



SOICT

PHÁT HIỆN XÂM NHẬP MẠNG

Giảng viên hướng dẫn: PSG.TS. Nguyễn Linh Giang

Nhóm sinh viên thực hiện:

- Mai Văn Đăng
- Trương Ngọc Mai
- Nguyễn Thanh Tân
- Nguyễn Khắc Quang

MSSV: 20225699

MSSV: 20225879

MSSV: 20225923

MSSV: 20225760

Planet Earth
Solar system
Milky Way Galaxy
Tentura, to the left of the
Old Georgy Danella's
Earthlings in the C-

PHÁT HIỆN XÂM NHẬP MẠNG

1. Thực trạng
2. Cơ sở lý thuyết
3. Phần mềm Snort
4. Ứng dụng
5. Chạy kiểm thử
6. Kết luận





Thực trạng hiện nay

01

Năm 2023 có 13.900 vụ tấn công mạng vào các tổ chức tại Việt Nam, trung bình mỗi tháng xảy ra 1160 vụ. 3 điểm yếu bị tấn công nhiều nhất con người (32,6%), lỗ hổng nền tảng (27,4%) và lỗ hổng website (25,3%).

02

Tấn công Twitter 2020 :khi tin tặc xâm chiếm các tài khoản nhằm lừa đảo người dùng gửi cho chúng bitcoin. 130 tài khoản là mục tiêu của cuộc tấn công. Ước tính đã chiếm đoạt hơn 120.000USD bằng hình thức này.

03

Vào năm 2013, Yahoo đã trải qua một trong những vụ tấn công mạng lớn nhất trong lịch sử internet. Khoảng 3 tỷ tài khoản của người dùng Yahoo đã bị chiếm đoạt thông tin cá nhân, bao gồm tên người dùng, địa chỉ email, mật khẩu đã được mã hóa, ngày tháng sinh, số điện thoại, và nhiều thông tin khác.



Cơ sở lý thuyết

01

An ninh mạng
và các mối đe
dọa

02

Phát hiện
xâm nhập mạng

03

Các thành phần

04

Các kĩ thuật và
công nghệ hỗ trợ

01

An ninh mạng và các mối đe dọa

- Là lĩnh vực quan trọng của bảo mật thông tin
- Giám sát các hoạt động mạng
- Chống lại các cuộc tấn công

Các mối đe dọa xâm nhập mạng

- Malware
- Các cuộc tấn công từ bên ngoài
- Các cuộc tấn công từ bên trong
- Social engineering
- Các lỗ hổng bảo mật



02

Phát hiện xâm nhập mạng

2.1 Hệ thống phát hiện xâm nhập mạng (IDS)

- Giám sát các luồng dữ liệu đang lưu thông trên mạng
- Phát hiện những hành động khả nghi hay xâm nhập trái phép
- Phân biệt những cuộc tấn công bên trong hay bên ngoài
- Chặn các cuộc tấn công (IPS)

2.2 Các phương pháp

- Dựa trên chữ ký
- Dựa trên hành vi
- Dựa trên dữ liệu bất thường
- Phối hợp các phương pháp trên



03

Các thành phần của hệ thống

1. Cảm biến (Sensors)
2. Bộ phân tích (Analysis Engine)
3. Cơ sở dữ liệu chữ ký (Signature Database)
4. Mô hình hành vi (Behavior Model)
5. Giao diện người dùng (User Interface)
6. Hệ thống cảnh báo (Alerting System)
7. Cơ sở dữ liệu (Database)



04

Các kĩ thuật và công nghệ hỗ trợ

1. Mã hóa dữ liệu (Encryption)
2. Firewalls (Tường lửa)
3. Proxy Servers
4. Network Intrusion Detection Systems (NIDS)
5. Host Intrusion Detection Systems (HIDS)
6. Thiết bị phát hiện xâm nhập (IDA)
7. Công nghệ Mô hình Học máy (Machine Learning Models)
8. Phân tích Giao thức (Protocol Analysis)
9. Cơ sở dữ liệu chữ ký (Signature Databases)
10. Công nghệ Xử lý sự kiện thời gian thực (Real-Time Event Processing)



PHẦN MỀM SNORT

01

Giới thiệu về
Snort

02

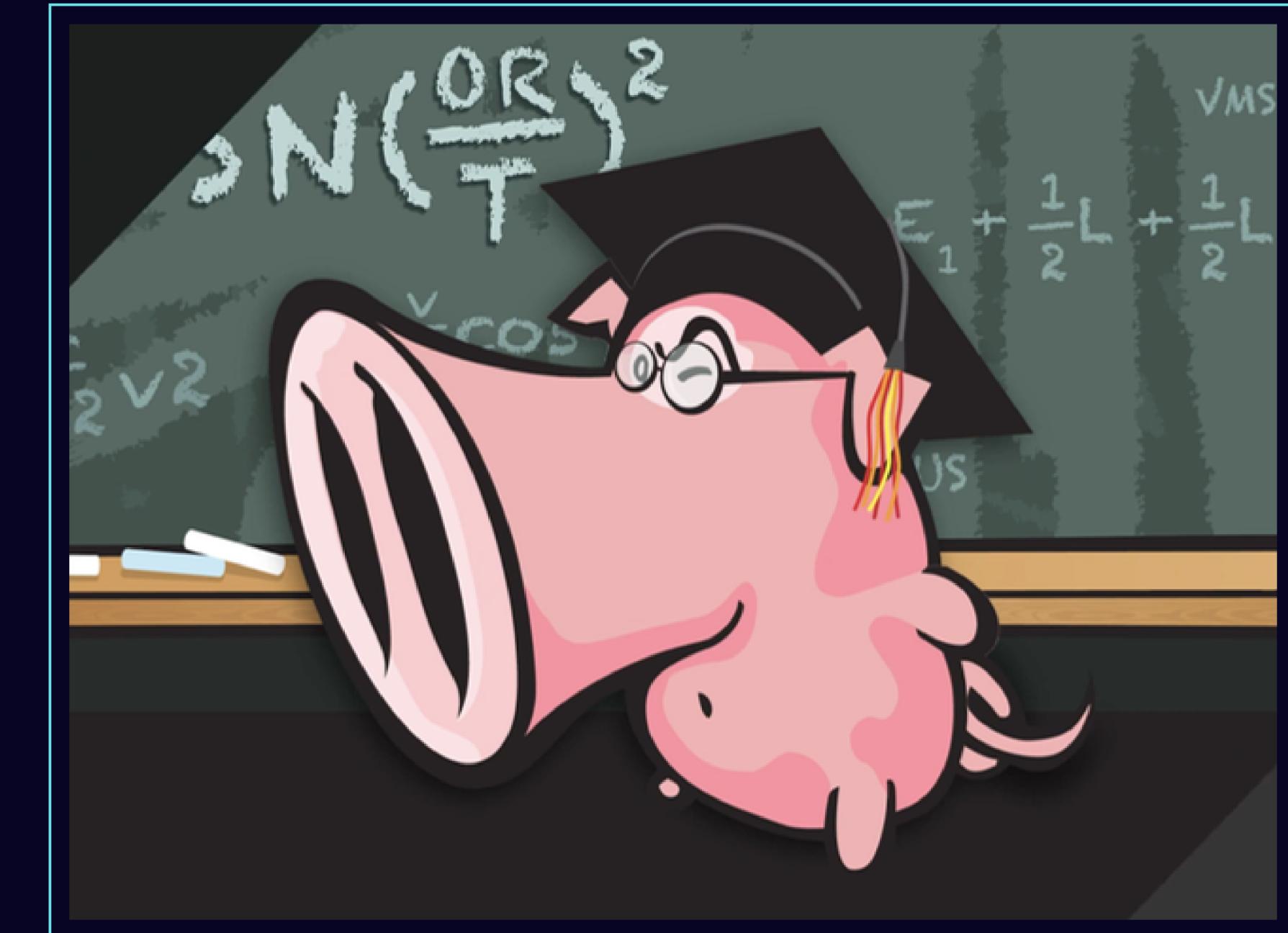
Kiến trúc của
Snort

03

Cấu trúc luật
của Snort

04

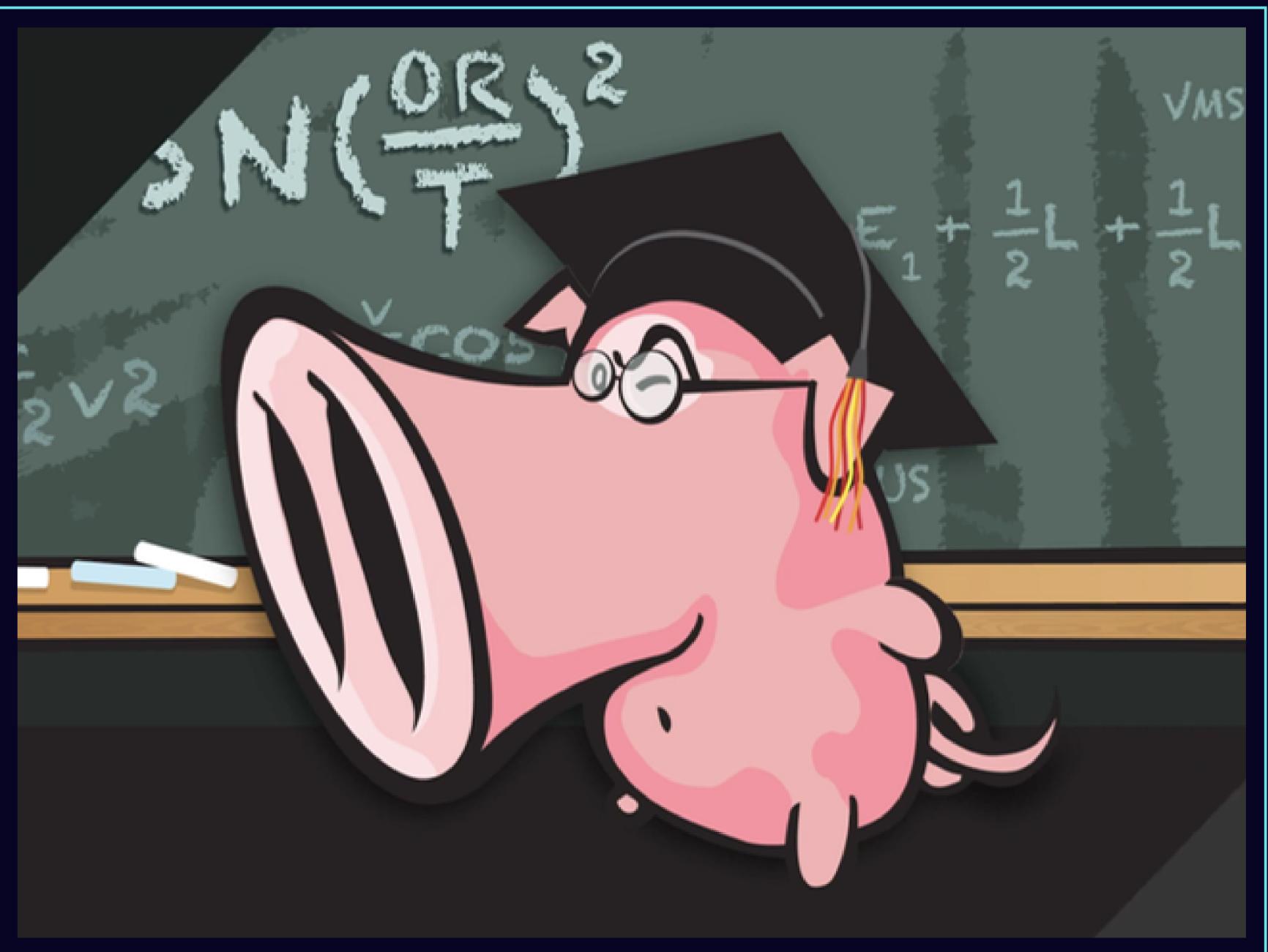
Kiểm tra và bổ
sung các quy tắc



01

Giới thiệu về Snort

- Mô hình mã nguồn mở
- Có rất nhiều tính năng
- Kiến trúc thiết kế theo kiểu module
- Cơ sở dữ liệu luật chứa 2930 luật và được cập nhật thường xuyên
- Có thể chạy trên nhiều hệ thống nền như Windows, Linux, OpenBSD, NetBSD, Solaris, MacOS...
- Có thể được cấu hình để chạy như một NIDS
- Hỗ trợ khả năng hoạt động trên các giao thức: Ethernet, Token Ring, FDDI, Cisco HDLC, SLIP, và PF của OpenBSD.



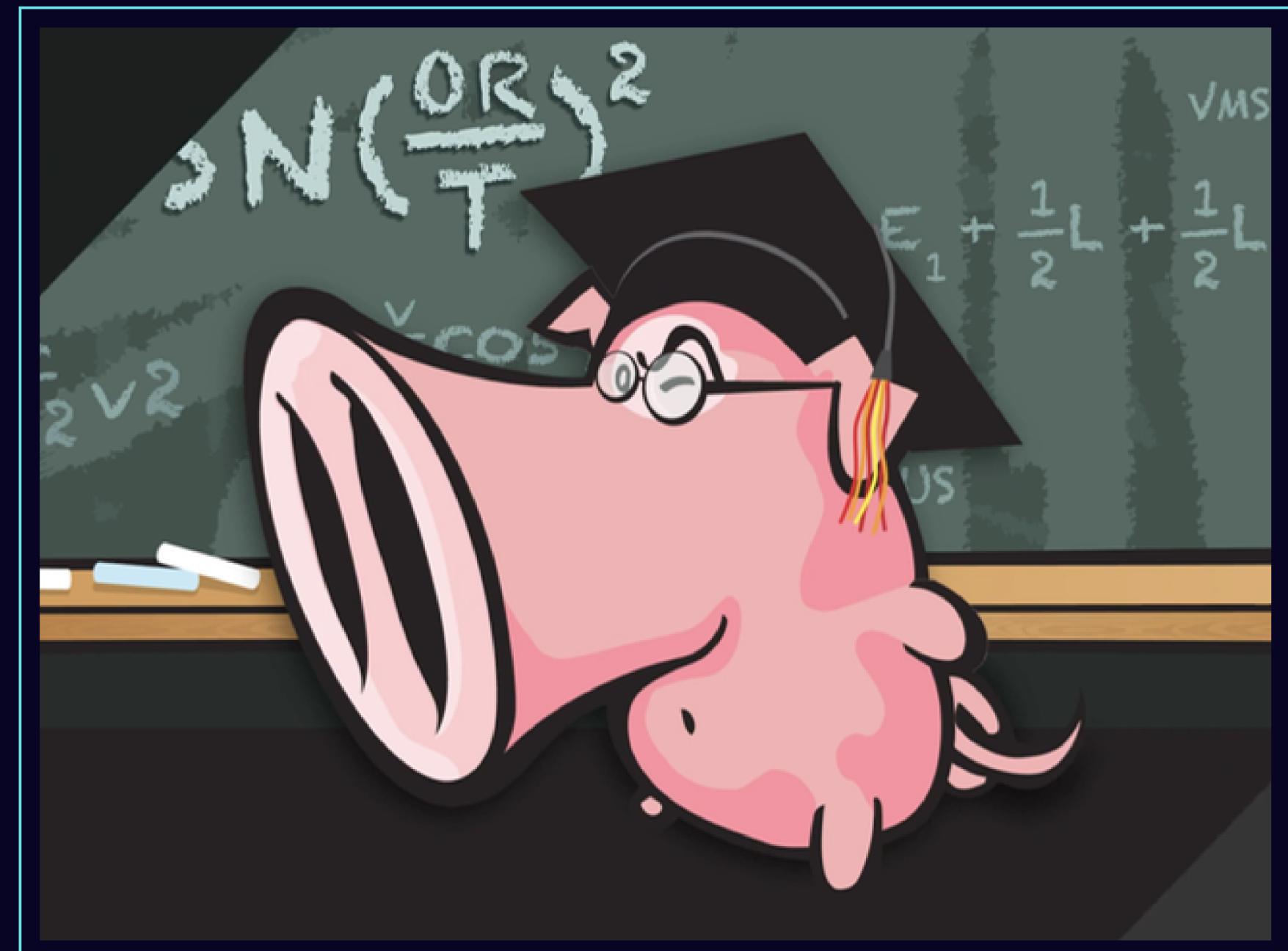
02

Kiến trúc của Snort

Snort có 3 chế độ hoạt động:

- Sniffer mode
- Packet Logger mode
- Network instruction detect system (NIDS)

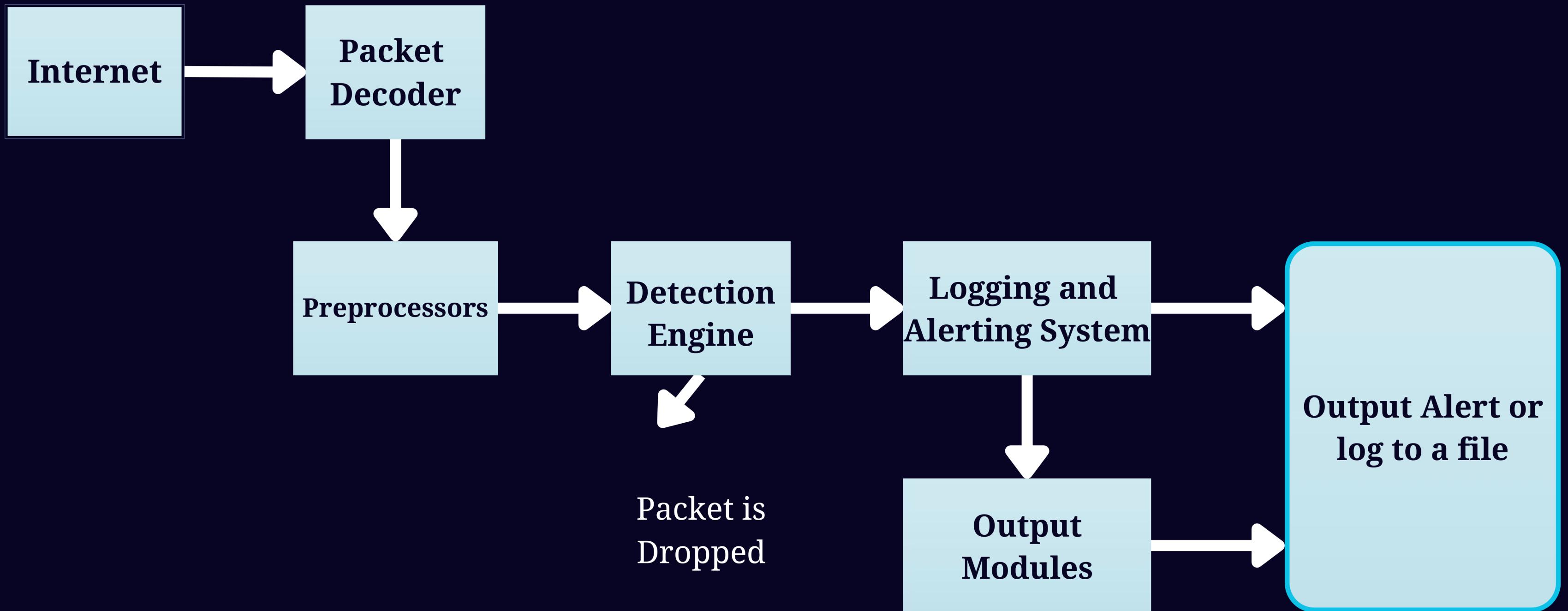
- Bộ giải mã gói tin - Packet Decoder
- Các bộ tiền xử lý - PreProcessers
- Máy phát hiện - Detection Engine
- Hệ thống cảnh báo và ghi dấu - Logging and Alerting System
- Môđun xuất - Output Modules



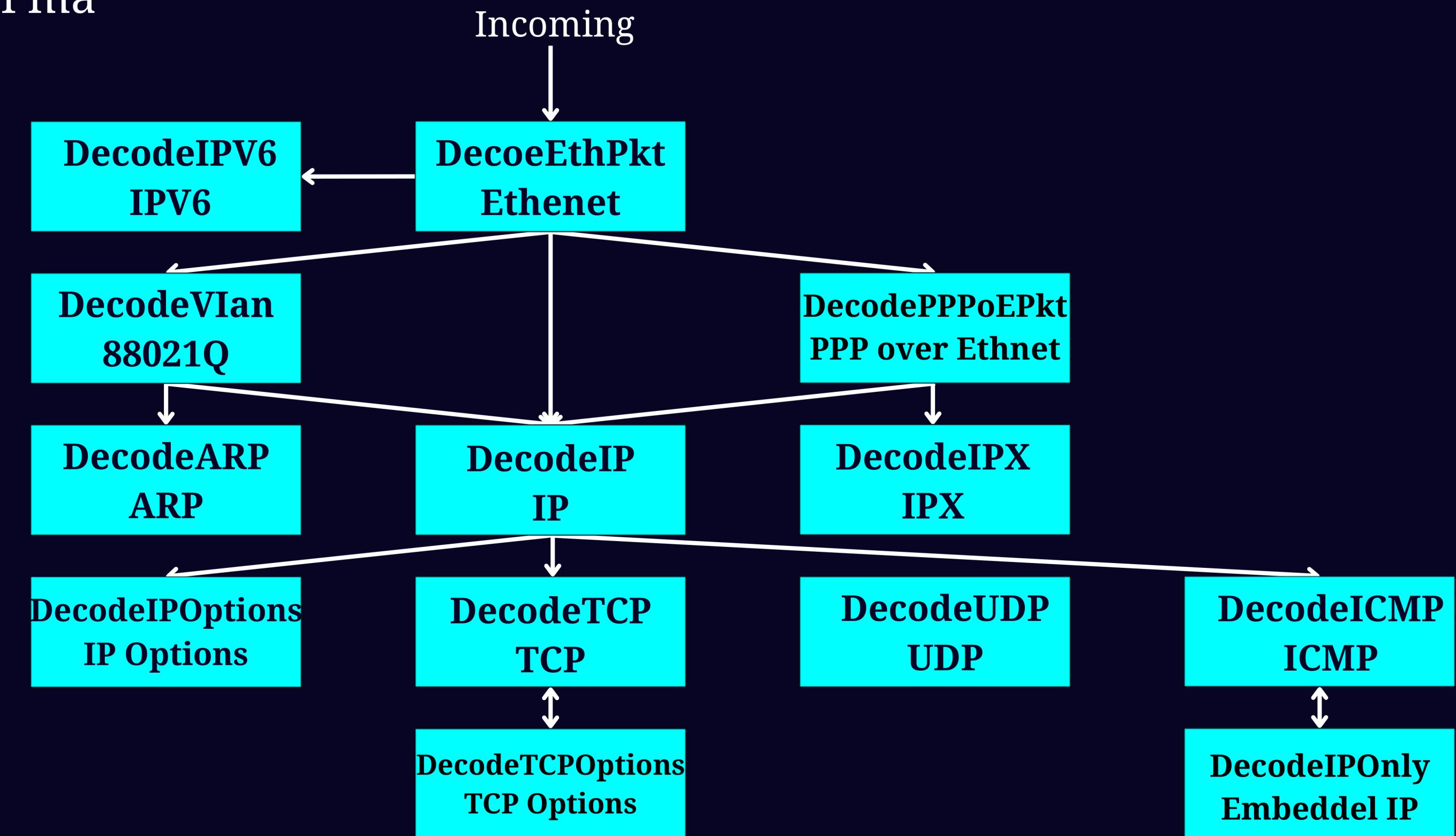
02

Kiến trúc của Snort

Sơ đồ



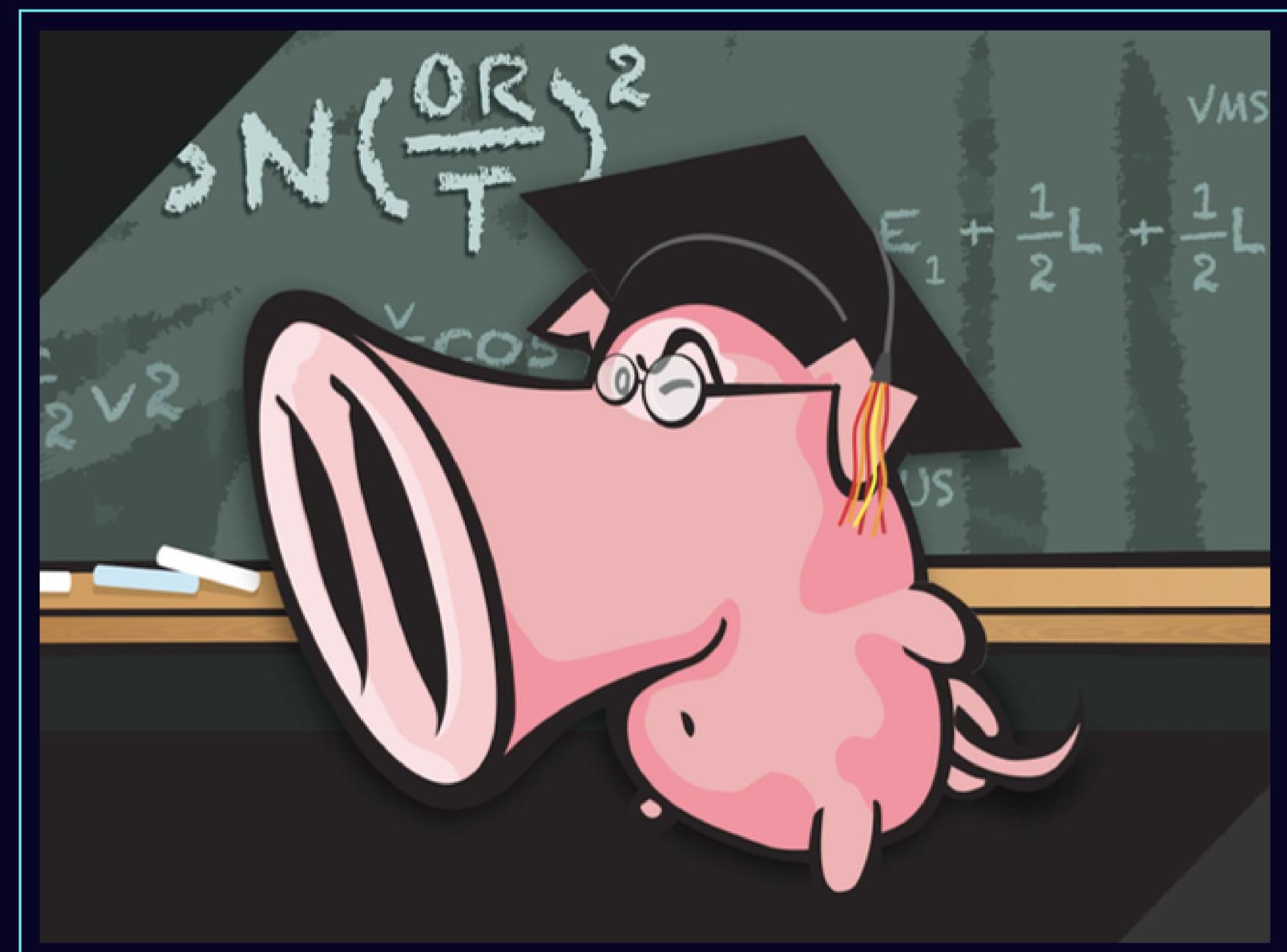
2.1

Module giải mã
gói tin

2.2 Module tiền xử lý

3 nhiệm vụ chính

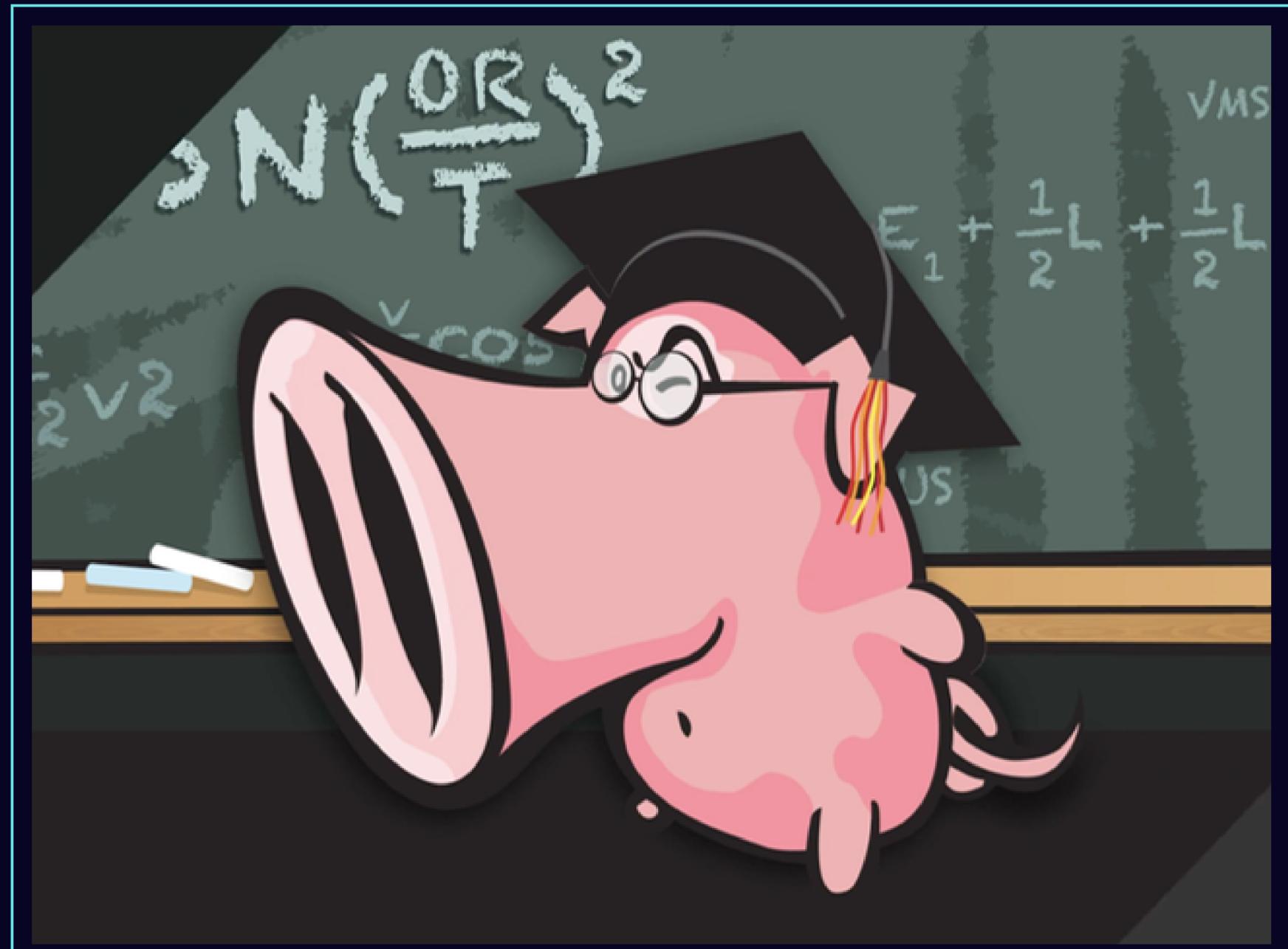
- Kết hợp lại các gói tin
- Giải mã và chuẩn hóa giao thức (decode/normalize)
- Phát hiện các xâm nhập bất thường (nonrule/anormal)



2.3

Module phát hiện

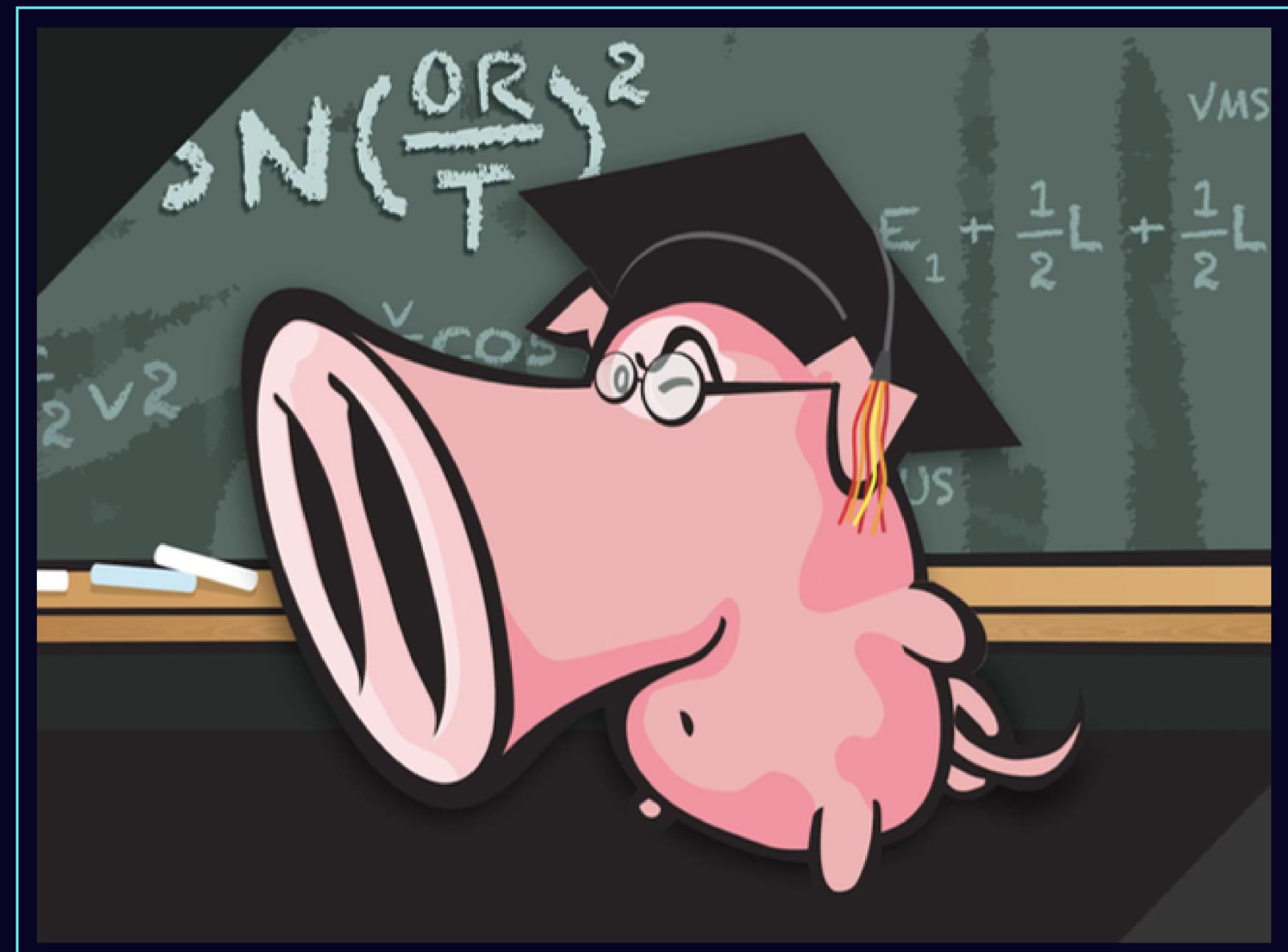
- Module quan trọng nhất
- Chịu trách nhiệm phát hiện xâm nhập
- Khả năng xử lý của module luật phụ thuộc vào:
 - Số lượng luật
 - Tốc độ hệ thống
 - Băng thông mạng
- Một module phát hiện có khả năng tách các phần của gói tin ra và áp dụng luật lên từng phần của gói tin:
 - IP header
 - Header tầng transport : TCP, UDP
 - Header tầng application: DNS, HTTP, FTP
 - Phần tải gói



2.4

Module log và cảnh báo

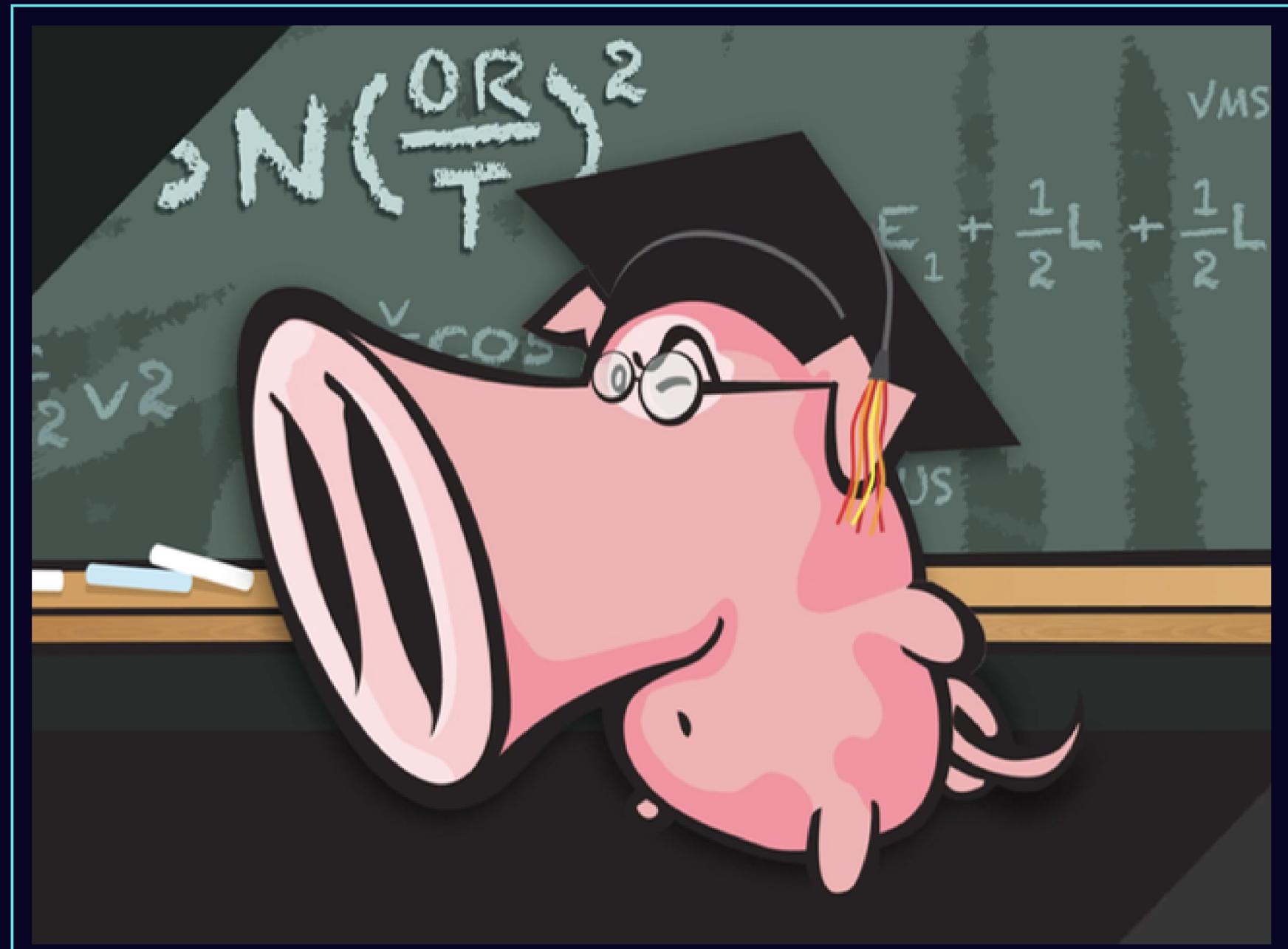
- Tùy thuộc vào việc module phát hiện có nhận dạng được xâm nhập hay không mà gói tin có thể bị ghi log hoặc đưa ra cảnh báo.
- Các file log là các file text dữ liệu



2.5

Module kết xuất thông tin

- Ghi log file
- Ghi syslog
- Ghi cảnh báo vào cơ sở dữ liệu
- Tạo file log dạng xml
- Cấu hình lại Router, firewall
- Gửi các cảnh báo được gói trong gói tin sử dụng giao thức SNMP
- Gửi các thông điệp SMB



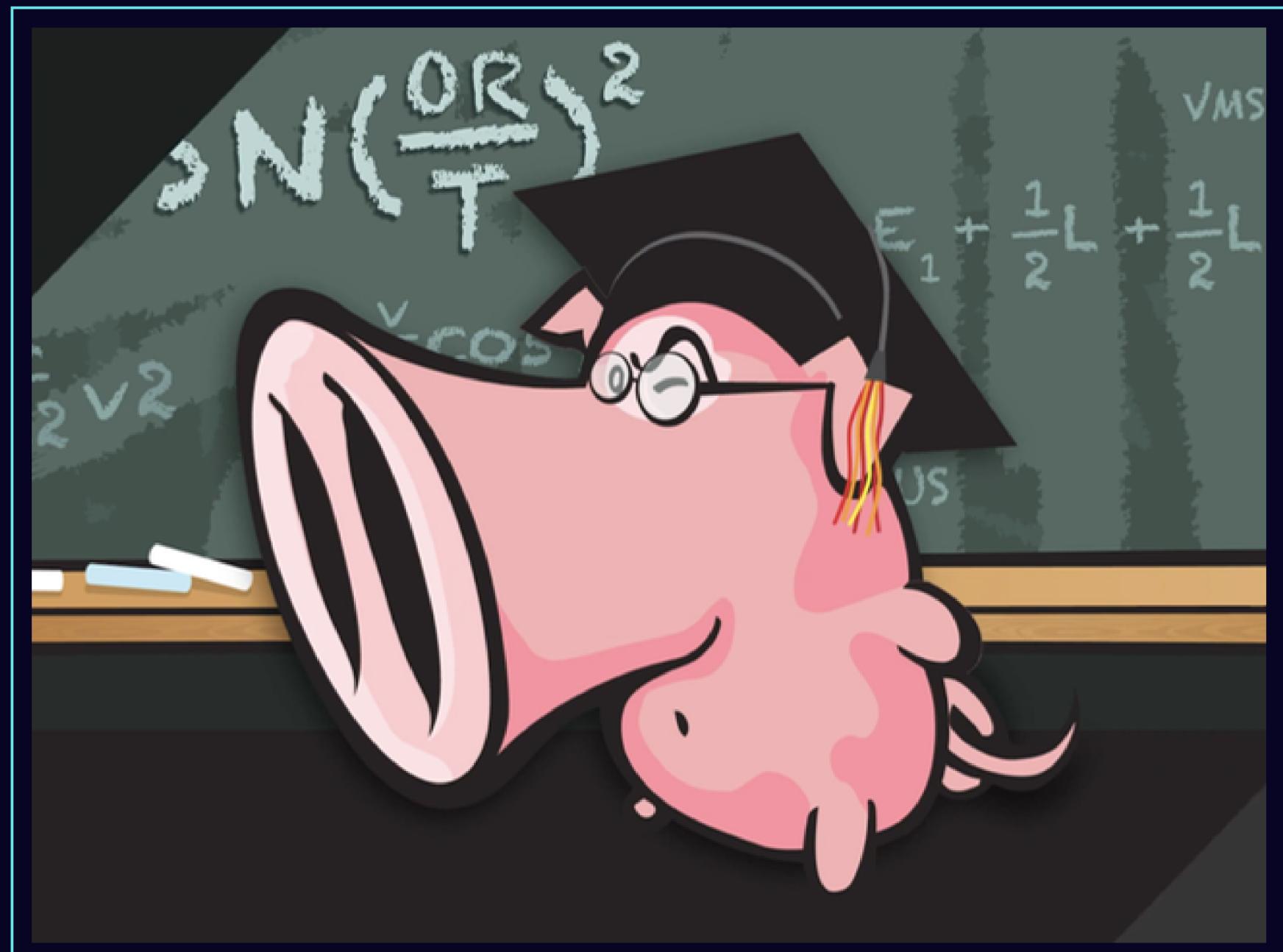
03

Cấu trúc luật của Snort

Dạng cấu trúc

- Phần Header
- Phần Option

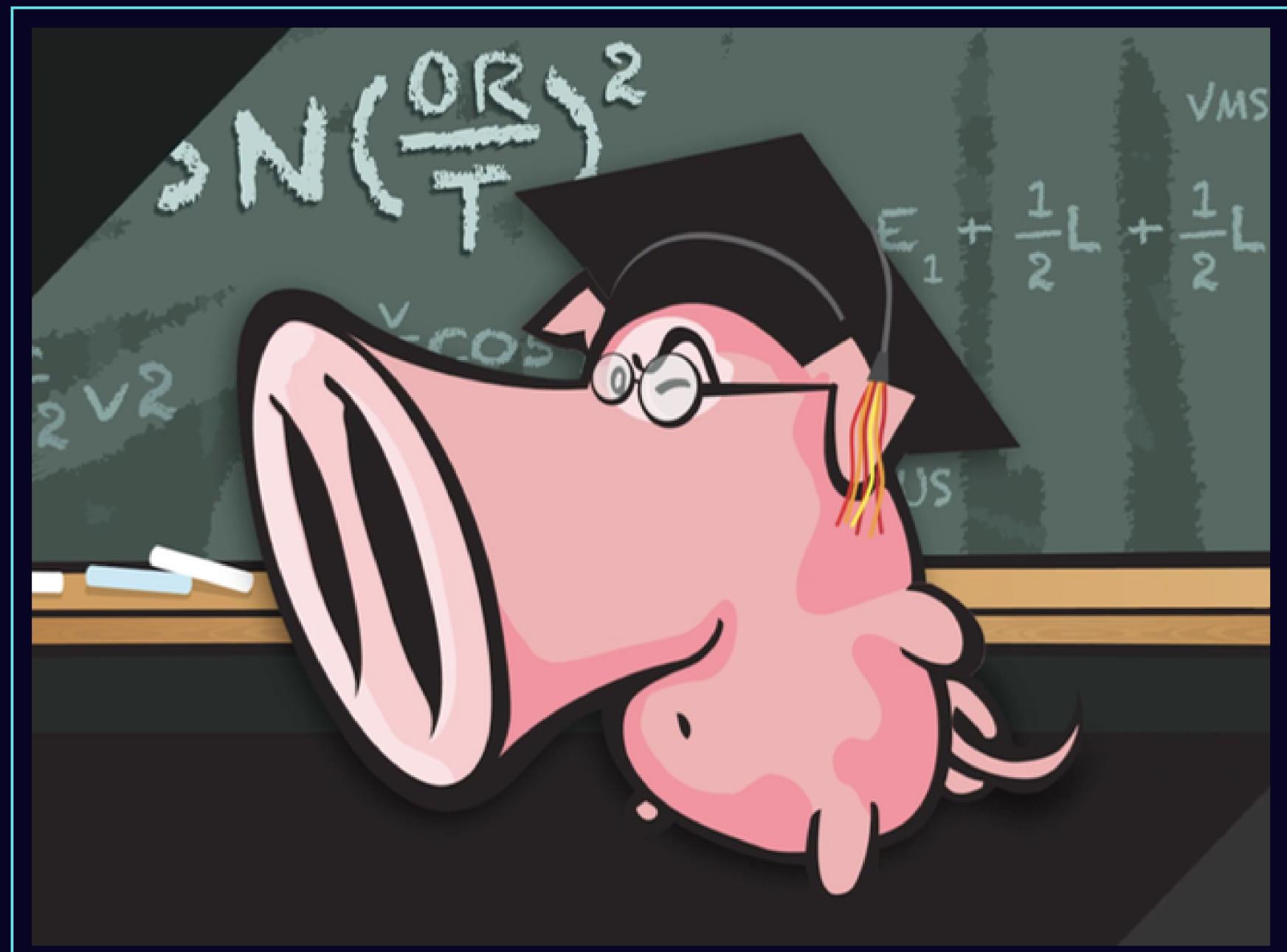
**Alert tcp any any -> 192.168.0.0/22 any (flags:
A; ack: 0; msg: “TCP ping detected”)**



3.1 Phần Header

- Action: là phần quy định loại hành động nào được thực thi
- Protocol: giao thức cụ thể
- Address: địa chỉ nguồn và địa chỉ đích
- Port: xác định các cổng nguồn, cổng đích của một gói tin
- Direction: phần này sẽ chỉ ra địa chỉ nguồn và địa chỉ đích

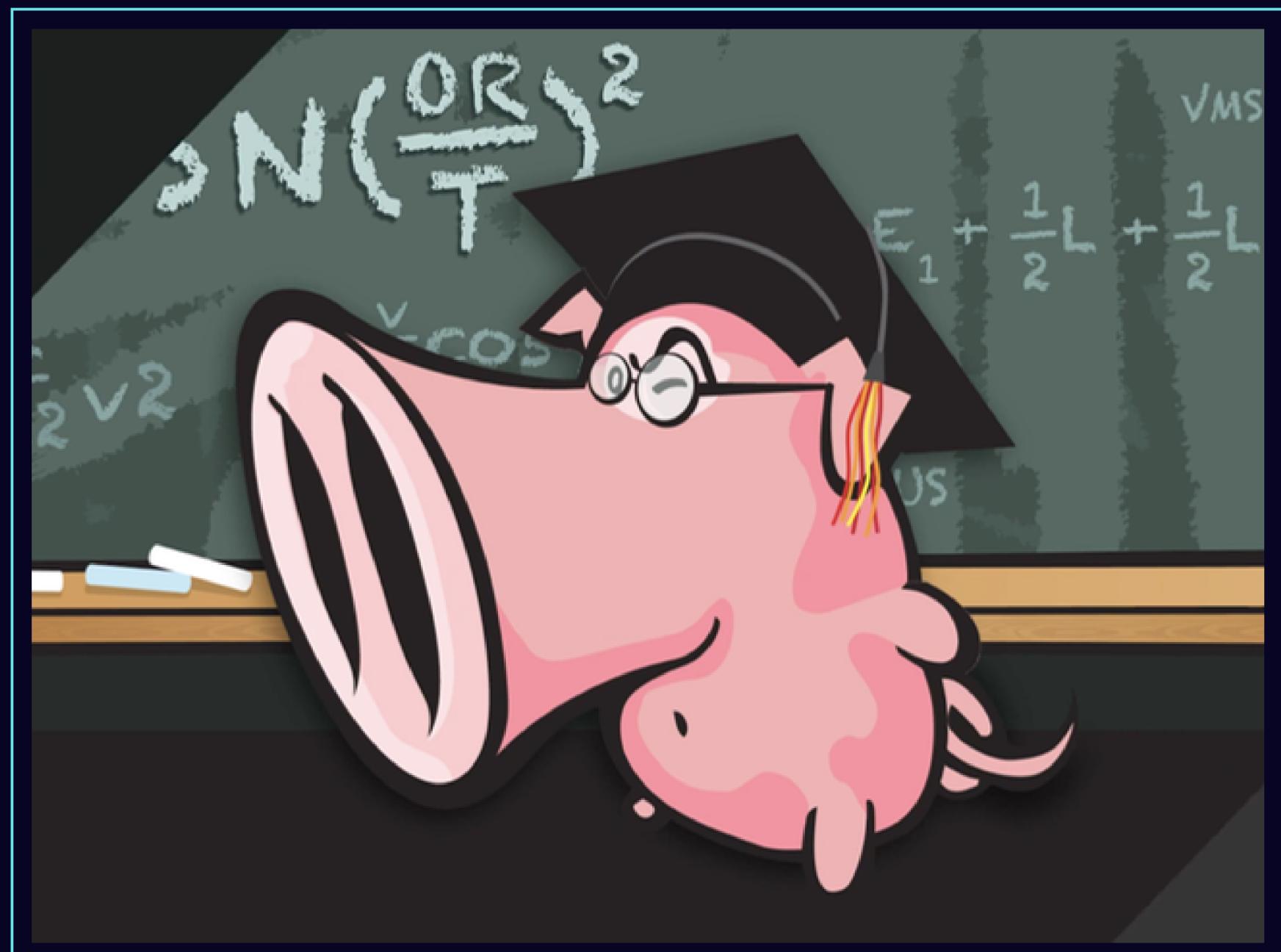
Alert tcp any any -> 192.168.0.0/22 any (flags: A; ack: 0; msg: "TCP ping detected")



3.1.1 Action

Có 4 luật được định nghĩa:

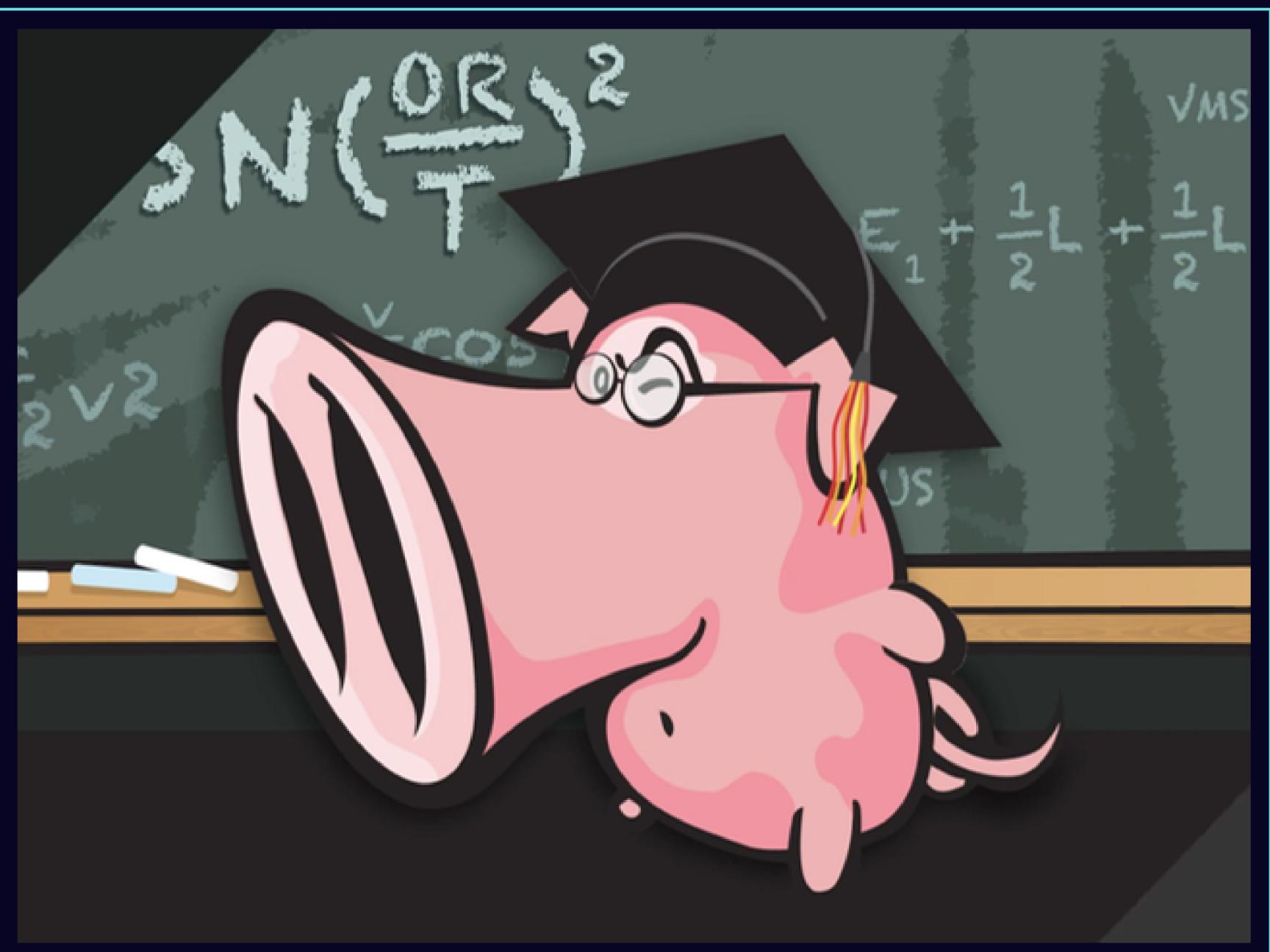
- Pass: Hỗ trợ snort bỏ qua các gói tin không mong muốn
- Log : Dùng để log vào gói tin
- Alert : Gửi thông điệp cảnh báo
- Activate: tạo ra cảnh báo và kích hoạt thêm các luật khác



3.1.2 Giao thức mạng

Các loại giao thức được sử dụng:

- IP : IP định danh và định vị các máy tính và thiết bị mạng.
- ICMP : Giao thức hỗ trợ IP, sử dụng để báo cáo lỗi và trao đổi thông tin điều khiển giữa các thiết bị mạng.
- TCP : Giao thức truyền thông tin tin cậy, thiết lập các kết nối giữa các ứng dụng, đảm bảo dữ liệu được truyền đầy đủ và theo đúng thứ tự.
- UDP : giao thức truyền thông tin không tin cậy, không thiết lập kết nối, phù hợp cho các ứng dụng có yêu cầu thời gian thực như streaming video, VoIP



3.1.3 Địa chỉ và cổng

Địa chỉ gồm 2 loại:

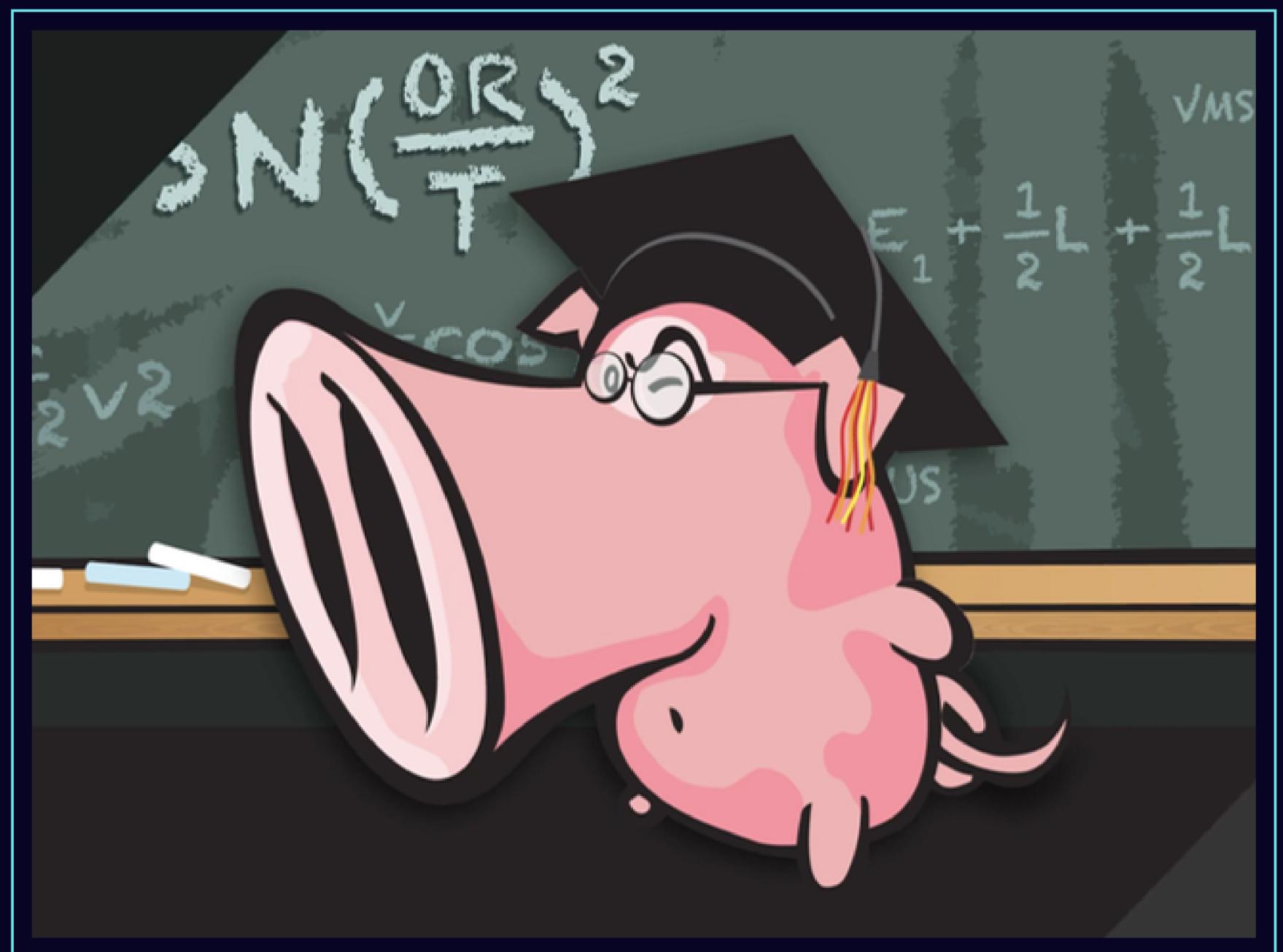
- Địa chỉ nguồn
- địa chỉ đích

Một số cổng điển hình:

- 20 FTP data
- 21 FTP
- 22 SSH
- 23 Telnet
- 24 SMTP

alert icmp !192.168.126.1 24 -> 192.168.1.1 24

(msg: “Ping with TTL=100”; ttl: 100;)

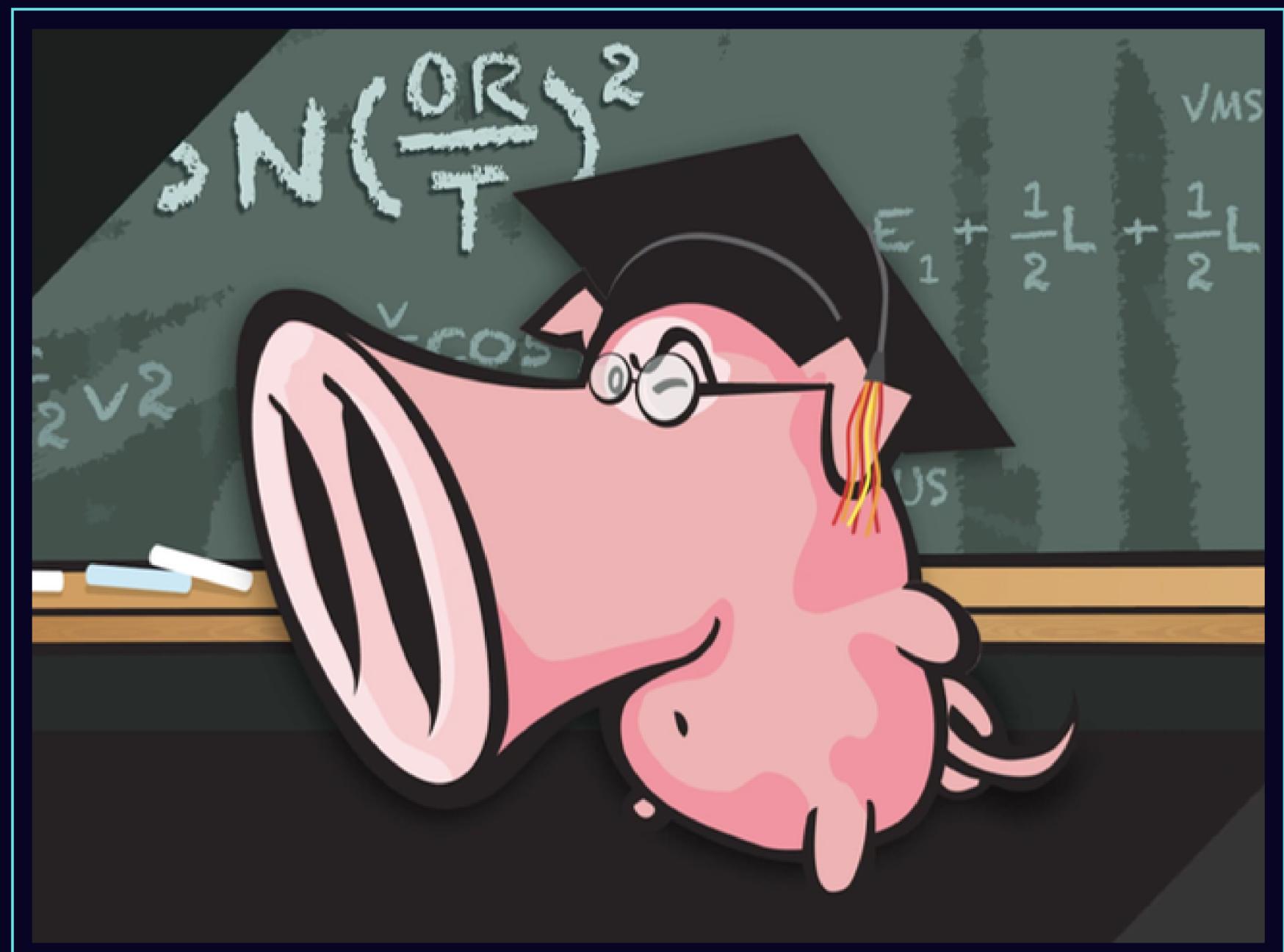


3.1.4 Direction

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

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

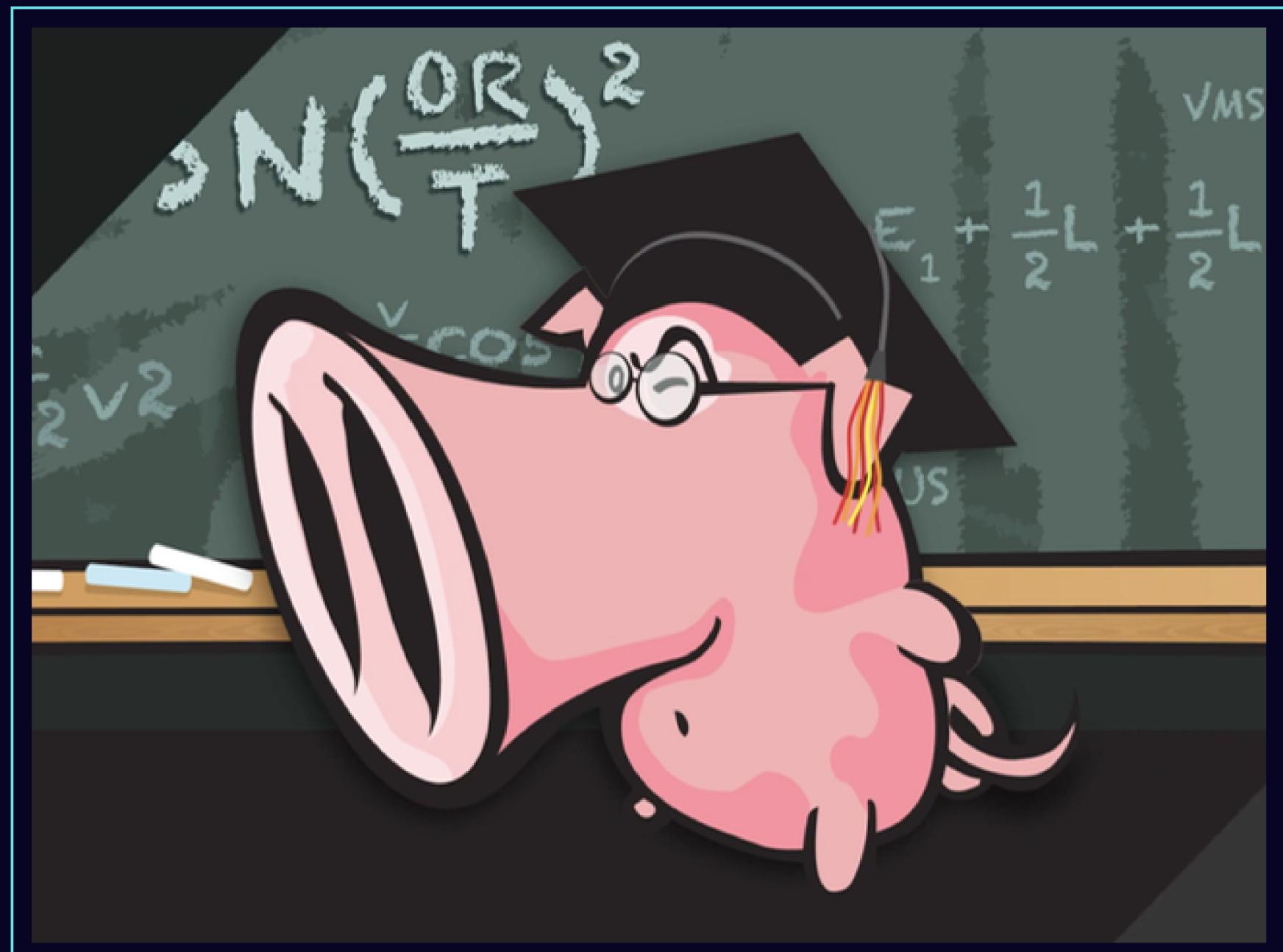
```
alert icmp !192.168.126.1 24 -> 192.168.1.1 24  
(msg: "Ping with TTL=100"; ttl: 100;)
```



3.2 Phần Option

- Bao bọc trong dấu ngoặc đơn
- Các option ngăn cách nhau bởi dấu phẩy
- Được định nghĩa bằng các từ khóa
- Gồm 2 phần:
 - Từ khóa
 - Tham số

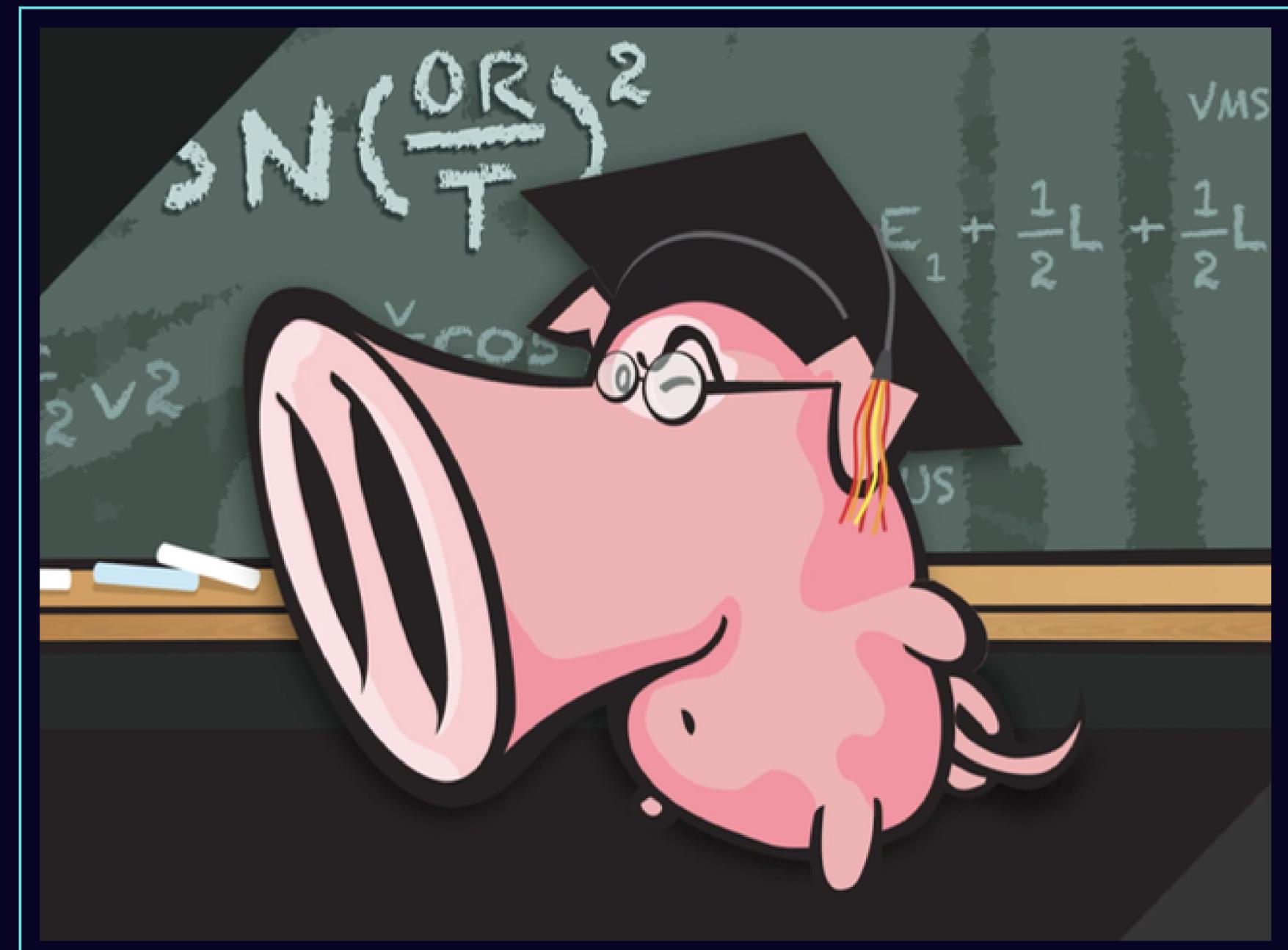
Alert tcp any any -> 192.168.0.0/22 any (flags: A; ack: 0; msg: “TCP ping detected”)



3.2 Phân Option

Một số từ khóa hay gặp:

- Từ khóa ack
- Từ khóa classtype
- Từ khóa content
- Từ khóa dsizе
- Từ khóa Flags
- Từ khóa fragbits



4. KIỂM TRA VÀ BỔ SUNG CÁC QUY TẮC TRONG SNORT

01

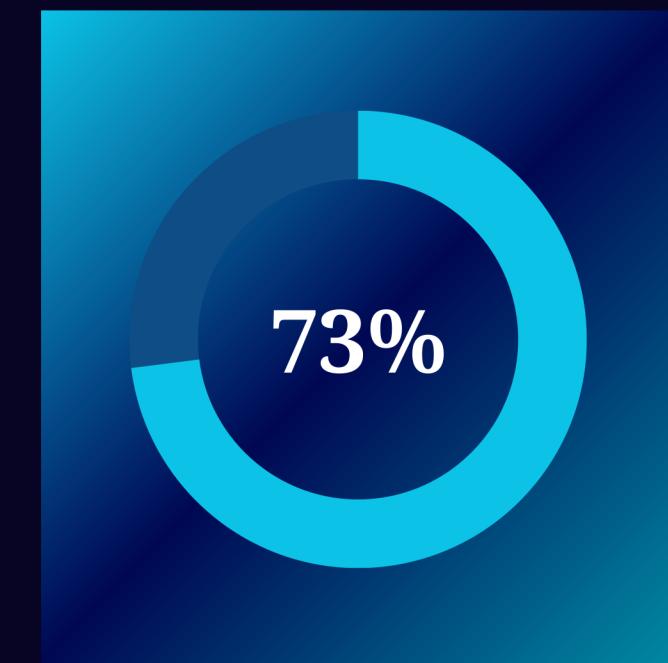
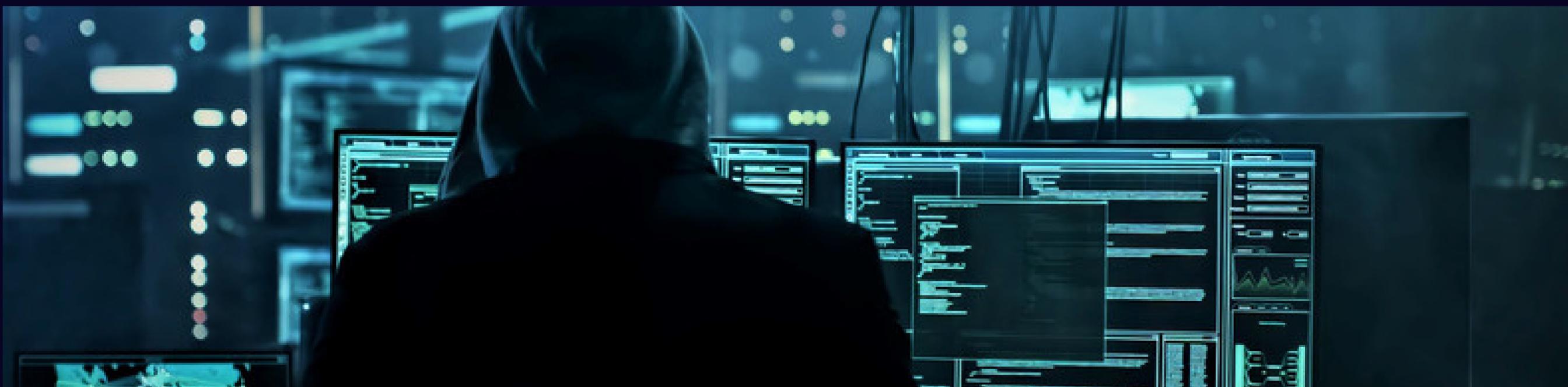
Phát hiện dựa trên quy tắc

02

Viết quy tắc

03

Tùy chọn các quy tắc



Quy tắc trong Snort

Các luật được đặt trong các tệp (.rules)

```
ddos.rules
deleted.rules
dns.rules
dos.rules
experimental.rules
exploit.rules
finger.rules
```

Khai báo trong tệp snort.conf bởi include

```
#include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
#include $RULE_PATH/ddos.rules
#include $RULE_PATH/dns.rules
```





SOICT

KIỂM TRA VÀ BỔ SUNG

Phát hiện dựa trên quy tắc

Hệ thống phát hiện của Snort dựa trên các quy tắc. Các quy tắc này lần lượt dựa trên chữ ký của kẻ xâm nhập. Những chữ ký này có thể có mặt trong các phần tiêu đề của gói hoặc trong tải trọng.

Thứ tự phát hiện các quy tắc dựa trên hành động:

- ***ALERT RULES***
- ***PASS RULES***
- ***LOG RULES***



SOLCT

KIỂM TRA VÀ BỔ SUNG

Phát hiện dựa trên quy tắc

Snort sử dụng một số kỹ thuật và chiến lược để kiểm tra các gói tin mạng nhanh chóng và hiệu quả khi phát hiện tấn công, đặc biệt khi xử lý một số lượng lớn luật (rules).

1. Tối ưu hóa luật (Rule Optimization)
2. Sử dụng cây phân loại (Classification Trees)
3. Để kiểm tra các chuỗi ký tự trong gói tin, Snort sử dụng thuật toán Aho-Corasick, một thuật toán tìm kiếm chuỗi nhanh.
4. Phân mảnh gói tin (Packet Fragmentation)
5. Sử dụng bộ lọc trước (Preprocessors)

Tối ưu hóa luật

- Sắp xếp và nhóm các luật
- Sử dụng các luật tiên quyết (Precedence Rules)
- Áp dụng chiến lược từ chối trước

```
# Cấu hình tối ưu hóa chuỗi tìm kiếm
config detection: search-method ac-bnfa

# Cấu hình luật tiên quyết và nhóm luật
pass ip any any -> any any (msg:"Non-malicious traffic"; sid:1000001;)
alert tcp any any -> any 80 (msg:"HTTP traffic"; content:"GET"; sid:1000002;)

# Sử dụng bộ lọc trước để loại bỏ gói tin không hợp lệ
processor stream5_global: track_tcp yes, track_udp yes
processor stream5_tcp: policy windows, detect_anomalies, require_3whs 180, timeout
```

Sử dụng bộ lọc trước (Preprocessors)

- Chuẩn bị và phân tích giao thức
- Phân mảnh và tái tạo gói tin
- Kiểm tra và lọc lưu lượng

```
# Bộ lọc trước cho phân mảnh gói tin IP
preprocessor frag3_engine: policy windows detect_anomalies

# Bộ lọc trước cho kiểm tra các bất thường của TCP
preprocessor stream5_global: track_tcp yes, track_udp yes
preprocessor stream5_tcp: policy windows, detect_anomalies, require_3whs 180, timeout 180

# Bộ lọc trước cho HTTP
preprocessor http_inspect: global iis_unicode_map unicode.map 1252
preprocessor http_inspect_server: server default profile all ports { 80 8080 8180 }
```



SOLCT

KIỂM TRA VÀ BỔ SUNG

Phát hiện dựa trên quy tắc

Hệ thống phát hiện của Snort dựa trên các quy tắc. Các quy tắc này lần lượt dựa trên chữ ký của kẻ xâm nhập. Những chữ ký này có thể có mặt trong các phần tiêu đề của gói hoặc trong tải trọng.

Thứ tự phát hiện các quy tắc dựa trên hành động:

- ***ALERT RULES***
- ***PASS RULES***
- ***LOG RULES***

Các quy tắc mới trong SNORT 3

- 01 Quy tắc dịch vụ
- 02 Quy tắc tệp
- 03 Quy tắc nhận dạng tệp

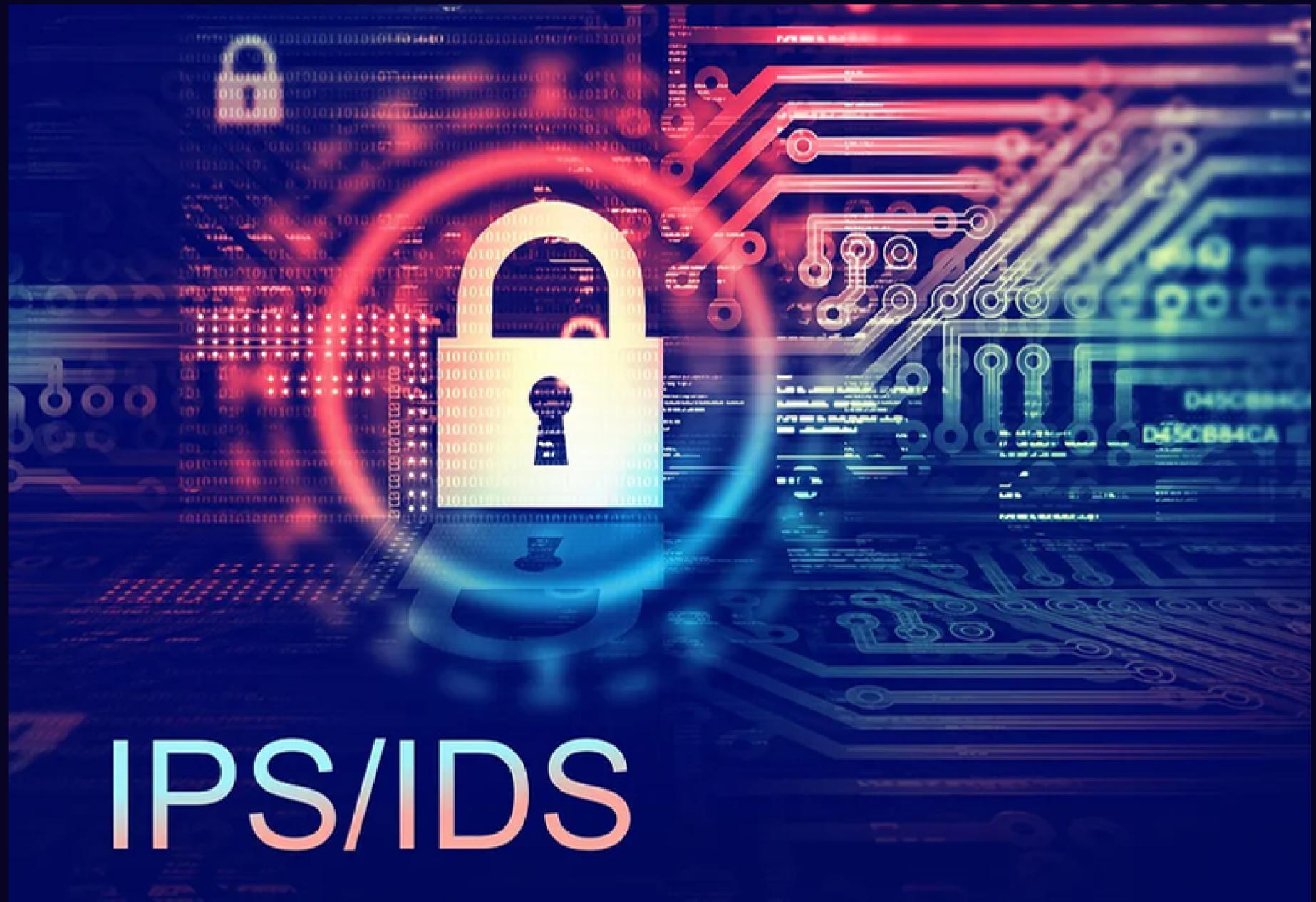


Quy tắc dịch vụ

Cho phép người viết quy tắc đõi sánh lưu lượng truy cập của một dịch vụ cụ thể

Loại quy tắc này đặc biệt hữu ích cho các dịch vụ như HTTP, nơi thường thấy các máy chủ web chạy trên các cổng TCP khác 80.

KIỂM TRA VÀ BỔ SUNG





SOICT

KIỂM TRA VÀ BỔ SUNG

Quy tắc sau sẽ cảnh báo về việc khớp lưu lượng HTTP bất kể cổng hoặc địa chỉ IP được sử dụng trong kết nối:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
```

```
(
```

```
msg:"WEB-ATTACKS /usr/bin/perl execution attempt";
```

```
flow:to_server,established;
```

```
content:"/usr/bin/perl"; nocase;
```

```
classtype: web-application-attack;
```

```
sid:1355;
```

```
rev:5;
```

```
)
```

```
06/06-00:38:54.894389 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {ICMP} 192.168.126.254 -> 192.168.126.130
06/06-00:38:55.858471 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {IPV6-ICMP} fe80::de57:5c11:fbac:95e6 -> ff02::16
06/06-00:38:55.874413 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {IPV6-ICMP} fe80::de57:5c11:fbac:95e6 -> ff02::16
06/06-00:38:55.930246 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {IPV6-ICMP} fe80::de57:5c11:fbac:95e6 -> ff02::16
06/06-00:38:55.945971 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {IPV6-ICMP} fe80::de57:5c11:fbac:95e6 -> ff02::16
06/06-00:38:56.074136 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {IPV6-ICMP} fe80::de57:5c11:fbac:95e6 -> ff02::12
06/06-00:38:56.226515 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {ICMP} 192.168.126.254 -> 192.168.126.130
06/06-00:38:56.249787 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {ICMP} 192.168.126.130 -> 192.168.126.254
06/06-00:39:00.033701 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {IPV6-ICMP} fe80::de57:5c11:fbac:95e6 -> ff02::2
06/06-00:39:04.035754 [**] [1:10000001:1] WEB-ATTACKS /usr/bin/perl execution attempt [**] [Priority: 0] {IPV6-ICMP} fe80::de57:5c11:fbac:95e6 -> ff02::2
```



Quy tắc tệp

"Quy tắc tệp" mới của Snort cho phép người viết quy tắc tạo quy tắc để khớp với một tệp cụ thể bất kể giao thức, IP nguồn, IP đích, cổng và dịch vụ. Snort có thể xử lý các tệp được gửi bằng bất kỳ giao thức lớp ứng dụng nào sau đây:

- HTTP
- SMTP
- POP3
- IMAP
- SMB
- FTP

Các quy tắc này được tạo bằng tiêu đề quy tắc chỉ chứa một hành động sau là tệp từ khóa: *action file*

Quy tắc nhận dạng tệp

Các quy tắc này là các quy tắc cơ bản của Snort 3, nhưng thay vì cảnh báo và/hoặc chặn lưu lượng truy cập, chúng xác định các tệp dựa trên nội dung của tệp đó và sau đó xác định loại tệp có thể được sử dụng trong các quy tắc tiếp theo với các tùy chọn file_type .





SOICT

KIỂM TRA VÀ BỔ SUNG

Quy tắc nhận dạng tệp có hai thành phần chính:

- Một tiêu đề quy tắc chỉ bao gồm file_id, cho Snort biết rằng quy tắc sau là định nghĩa loại tệp
- Tùy file_meta chọn quy tắc đặt siêu dữ liệu tệp cho quy tắc nhận dạng tệp nhất định

```
file_id (
    msg:"Windows/DOS executable file";
    file_meta:type MSEXE, id 21, category "Executables,Dynamic Analysis Capable,Local Malware
Analysis Capable";
    file_data;
    content:"| 4D 5A |", depth 2, offset 0;
    gid:4;
    sid:16;
    rev:1;
)
```

ỨNG DỤNG



Đối với doanh nghiệp và tổ chức

Đóng vai trò quan trọng trong việc giám sát lưu lượng mạng, cung cấp khả năng hiển thị thời gian thực về các mối đe dọa tiềm ẩn. Bằng cách liên tục phân tích các gói, nó có thể phát hiện và phản hồi các hoạt động đáng ngờ.

Thiết kế mạng an toàn

Tích hợp Snort vào kiến trúc mạng, các tổ chức có thể củng cố vị thế bảo mật tổng thể của họ. Snort hoạt động như một lớp phòng thủ bổ sung, bổ sung cho Tường lửa, phần mềm chống vi-rút.

Ứng dụng trong cuộc sống hằng ngày

- Bảo vệ mạng gia đình
- Bảo vệ tài khoản trực tuyến



SOLCT

CẢM ƠN
THẦY VÀ
CÁC BẠN ĐÃ
LẮNG NGHE

