

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA AN TOÀN THÔNG TIN**



**Môn: Thực tập cơ sở**

**BÀI BÁO THỰC TẬP CƠ SỞ  
Bài 6: Cài đặt cấu hình HIDS/NIDS**

<b>Họ và tên giảng viên:</b>	TS. Đinh Trường Duy
<b>Họ và tên:</b>	Nguyễn Quốc Khánh
<b>Mã sinh viên:</b>	B20DCAT103
<b>Lớp:</b>	D20CQAT03-B
<b>Số điện thoại:</b>	0964137761

*Hà Nội 2023*

# 1. Tìm hiểu lý thuyết

## a. Khái quát về các hệ thống phát hiện tấn công, xâm nhập

Hệ thống phát hiện, ngăn chặn tấn công, xâm nhập IDS (Intrusion Detection System) là một lớp phòng vệ quan trọng trong các lớp giải pháp đảm bảo an toàn cho hệ thống thông tin và mạng theo mô hình phòng thủ có chiều sâu. IDS thường được kết nối vào các bộ định tuyến, switch, card mạng và chủ yếu làm nhiệm vụ giám sát và cảnh báo, không có khả năng chủ động ngăn chặn tấn công, xâm nhập.

Nhiệm vụ chính của hệ thống IDS bao gồm:

- Giám sát lưu lượng mạng hoặc các hành vi trên một hệ thống để nhận dạng các dấu hiệu của tấn công, xâm nhập.
- Khi phát hiện các hành vi tấn công, xâm nhập, thì ghi logs các hành vi này cho phân tích bổ sung sau này.
- Gửi thông báo cho người quản trị về các hành vi tấn công, xâm nhập đã phát hiện. Có 2 phương pháp phân loại chính các hệ thống IDS, gồm phân loại theo nguồn dữ liệu và phân loại theo kỹ thuật phân tích dữ liệu. Theo nguồn dữ liệu, có 2 loại hệ thống phát hiện xâm nhập:
  - Hệ thống phát hiện xâm nhập mạng (NIDS – Network-based IDS): NIDS phân tích lưu lượng mạng để phát hiện tấn công, xâm nhập cho cả mạng hoặc một phần mạng.
  - Hệ thống phát hiện xâm nhập cho host (HIDS – Host-based IDS): HIDS phân tích các sự kiện xảy ra trong hệ thống/dịch vụ để phát hiện tấn công, xâm nhập cho hệ thống đó.

Theo kỹ thuật phân tích dữ liệu, có 2 kỹ thuật phân tích chính:

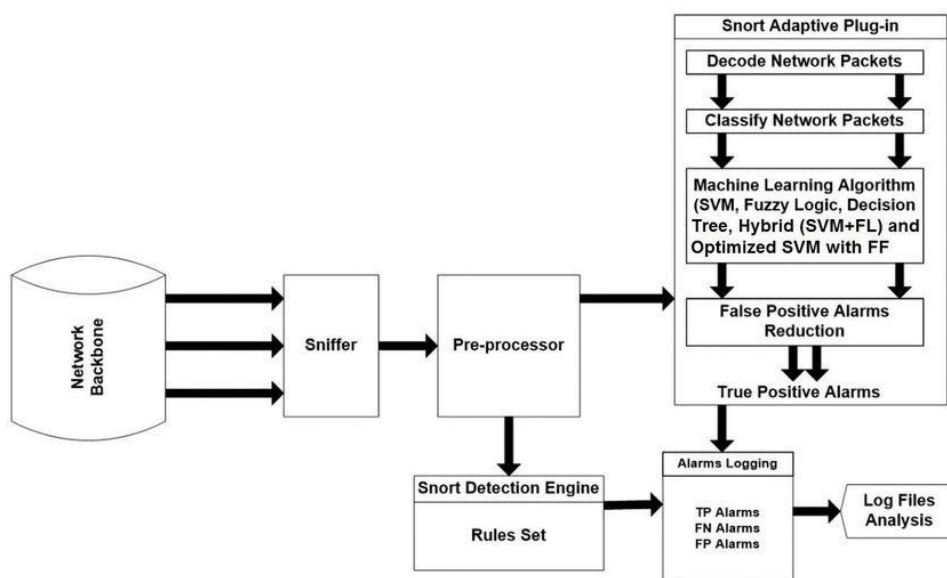
- Phát hiện xâm nhập dựa trên chữ ký: Phát hiện xâm nhập dựa trên chữ ký trước hết cần xây dựng cơ sở dữ liệu các chữ ký, hoặc các dấu hiệu của các loại tấn công, xâm nhập đã biết. Bước tiếp theo là sử dụng cơ sở dữ liệu

các chữ ký để giám sát các hành vi của hệ thống, hoặc mạng, và cảnh báo nếu phát hiện chữ ký của tấn công, xâm nhập.

- Phát hiện xâm nhập dựa trên các bất thường: Dựa trên giả thiết các hành vi tấn công, xâm nhập thường có quan hệ chặt chẽ với các hành vi bất thường. Quá trình xây dựng và triển khai gồm 2 giai đoạn huấn luyện và phát hiện.

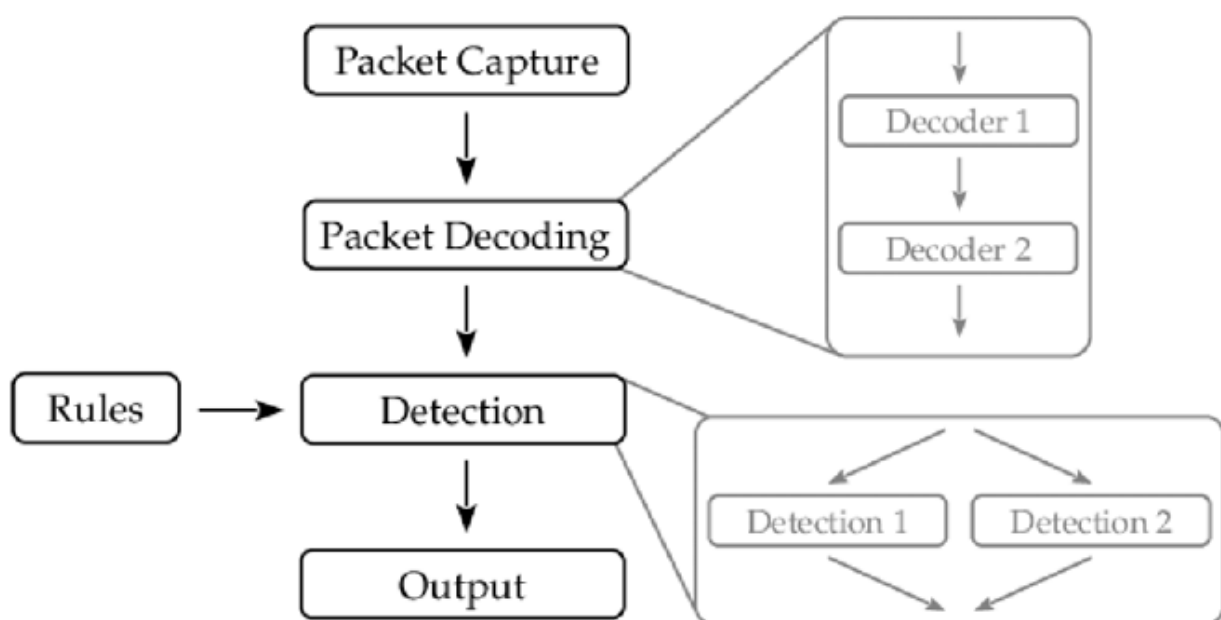
## b. Kiến trúc và tính năng của một số hệ thống phát hiện tấn công, xâm nhập

Snort là một hệ thống phòng chống xâm nhập mạng nguồn mở, có khả năng thực hiện phân tích lưu lượng truy cập và đăng nhập gói trên mạng IP. Nó có thể thực hiện phân tích giao thức, tìm kiếm/so sánh nội dung, và có thể được sử dụng để phát hiện ra các loại tấn công và thăm dò, chẳng hạn như tràn bộ đệm, quét cổng tàng hình, tấn công CGI, thăm dò SMB, các nỗ lực nhận dạng đặc điểm hệ điều hành, và nhiều hơn nữa. Snort dựa trên thư viện bắt gói (libpcap). libpcap là một công cụ được sử dụng rộng rãi trong giao thức điều khiển truyền dẫn /giao thức internet, tìm kiếm và phân tích nội dung để ghi nhật ký, phân tích lưu lượng truy cập, phân tích giao thức và kết hợp nội dung.



**Hình 1: Sơ đồ khối các thành phần của Snort**

Suricata là IDS, IPS mạng có hiệu suất cao và là công cụ giám sát bảo mật mạng. Nó có mã nguồn mở và thuộc sở hữu của một nền tảng phi lợi nhuận chạy cộng đồng, nền tảng bảo mật thông tin mở (OISF). Suricata có thể ghi nhật ký các yêu cầu HTTP, đăng nhập và lưu trữ chứng chỉ TLS, trích xuất các tệp từ lưu lượng mạng và lưu trữ chúng vào đĩa, hỗ trợ bắt pcap đầy đủ cho phép dễ dàng phân tích. Ghi nhật ký và phân tích TLS / SSL, HTTP, DNS. Phân tích chức năng nâng cao với lập trình Lua để phát hiện những thứ không thể trong quy tắc cú pháp.



**Hình 2: Kiến trúc hoạt động của Suricata**

OSSEC là một nền tảng để theo dõi và kiểm soát hệ thống của bạn. Nó kết hợp tất cả các khía cạnh của HIDS (Phát hiện xâm nhập dựa trên máy chủ), giám sát nhật ký và quản lý sự cố bảo mật (SIM) / Thông tin bảo mật và quản lý sự kiện (SIEM) với nhau trong một giải pháp đơn giản, mạnh mẽ, mã nguồn mở.



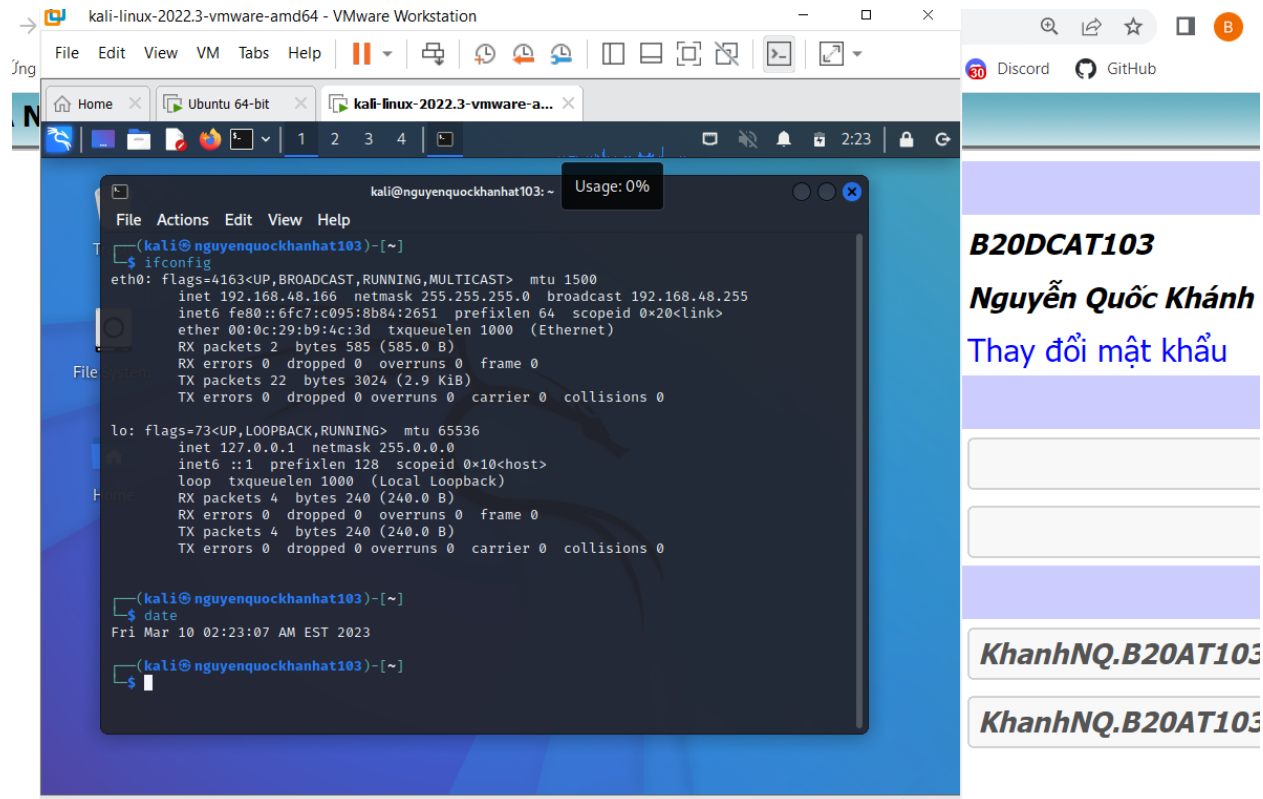
**Hình 3: Kiến trúc hoạt động của OSSEC**

Wazuh là một nền tảng miễn phí và nguồn mở để phát hiện mối đe dọa, giám sát an ninh, đáp ứng sự cố và tuân thủ quy định. Nó có thể được sử dụng để giám sát các điểm cuối, dịch vụ đám mây và các container, tổng hợp và phân tích dữ liệu từ các nguồn bên ngoài. Với khả năng phân tích bảo mật, phát hiện tấn công, thăm dò, phân tích nhật ký, theo dõi tính toàn vẹn tập tin, phát hiện lỗi hỏng, báo cáo sự việc, đáp ứng các quy định,... Kiến trúc Wazuh dựa trên các đại lý(Agent), chạy trên các điểm cuối được giám sát, chuyển tiếp dữ liệu bảo mật đến một máy chủ trung tâm. Hơn nữa, các thiết bị không có đại lý(như tường lửa, bộ chuyển đổi, bộ định tuyến, điểm truy cập, v.v.) được hỗ trợ và có thể chủ động gửi dữ liệu nhật ký qua SYSLOG, SSH hoặc sử dụng API riêng. Máy chủ trung tâm giải mã và phân tích thông tin đến và chuyển kết quả theo cụm ElasticSearch để lập chỉ mục và lưu trữ.

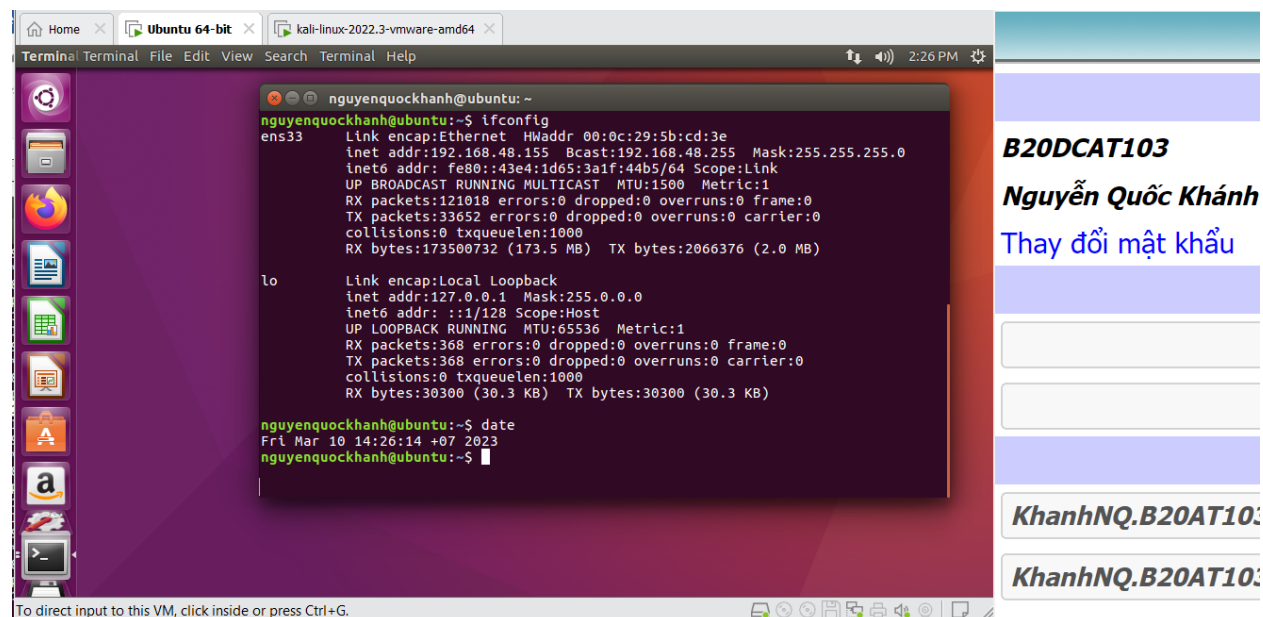
## 2. Các bước thực hiện

### Bước 1: Chuẩn bị các máy tính

**Chuẩn bị máy Kali với địa chỉ IP và kết nối được với máy Snort:**

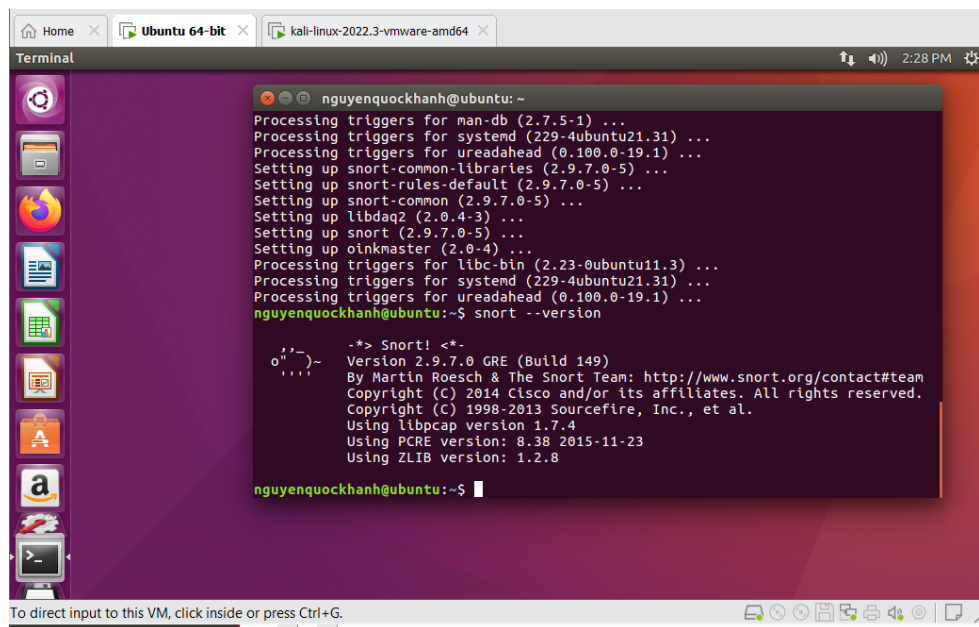


**Chuẩn bị máy Snort với địa chỉ IP và kết nối được với máy Kali:**



## Bước 2: Tải, cài đặt Snort và chạy thử Snort

### Tải và cài đặt snort



```
nguyenquockhanh@ubuntu: ~  
Processing triggers for man-db (2.7.5-1) ...  
Processing triggers for systemd (229-4ubuntu21.31) ...  
Processing triggers for ureadahead (0.100.0-19.1) ...  
Setting up snort-common-libraries (2.9.7.0-5) ...  
Setting up snort-rules-default (2.9.7.0-5) ...  
Setting up snort-common (2.9.7.0-5) ...  
Setting up libdaq2 (2.0.4-3) ...  
Setting up snort (2.9.7.0-5) ...  
Setting up oinkmaster (2.0-4) ...  
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...  
Processing triggers for systemd (229-4ubuntu21.31) ...  
Processing triggers for ureadahead (0.100.0-19.1) ...  
nguyenquockhanh@ubuntu:~$ snort --version  
  
-*> Snort! <-*  
o"/~  
....~  
Version 2.9.7.0 GRE (Build 149)  
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.7.4  
Using PCRE version: 8.38 2015-11-23  
Using ZLIB version: 1.2.8  
  
nguyenquockhanh@ubuntu:~$
```

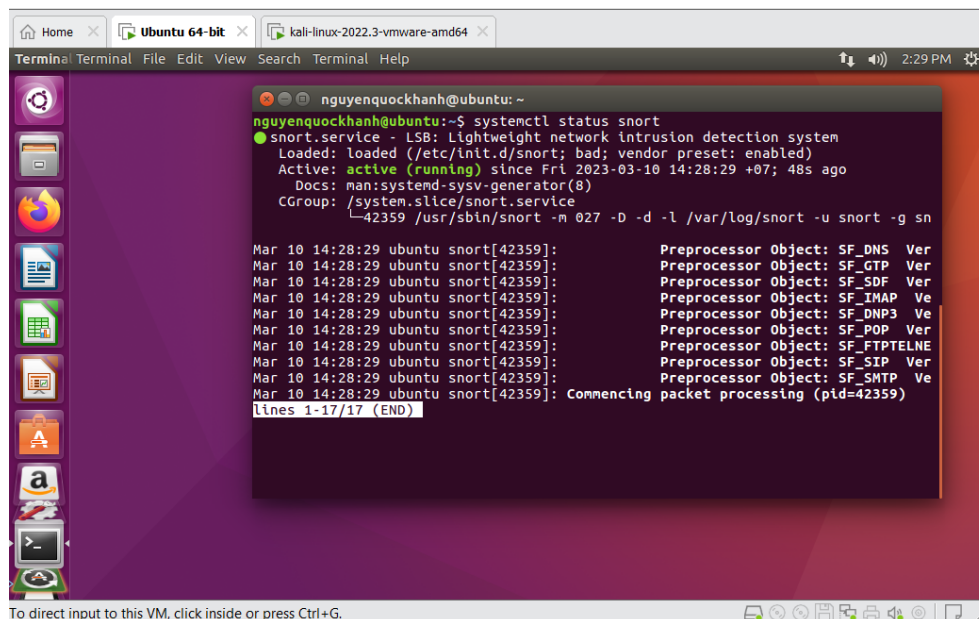
B20DCAT103

Nguyễn Quốc Khánh

Thay đổi mật khẩu

KhanhNQ.B20AT103

KhanhNQ.B20AT103



```
nguyenquockhanh@ubuntu: ~  
nguyenquockhanh@ubuntu:~$ systemctl status snort  
● snort.service - LSB: Lightweight network intrusion detection system  
   Loaded: loaded (/etc/init.d/snort; bad; vendor preset: enabled)  
   Active: active (running) since Fri 2023-03-10 14:28:29 +07; 48s ago  
     Docs: man:systemd-sysv-generator(8)  
    CGroup: /system.slice/snort.service  
            └─42359 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -g sn  
  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_DNS Ver  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_GTP Ver  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_SDF Ver  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_IMAP Ve  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_DNP3 Ve  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_POP Ver  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_FTPTELNE  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_SIP Ver  
Mar 10 14:28:29 ubuntu snort[42359]: Preprocessor Object: SF_SMTP Ve  
Mar 10 14:28:29 ubuntu snort[42359]: Commencing packet processing (pid=42359)  
lines 1-17/17 (END)
```

B20DCAT103

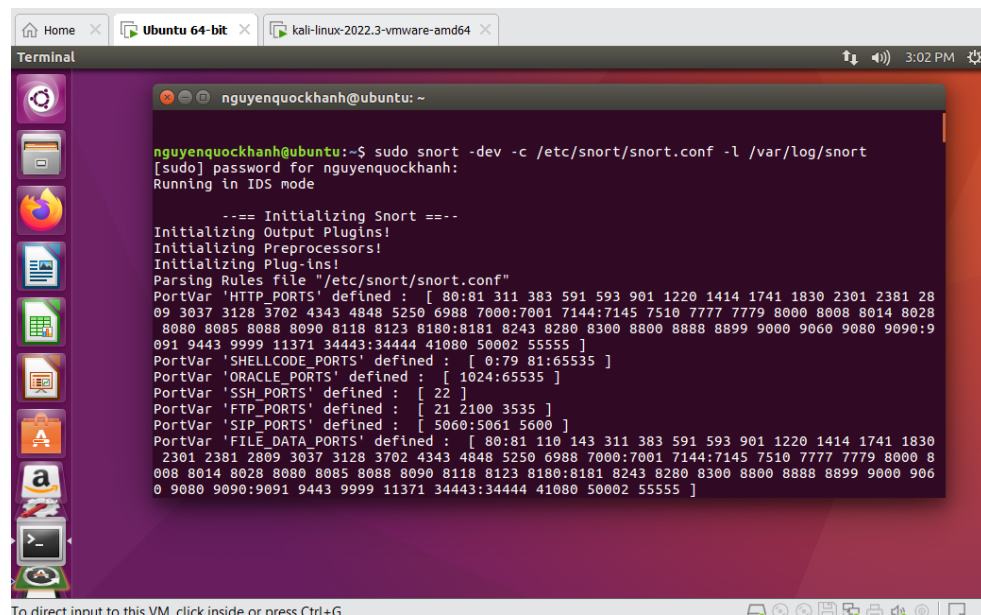
Nguyễn Quốc Khánh

Thay đổi mật khẩu

KhanhNQ.B20AT103

KhanhNQ.B20AT103

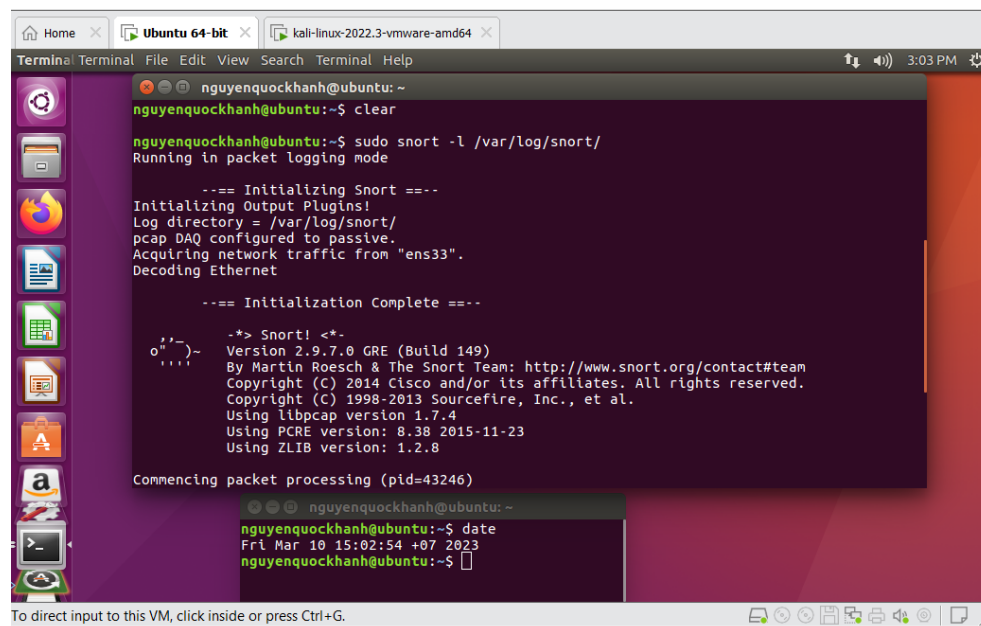
## Chạy thử Snort



The screenshot shows a terminal window with the following output:

```
nguyenquockhanh@ubuntu: ~  
nguyenquockhanh@ubuntu:~$ sudo snort -dev -c /etc/snort/snort.conf -l /var/log/snort  
[sudo] password for nguyenquockhanh:  
Running in IDS mode  
  
==== Initializing Snort ====  
Initializing Output Plugins!  
Initializing Preprocessors!  
Initializing Plug-ins!  
Parsing Rules file "/etc/snort/snort.conf"  
PortVar 'HTTP_PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]  
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]  
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]  
PortVar 'SSH_PORTS' defined : [ 22 ]  
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]  
PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ]  
PortVar 'FILE_DATA_PORTS' defined : [ 80:81 110 143 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002 55555 ]
```

## Kiểm tra log của Snort:



The screenshot shows a terminal window with the following output:

```
nguyenquockhanh@ubuntu: ~  
nguyenquockhanh@ubuntu:~$ clear  
nguyenquockhanh@ubuntu:~$ sudo snort -l /var/log/snort/  
Running in packet logging mode  
  
==== Initializing Snort ====  
Initializing Output Plugins!  
Log directory = /var/log/snort/  
pcap DAQ configured to passive.  
Acquiring network traffic from "ens33".  
Decoding Ethernet  
  
==== Initialization Complete ====  
  
-*> Snort! <*-  
Version 2.9.7.0 GRE (Build 149)  
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team  
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.  
Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
Using libpcap version 1.7.4  
Using PCRE version: 8.38 2015-11-23  
Using ZLIB version: 1.2.8  
  
Commencing packet processing (pid=43246)  
  
nguyenquockhanh@ubuntu: ~  
nguyenquockhanh@ubuntu:~$ date  
Fri Mar 10 15:02:54 +07 2023  
nguyenquockhanh@ubuntu:~$
```

**B20DCAT103**

**Nguyễn Quốc Khánh**

Thay đổi mật khẩu

KhanhNQ.B20AT103

KhanhNQ.B20AT103

**B20DCAT103**

**Nguyễn Quốc Khánh**

Thay đổi mật khẩu

KhanhNQ.B20AT103

KhanhNQ.B20AT103



### Bước 3: Tạo rule cho Snort để phát hiện 3 dạng tấn công

```
nguyenquockhanh@ubuntu:~$ cd /etc/snort/rules/  
nguyenquockhanh@ubuntu:/etc/snort/rules$ sudo gedit B20DCAT103.rules  
[sudo] password for nguyenquockhanh:
```

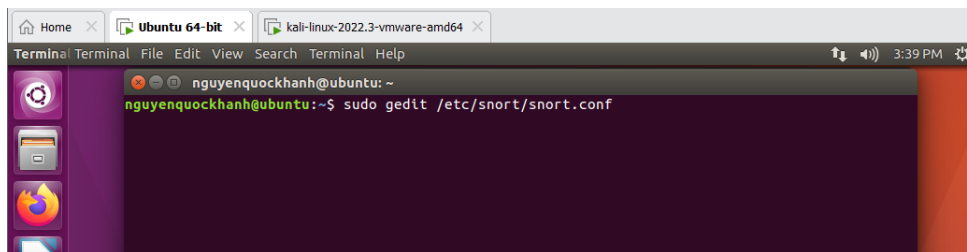


```
*B20DCAT103.rules  
/etc/snort/rules  
alert icmp any any -> any any (msg:"B20DCAT103-NguyenQuockKhanh-Snort phat hien co cac goi tin Ping den."  
";sid:1)  
alert tcp any any -> any 80 (msg:"B20DCAT103-NguyenQuockKhanh-Snort phat hien co cac goi tin ra quet cong  
80. ";sid:2)  
alert tcp any any -> any any (msg:"B20DCAT103-NguyenQuockKhanh-Snort phat hien dang bi tan cong TCP SYN  
Flood. ";flags: S;threshold: type both, track by_src, count 10, seconds 10;sid:3;rev:1;)
```

**B20DCAT103**

**Nguyễn Quốc Khánh**

Thay đổi mật khẩu

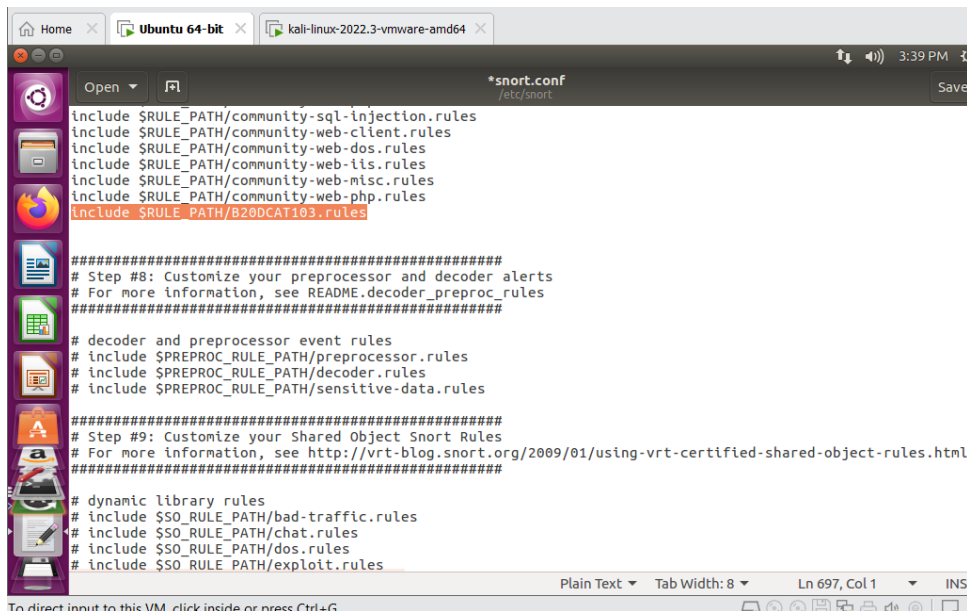


```
nguyenquockhanh@ubuntu:~  
nguyenquockhanh@ubuntu:~$ sudo gedit /etc/snort/snort.conf
```

**B20DCAT103**

**Nguyễn Quốc Khánh**

Thay đổi mật khẩu



```
*snort.conf  
/etc/snort  
include $RULE_PATH/community-sql-injection.rules  
include $RULE_PATH/community-web-client.rules  
include $RULE_PATH/community-web-dos.rules  
include $RULE_PATH/community-web-tls.rules  
include $RULE_PATH/community-web-misc.rules  
include $RULE_PATH/community-web-php.rules  
include $RULE_PATH/B20DCAT103.rules  
#####  
# Step #8: Customize your preprocessor and decoder alerts  
# For more information, see README.decoder_preproc_rules  
#####  
# decoder and preprocessor event rules  
# include $PREPROC_RULE_PATH/preprocessor.rules  
# include $PREPROC_RULE_PATH/decoder.rules  
# include $PREPROC_RULE_PATH/sensitive-data.rules  
#####  
# Step #9: Customize your Shared Object Snort Rules  
# For more information, see http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-rules.html  
#####  
# dynamic library rules  
# include $SO_RULE_PATH/bad-traffic.rules  
# include $SO_RULE_PATH/chat.rules  
# include $SO_RULE_PATH/dos.rules  
# include $SO_RULE_PATH/exploit.rules
```

**B20DCAT103**

**Nguyễn Quốc Khánh**

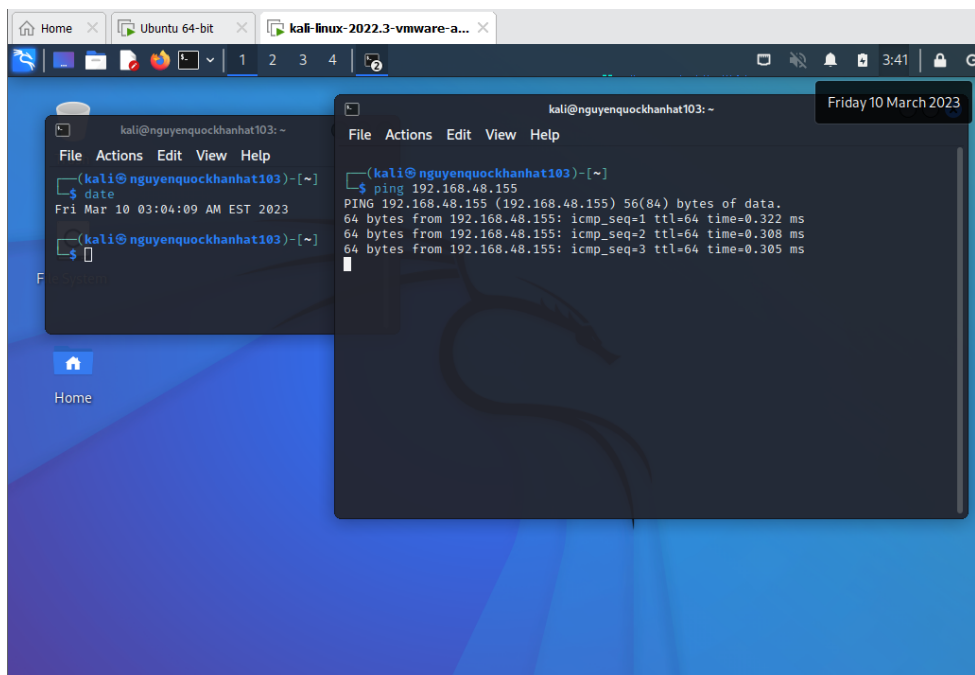
Thay đổi mật khẩu

**KhanhNQ.B20AT103**

**KhanhNQ.B20AT103**

## Bước 4: Thực thi tấn công và phát hiện sử dụng Snort

Máy Kali gửi 3 gói tin ping cho máy Snort



**B20DCAT103**

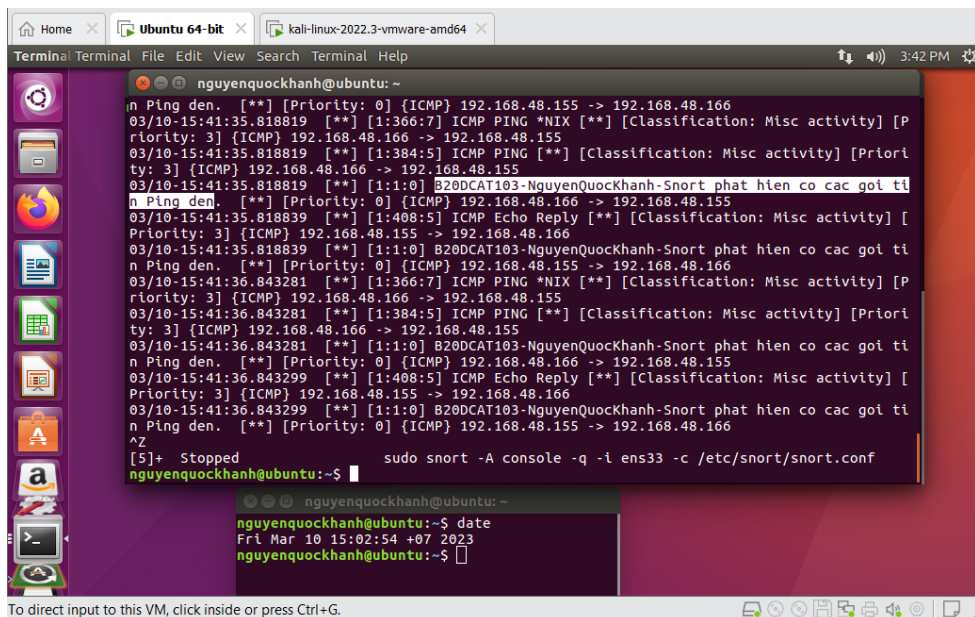
**Nguyễn Quốc Khánh**

Thay đổi mật khẩu

**KhanhNQ.B20AT103**

**KhanhNQ.B20AT103**

Snort phát hiện có các gói Ping gửi đến



**B20DCAT103**

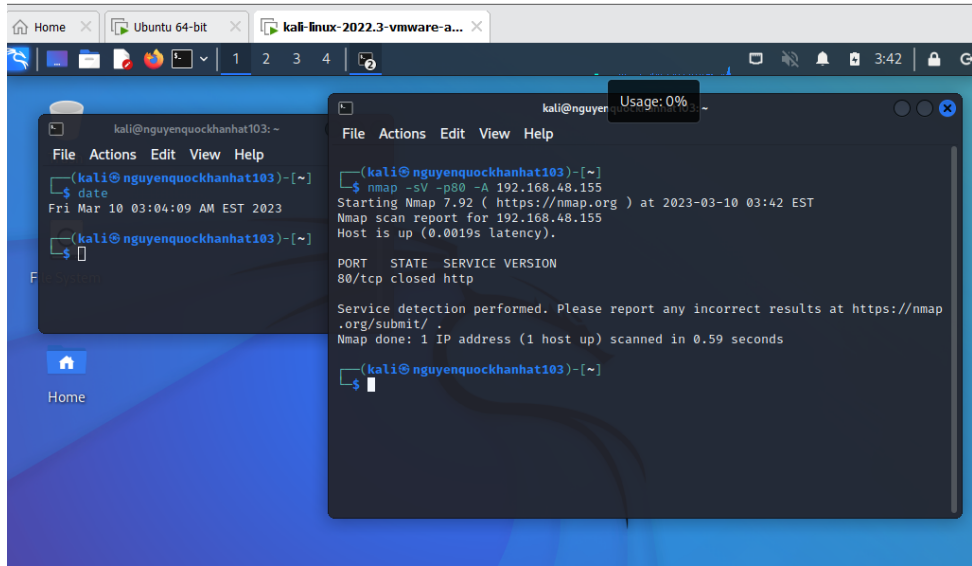
**Nguyễn Quốc Khánh**

Thay đổi mật khẩu

**KhanhNQ.B20AT103**

**KhanhNQ.B20AT103**

## Máy Kali gửi gói tin rà quét cho máy Snort



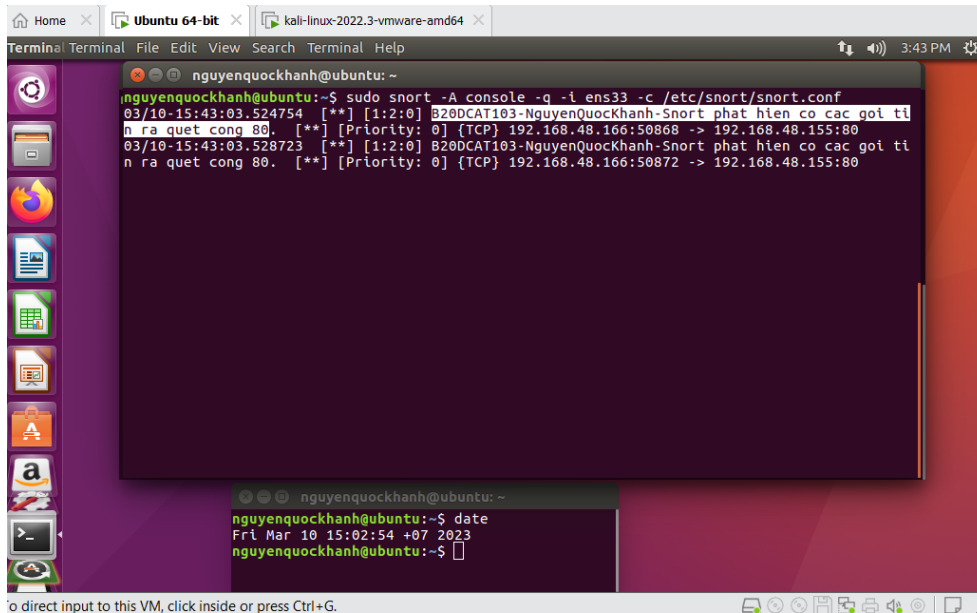
**B20DCAT103**

**Nguyễn Quốc Khánh**

Thay đổi mật khẩu

**KhanhNQ.B20AT103**

## Snort phát hiện có các gói tin rà quét trên cổng 80



**B20DCAT103**

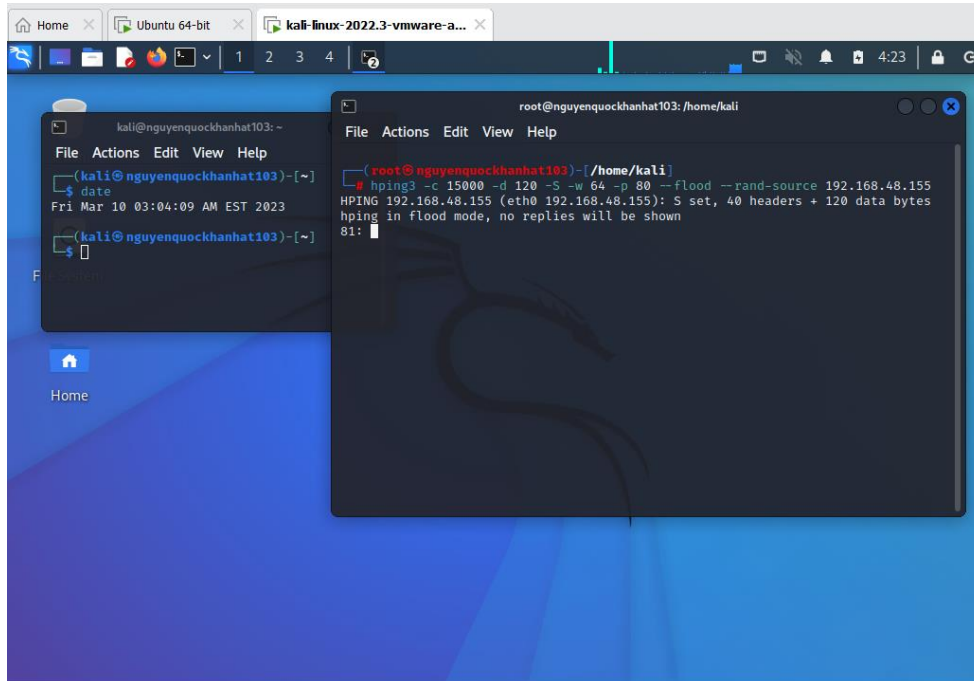
**Nguyễn Quốc Khánh**

Thay đổi mật khẩu

**KhanhNQ.B20AT103**

**KhanhNQ.B20AT103**

## Máy Kali tấn công TCP SYN Flood vào máy Snort



**B20DCAT103**

**Nguyễn Quốc Khánh**

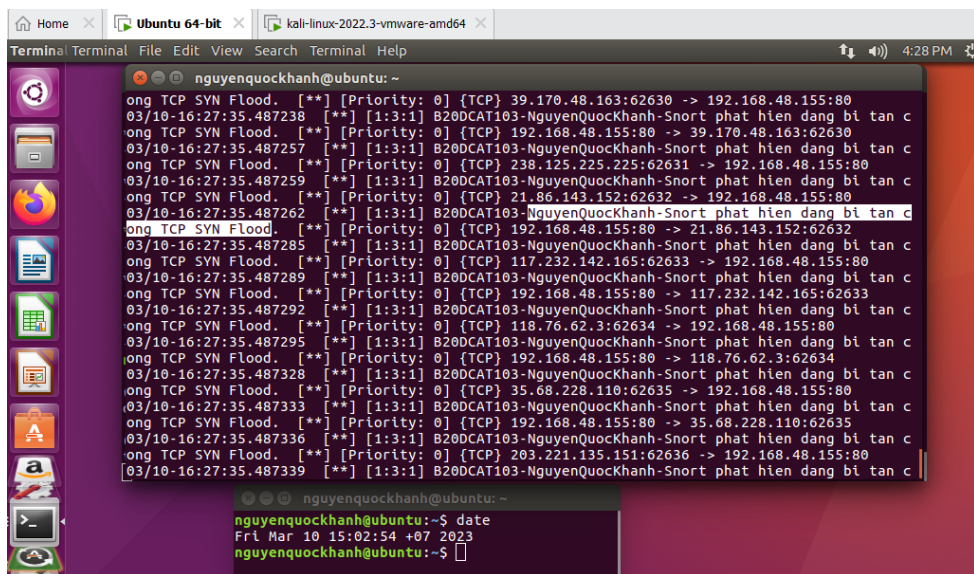
Thay đổi mật khẩu

**KhanhNQ.B20AT103**

**KhanhNQ.B20AT103**

Hủy bỏ

## Snort phát hiện đang bị tấn công TCP SYN Flood



**B20DCAT103**

**Nguyễn Quốc Khánh**

Thay đổi mật khẩu

**KhanhNQ.B20AT103**

**KhanhNQ.B20AT103**