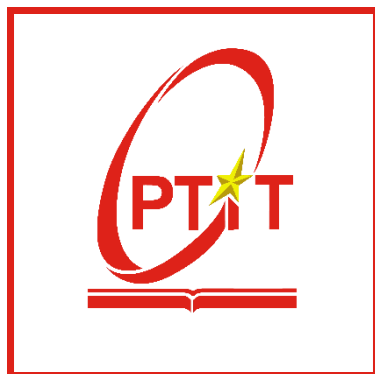


HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN



THỰC TẬP CƠ SỞ
Bài 6: Cài đặt và cấu hình HIDS/NIDS

Sinh viên	Nguyễn Duy Đạt
MSV	B21DCAT056
Giảng viên	Vũ Minh Mạnh

Hà Nội – 2024

Môn học Thực tập cơ sở

Bài 6: Cài đặt và cấu hình HIDS/NIDS

I. Lý thuyết

1. Khái quát về HIDS/NIDS

IDS (Intrusion Detection Systems - Hệ thống phát hiện xâm nhập) là thiết bị hoặc phần mềm có nhiệm vụ giám sát traffic mạng, các hành vi đáng ngờ và cảnh báo cho admin hệ thống. Mục đích của IDS là phát hiện và ngăn ngừa các hành động phá hoại bảo mật hệ thống, hoặc những hành động trong tiến trình tấn công như dò tìm, quét các cổng. IDS cũng có thể phân biệt giữa những cuộc tấn công nội bộ (từ chính nhân viên hoặc khách hàng trong tổ chức) và tấn công bên ngoài (từ hacker). Trong một số trường hợp, IDS có thể phản ứng lại với các traffic bất thường/độc hại bằng cách chặn người dùng hoặc địa chỉ IP nguồn truy cập mạng.

Phân loại IDS

IDS có nhiều loại và tiếp cận các traffic đáng ngờ theo nhiều cách khác nhau. Có IDS dựa trên mạng (NIDS) và dựa trên máy chủ (HIDS). Có IDS phát hiện dựa trên việc tìm kiếm các chữ ký cụ thể của những mối đe dọa đã biết (tương tự như cách các phần mềm diệt virus phát hiện và ngăn chặn malware) và IDS phát hiện bằng cách so sánh các mẫu traffic với baseline rồi xem có sự bất thường nào không. Có những IDS chỉ theo dõi và cảnh báo, có IDS sẽ thực hiện hành động khi phát hiện xâm nhập. Chúng ta sẽ xem xét cụ thể dưới đây:

- **NIDS: Network Intrusion Detection Systems** được đặt tại một điểm chiến lược hoặc những điểm giám sát traffic đến và đi từ tất cả các thiết bị trên mạng. Lý tưởng nhất là bạn có thể quét tất cả traffic inbound và outbound, nhưng việc này có thể tạo ra nút thắt cổ chai làm giảm tốc độ chung của mạng.
- **HIDS: Host Intrusion Detection Systems**, hệ thống phát hiện xâm nhập này chạy trên máy chủ riêng hoặc một thiết bị đặc biệt trên mạng. HIDS chỉ giám sát các gói dữ liệu inbound và outbound từ thiết bị và cảnh báo người dùng hoặc quản trị viên về những hoạt động đáng ngờ được phát hiện.
- **Signature-Based:** Là các IDS hoạt động dựa trên chữ ký, giám sát các gói tin trên mạng và so sánh chúng với cơ sở dữ liệu chữ ký, thuộc tính từ những mối đe dọa đã biết, tương tự như cách phần mềm diệt virus hoạt động. Vấn đề đối với hệ thống IDS này là có

thể không phát hiện ra mối đe dọa mới, khi chữ ký để nhận biết nó chưa được IDS kịp cập nhật.

- Anomaly-Based: IDS này sẽ phát hiện mối đe dọa dựa trên sự bất thường. Nó giám sát traffic mạng và so sánh với baseline đã được thiết lập. Baseline sẽ xác định đâu là mức bình thường của mạng: loại băng thông thường được dùng, giao thức thường dùng, cổng và thiết bị thường kết nối với nhau, cảnh báo cho quản trị viên mạng hoặc người dùng khi phát hiện traffic truy cập bất thường hoặc những khác biệt đáng kể so với baseline.
- Passive: IDS thụ động sẽ chỉ phát hiện và cảnh báo. Khi phát hiện traffic đáng ngờ hoặc độc hại, nó sẽ tạo cảnh báo và gửi đến quản trị viên hoặc người dùng. Việc hành động như nào sau đó tùy thuộc vào người dùng và quản trị viên.
- Reactive: Loại IDS này bên cạnh nhiệm vụ như IDS Passive, nó còn thực hiện những hành động được thiết lập sẵn để ngay lập tức phản ứng lại các mối đe dọa, ví như: chặn truy cập, khóa IP.

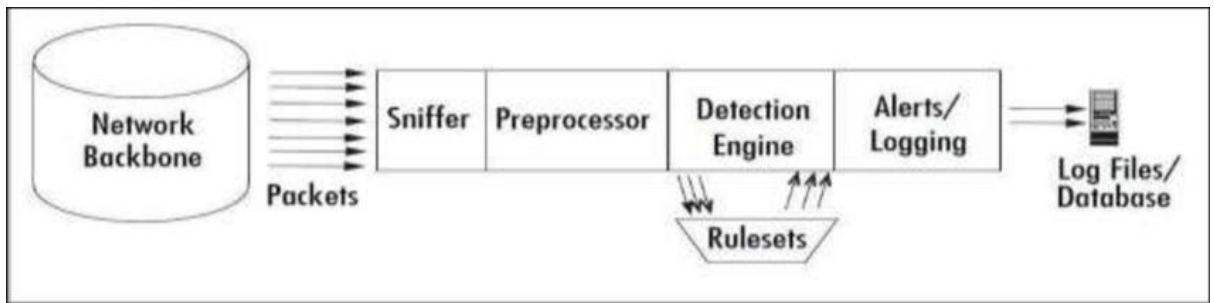
2. Tìm hiểu về Snort

Snort là phần mềm IDS được phát triển bởi Martin Roesch dưới dạng mã nguồn mở. Snort ban đầu được xây dựng trên nền Unix nhưng sau đó phát triển sang các nền tảng khác. Snort được đánh giá rất cao về khả năng phát hiện xâm nhập. Tuy snort miễn phí nhưng nó lại có rất nhiều tính năng tuyệt vời. Với kiến trúc kiểu module, người dùng có thể tự tăng cường tính năng cho hệ thống Snort của mình. Snort có thể chạy trên nhiều hệ thống như Windows, Linux, OpenBSD, FreeBSD, Solaris. Bên cạnh việc có thể hoạt động như một ứng dụng bắt gói tin thông thường, Snort còn được cấu hình để chạy như một NIDS.

Snort bao gồm nhiều thành phần, mỗi phần có một chức năng riêng biệt

- Module giải mã gói tin
- Module tiền xử lý
- Module phát hiện
- Module log và cảnh báo
- Module kết xuất thông tin

Kiến trúc của Snort được thể hiện qua mô hình sau:



Khi Snort hoạt động, nó sẽ lắng nghe tất cả các gói tin nào di chuyển qua nó. Các gói tin sau khi bị bắt sẽ được đưa vào module giải mã. Tiếp theo sẽ vào module tiền xử lý và rồi module phát hiện. Tại đây tùy vào việc có phát hiện được xâm nhập hay không mà gói tin có thể bỏ qua để lưu thông tin tiếp hoặc đưa vào module Log và cảnh báo để xử lý. Khi các cảnh báo được xác định, Module kết xuất thông tin sẽ thực hiện việc đưa ra cảnh báo theo đúng định dạng mong muốn.

II. Cài đặt bài thực hành

1. Cài đặt Snort

- Cài đặt thành công Snort

```
datnd@NguyenDuyDat-B21DCAT056:~$ snort --version
--> Snort! <*-
Version 2.9.15.1 GRE (Build 15125)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11

datnd@NguyenDuyDat-B21DCAT056:~$ echo "Nguyen Duy Dat B21DCAT056"
Nguyen Duy Dat B21DCAT056

datnd@NguyenDuyDat-B21DCAT056:~$ date
Thứ sáu, 15 Tháng 3 năm 2024 17:51:46 +07

datnd@NguyenDuyDat-B21DCAT056:~$
```

- Kiểm tra Snort hoạt động bình thường

```
Activities Terminal Thg 3 15 17:54
datnd@NguyenDuyDat-B21DCAT056: ~

--== Initialization Complete ==--

-*> Snort! <*-
Version 2.9.15.1 GRE (Build 15125)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11

Rules Engine: SF_SNORT DETECTION ENGINE Version 3.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: appid Version 1.1 <Build 5>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>

Snort successfully validated the configuration!
Snort exiting
datnd@NguyenDuyDat-B21DCAT056: $ echo "Nguyen Duy Dat B21DCAT056"
Nguyen Duy Dat B21DCAT056
datnd@NguyenDuyDat-B21DCAT056: $ date
Thứ sáu, 15 Tháng 3 năm 2024 17:54:02 +07
datnd@NguyenDuyDat-B21DCAT056: $
```

2. Tạo các luật Snort

```
bai2ttcsB21DCAT056 - VMware Workstation
File Edit View VM Tabs Help
bai2ttcsB21DCAT056 x kali-linux-2024.1-vmware-amd64 x
Activities Terminal Thg 3 15 18:32
datnd@NguyenDuyDat-B21DCAT056: ~

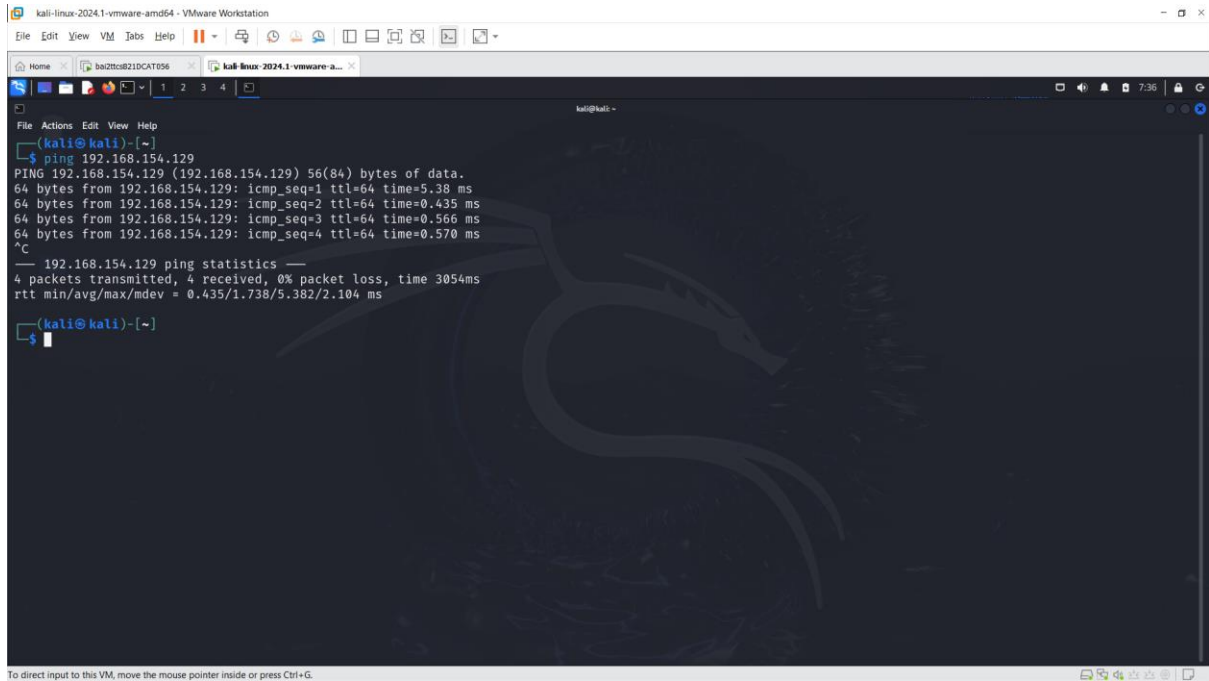
alert icmp any any -> 192.168.154.129 any (msg:"B21DCAT056-NguyenDuyDat-Snort phát hiện có các gói ping gửi đến"; sid:100001; rev:1;)
alert tcp any any -> 192.168.154.129 80 (msg:"B21DCAT056-NguyenDuyDat-Snort phát hiện có các gói tin rà quét trên cổng 80."; sid:100002; rev:1;)
alert tcp any any -> 192.168.154.129 any (flags: S; threshold: type both, track by_src, count 100, seconds 10; msg:"B21DCAT056-NguyenDuyDat-Snort phát hiện đang bị tấn công TCP SYN Flood."; sid: 100003; rev:1;)

#Nguyen Duy Dat B21DCAT056

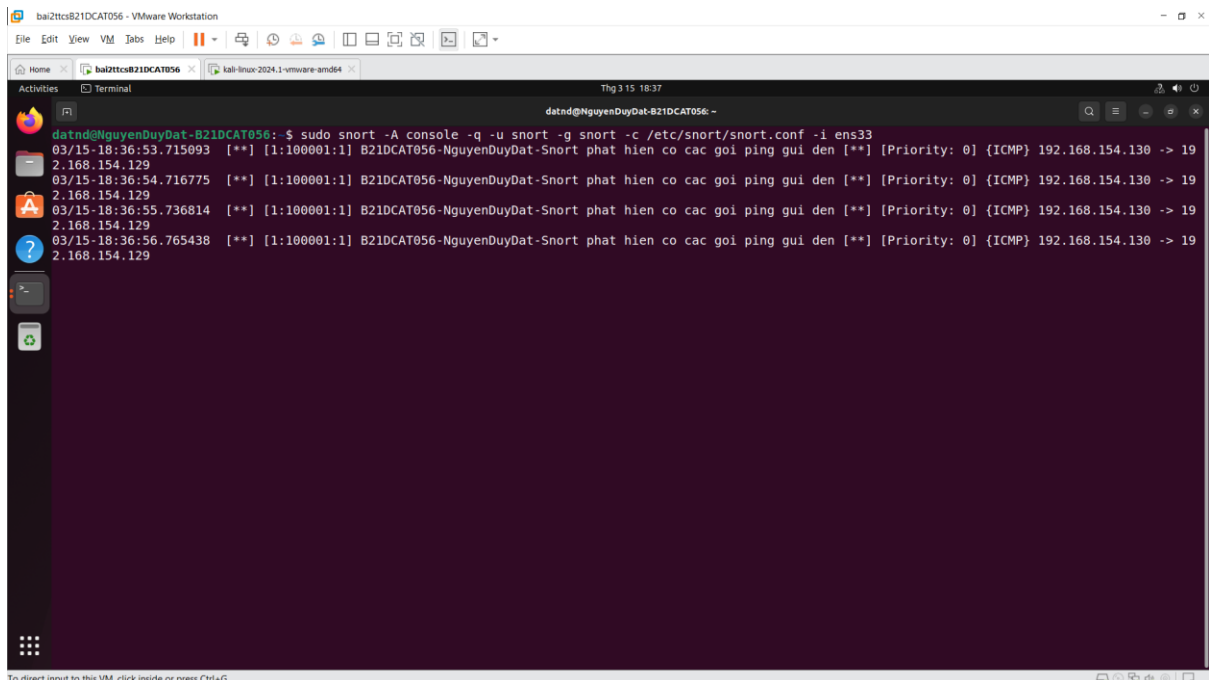
-- INSERT --
7,27 All
```

3. Thực thi tấn công và phát hiện sử dụng Snort

- Phát hiện ping

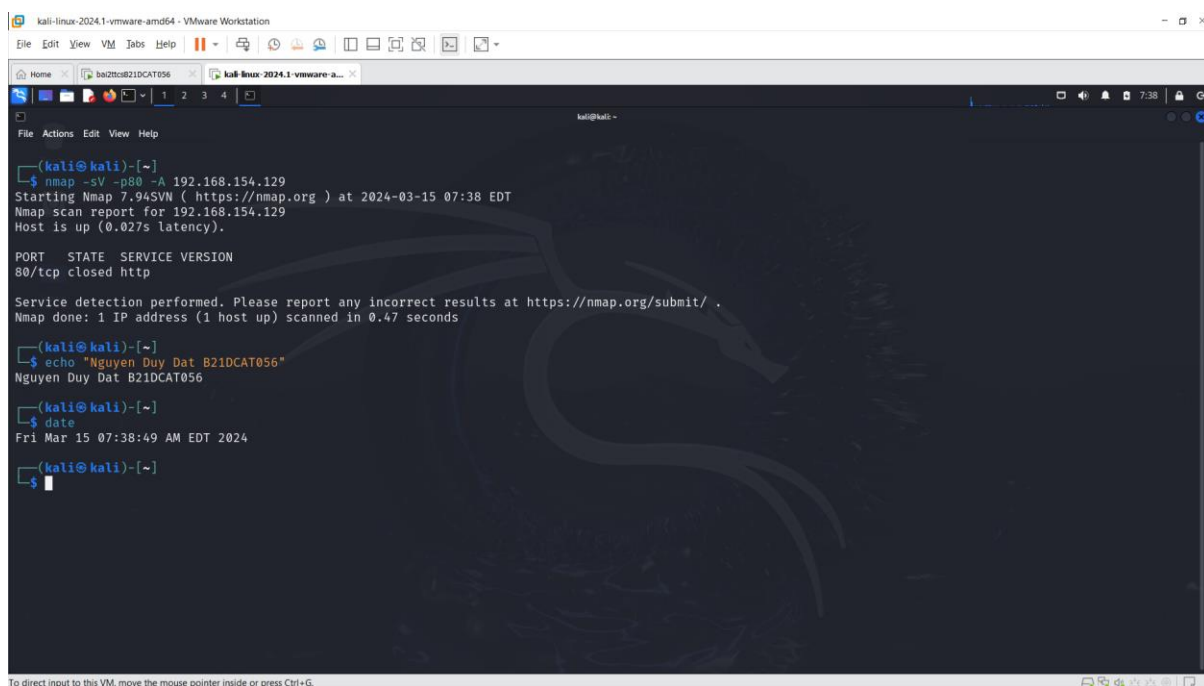


```
kali@kali:~$ ping 192.168.154.129
PING 192.168.154.129 (192.168.154.129) 56(84) bytes of data:
64 bytes from 192.168.154.129: icmp_seq=1 ttl=64 time=5.38 ms
64 bytes from 192.168.154.129: icmp_seq=2 ttl=64 time=0.435 ms
64 bytes from 192.168.154.129: icmp_seq=3 ttl=64 time=0.566 ms
64 bytes from 192.168.154.129: icmp_seq=4 ttl=64 time=0.570 ms
^C
--- 192.168.154.129 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.435/1.738/5.382/2.104 ms
kali@kali:~$
```



```
datnd@NguyenDuyDat-B21DCAT056:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
03/15-18:36:53.715093 [**] [1:100001:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi ping gui den [**] {Priority: 0} {ICMP} 192.168.154.130 -> 192.168.154.129
03/15-18:36:54.716775 [**] [1:100001:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi ping gui den [**] {Priority: 0} {ICMP} 192.168.154.130 -> 192.168.154.129
03/15-18:36:55.736814 [**] [1:100001:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi ping gui den [**] {Priority: 0} {ICMP} 192.168.154.130 -> 192.168.154.129
03/15-18:36:56.765438 [**] [1:100001:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi ping gui den [**] {Priority: 0} {ICMP} 192.168.154.130 -> 192.168.154.129
```

- Phát hiện gói tin rà quét trên cổng 80



```
kali@kali:~$ nmap -sV -p80 -A 192.168.154.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 07:38 EDT
Nmap scan report for 192.168.154.129
Host is up (0.027s latency).

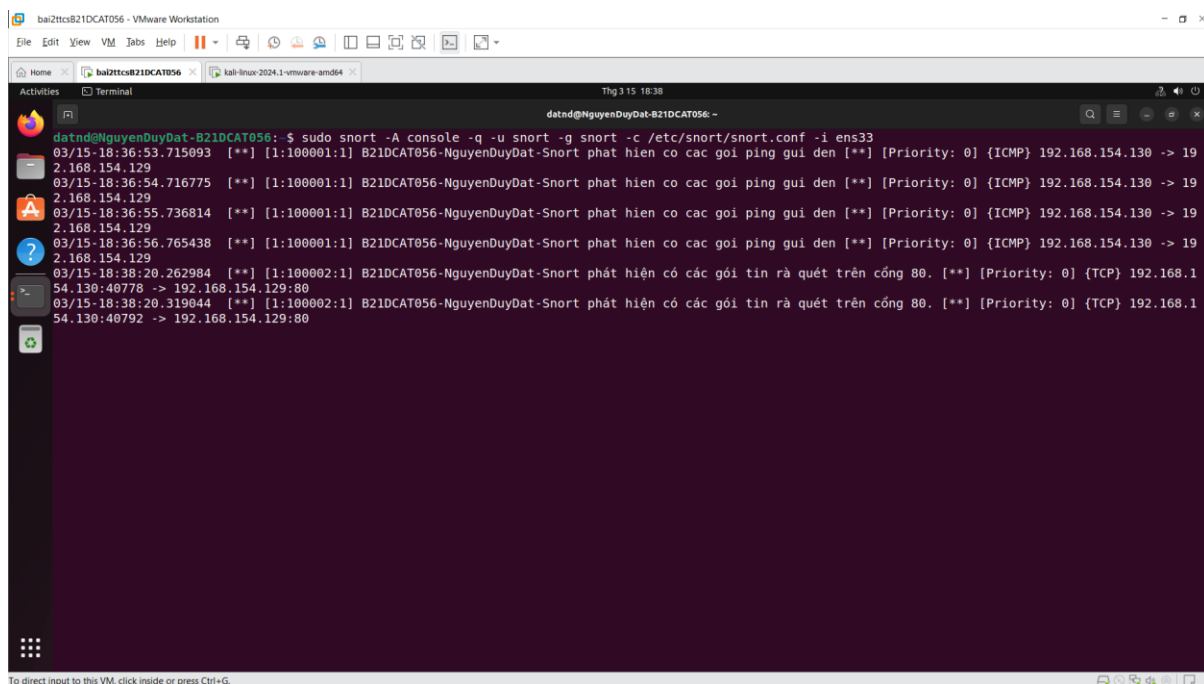
PORT      STATE SERVICE VERSION
80/tcp    closed http

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds

(kali@kali)-[~]
$ echo "Nguyen Duy Dat B21DCAT056"
Nguyen Duy Dat B21DCAT056

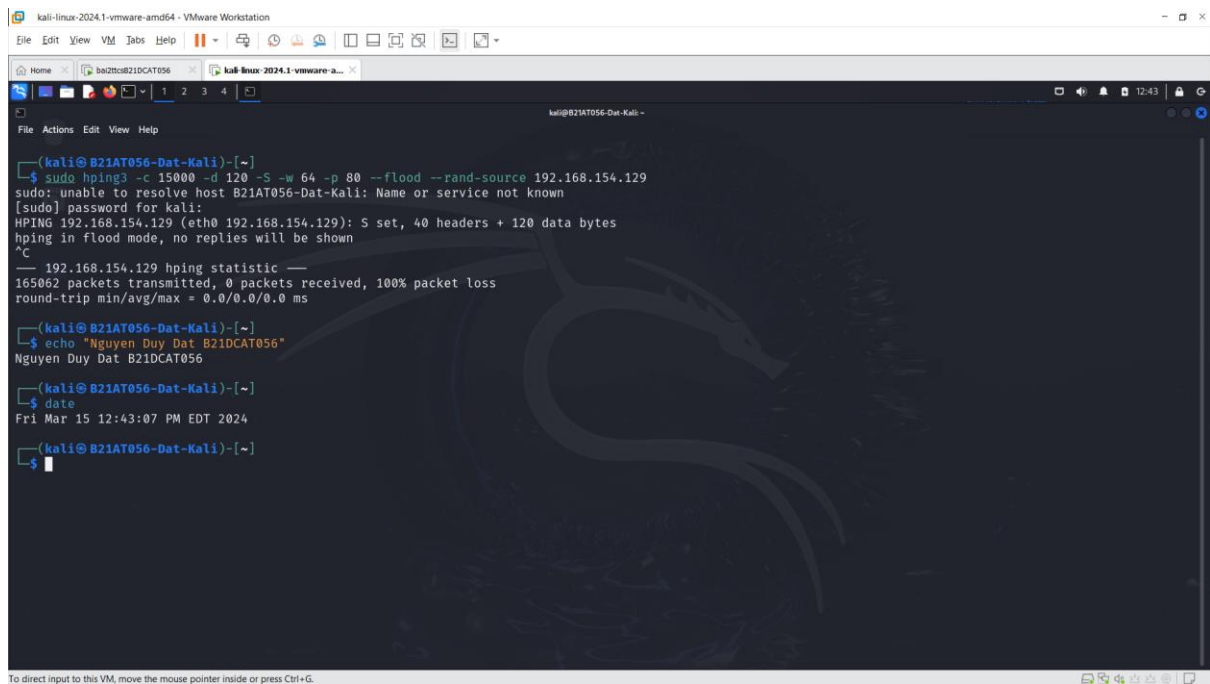
(kali@kali)-[~]
$ date
Fri Mar 15 07:38:49 AM EDT 2024

(kali@kali)-[~]
$
```



```
datnd@NguyenDuyDat-B21DCAT056:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
03/15-18:36:53.715093  [**] [1:100001:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi ping gui den [**] [Priority: 0] {ICMP} 192.168.154.130 -> 192.168.154.129
03/15-18:36:54.716775  [**] [1:100001:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi ping gui den [**] [Priority: 0] {ICMP} 192.168.154.130 -> 192.168.154.129
03/15-18:36:55.736814  [**] [1:100001:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi ping gui den [**] [Priority: 0] {ICMP} 192.168.154.130 -> 192.168.154.129
03/15-18:36:56.765438  [**] [1:100001:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi ping gui den [**] [Priority: 0] {ICMP} 192.168.154.130 -> 192.168.154.129
03/15-18:38:20.262984  [**] [1:100002:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi tin ra quet tren cong 80. [**] [Priority: 0] {TCP} 192.168.154.130:40778 -> 192.168.154.129:80
03/15-18:38:20.319044  [**] [1:100002:1] B21DCAT056-NguyenDuyDat-Snort phat hien co cac goi tin ra quet tren cong 80. [**] [Priority: 0] {TCP} 192.168.154.130:40792 -> 192.168.154.129:80
```

- Phát hiện tấn công TCP SYN Flood



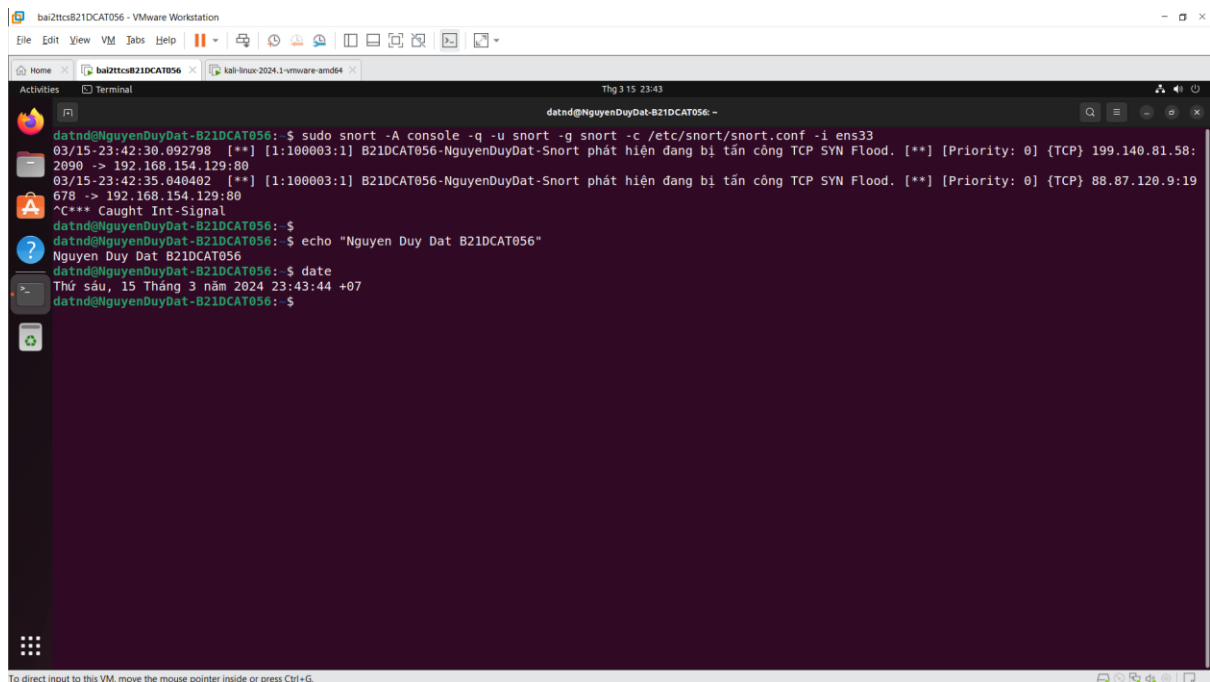
The screenshot shows a Kali Linux terminal window titled "kali@B21AT056-Dat-Kali". The user runs the command `sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.154.129`. The output shows that the host "B21AT056-Dat-Kali" cannot be resolved, and the hping3 tool is used to flood the target. The statistics show 165062 packets transmitted, 0 packets received, and 100% packet loss. The user then runs `echo "Nguyen Duy Dat B21DCAT056"` and `date`, showing the current date and time as "Fri Mar 15 12:43:07 PM EDT 2024".

```
(kali@B21AT056-Dat-Kali)-[~]
$ sudo hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.154.129
sudo: unable to resolve host B21AT056-Dat-Kali: Name or service not known
[sudo] password for kali:
HPING 192.168.154.129 (eth0 192.168.154.129): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
-- 192.168.154.129 hping statistic --
165062 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

(kali@B21AT056-Dat-Kali)-[~]
$ echo "Nguyen Duy Dat B21DCAT056"
Nguyen Duy Dat B21DCAT056

(kali@B21AT056-Dat-Kali)-[~]
$ date
Fri Mar 15 12:43:07 PM EDT 2024

(kali@B21AT056-Dat-Kali)-[~]
$
```



The screenshot shows a Kali Linux terminal window titled "datnd@NguyenDuyDat-B21DCAT056". The user runs the command `sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33`. The output shows a snort alert for a TCP SYN Flood attack. The alert details include the source IP "192.168.154.129", the destination IP "192.168.154.129", and the port "80". The user then runs `echo "Nguyen Duy Dat B21DCAT056"` and `date`, showing the current date and time as "Thứ sáu, 15 Tháng 3 năm 2024 23:43:44 +07".

```
datnd@NguyenDuyDat-B21DCAT056:~$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
03/15-23:42:30.092798  [**] [1:100003:1] B21DCAT056-NguyenDuyDat-Snort phát hiện đang bị tấn công TCP SYN Flood. [**] [Priority: 0] {TCP} 192.168.154.129:80
2090 -> 192.168.154.129:80
03/15-23:42:35.040402  [**] [1:100003:1] B21DCAT056-NguyenDuyDat-Snort phát hiện đang bị tấn công TCP SYN Flood. [**] [Priority: 0] {TCP} 88.87.120.9:19
678 -> 192.168.154.129:80
^C*** Caught Int-Signal
datnd@NguyenDuyDat-B21DCAT056:~$
datnd@NguyenDuyDat-B21DCAT056:~$ echo "Nguyen Duy Dat B21DCAT056"
Nguyen Duy Dat B21DCAT056
datnd@NguyenDuyDat-B21DCAT056:~$ date
Thứ sáu, 15 Tháng 3 năm 2024 23:43:44 +07
datnd@NguyenDuyDat-B21DCAT056:~$
```