:

alert icmp ![192.168.0.0/22] any -> any any (msg: “Ping with TTL=100”; ttl: 100;)

*Lưu ý*: dấu “[]” chỉ cần dùng khi đằng trước có “!”

**Port**

Số port để áp dụng cho các luật. Ví dụ telnet 23, DNS 53,... Port chỉ áp dụng cho 02 giao thức là TCP và UDP

Để sử dụng một dãy các port thì ta phân biệt bởi dấu ":". Ví dụ

alert udp any 1024:8080 -> any any (msg: “UDP port”;)

**Direction**

Chỉ ra đâu là nguồn, đâu là đích. có thể là -> hoặc <- hoặc <>

Trường hợp <> là khi ta muốn kiểm tra Client và Server.

**Phần Option**

Phần Option nằm ngay sau phần Header và được bao bọc trong dấu ngoặc đơn. Nếu có nhiều Option thì sẽ phân biệt bởi dấu chấm phẩy ";". Một Option gồm có 2 phần: một là từ khóa và một là tham số. 02 phần này sẽ phân cách nhau bằng dấu hai chấm ":"

**Từ khóa ACK**

Trong header TCP có chứa trường Acknowledgement Number với độ dài 32 bit. Trường này chỉ ra số thứ tự tiếp theo gói tin TCP của bên gửi đang chờ để nhận. Trường này chỉ có ý nghĩa khi mà cờ ACK được thiết lập.

Ví dụ: gửi gói tin TCP tới cổng 80 với cờ ACK được bật và số thứ tự là 0, bên nhận sẽ thấy gói tin không hợp lệ và gửi lại gói tin RST. Khi nhận được gói RST này, công cụ ping sẽ biết được IP này đang tồn tại hay không.

Để kiểm tra loại ping TCP này thì ta có thể dùng luật sau:

Alert tcp any any -> 192.168.0.0/22 any (flags: A; ack: 0; msg: “TCP ping detected”)

**Từ khóa classtype**

Các luật có thể được phân loại và gán cho một số chỉ độ ưu tiên nào đó để nhóm và phân biệt chúng với nhau. Để hiểu rõ hơn về classtype thì ta cần hiểu được file classification.config. Mỗi dòng trong file này đều có cấu trúc như sau: Config classification: name, description, priority

Trong đó:

* Name: tên dùng để phân loại, tên này sẽ được dùng với từ khóa classtype trong các luật Snort.
* Description: Mô tả
* Priority: là một chỉ số chỉ độ ưu tiên mặc định của lớp này. Độ ưu tiên này có thể được điều chỉnh trong từ khóa priority của phần Option trong Snort

Ví dụ:

Config classification: DoS, Denied of Service Attack, 2

Và luật:

Alert udp any any -> 192.168.0.0/22 6838 (msg:”DoS”; content: “server”; classtype: DoS; priority: 1;)

để ghi đè lên giá trị priority mặc định của lớp đã định nghĩa.

**Từ khóa content**

Một đặc tính quan trọng của Snort là có khả năng tìm một mẫu dữ liệu bên trọng một gói tin.

Ví dụ:

alert tcp 192.168.0.0/22 any -> ![192.168.0.0/22] any (content: “GET”; msg :”GET match”;)

Luật trên tìm mẫu "GET" trong phần dữ liệu của tất cả gói tin TCP có nguồn mạng là 192.168.0.0/22 đi đến các địa chỉ đích không nằm trong dải mạng đó.

Tuy nhiên khi sử dụng từ khóa content cần nhớ rằng:

* Đối chiếu nội dung cần phải xử lý rất lớn nên ta phải cân nhắc kỹ khi sử dụng nhiều luật đối chiếu nội dung.

Các từ khóa được áp dụng cùng với content để bổ sung thêm các điều kiện là:

* Offset: dùng để xác định vị trí bắt đầu tìm kiếm là offset tính từ đầu phần dữ liệu của gói tin. Ví dụ:

alert tcp 192.168.0.0/22 any -> any any (content: “HTTP”; offset: 4; msg: “HTTP matched”;)

* Dept: dùng để xác định vị trí mà từ đó Snort sẽ dừng việc tìm kiếm. VD sau sẽ tìm 10 byte đầu tiên của content

alert tcp 192.168.0.0/22 any -> any any (content: “HTTP”; dept: 10; msg: “HTTP matched”;)

**Từ khóa dsize**

Dùng để đối sánh theo chiều dài của phần dữ liệu. Rất nhiều cuộc tấn công sử dụng lỗi tràn bộ nhớ đệm bằng cách gửi các gói tin có kích thước rất lớn. Ví dụ:

alert ip any any -> 192.168.0.0/22 any (dsize > 5000; msg: “Goi tin co kich thuoc lon”;)

**Từ khóa Flags**

Từ khóa này dùng để phát hiện xem những bit cờ flag nào được bật trong phần TCP header của gói tin. Mỗi cờ có thể được sử dụng như 1 tham số trong từ khóa flags

Flag| Kí hiệu tham số dùng trong luật của Snort| | FIN – Finish Flag| F SYN – Sync Flag |S RST – Reset Flag| R PSH – Push Flag |

P ACK – Acknowledge Flag| A URG – Urgent Flag| U Reversed Bit 1| 1 Reversed Bit 2| 2 No Flag set| 0

Ví dụ: luật sau đây sẽ phát hiện một hành động quét dùng gói tin SYN-FIN:

Alert tcp any any -> 192.168.0.0/22 any (flags: SF; msg: “SYNC-FIN flag detected”;)

**Từ khóa flagbits**

Phần IP header của gói tin chứa 03 bit dùng để chống phân mảnh và tổng hợp các gói tin IP. Các bit đó là:

* Reversed bit (RB): dùng để dành cho tương lai)
* Don't Fragment Bit (DF): nếu bit này được thiết lập tức là gói tin không bị phân mảnh
* More Fragments Bit (MF): nếu được thiết lập thì các phần khác của gói tin vẫn đang trên đường đi mà chưa tới đích. Nếu bit này không được thiết lập thì đây là phần cuối cùng của gói tin.

Ví dụ: luật sau sẽ phát hiện xem bit DF trong gói tin ICMP có được bật hay không:

alert icmp any any -> 192.168.0.0/22 any (fragbits: D; msg: “Don’t Fragment bit set”;)

**từ khóa itype**

Chỉ kiểu của gói tin ICMP. ví dụ trong trường hợp sau itype: 8 tức là một gói tin ICMP có kiểu echo request

alert icmp !$HOME\_NET any -> $HOME\_NET any (msg:"IDS152 - PING BSD";

content: "|08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17|"; itype: 8;

depth: 32;)

**Tham khảo**

* <http://baomatmang.net/snort-la-gi-cau-truc-cua-snort/>
* <https://www.rokasecurity.com/ids-vs-ips/>
* <https://viblo.asia/p/network-tim-hieu-co-che-cach-hoat-dong-cua-ids-phan-2-pDljMbe5RVZn>