HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA AN TOÀN THÔNG TIN

Lab: idr_suricata_c2port

Sinh viên thực hiện: Đào Ngọc Ánh

Mã sinh viên: B21DCAT038

Hà Nội 2025

Phát hiện và ứng phó: C2 Beacon qua Non-Standard Port

1.1.1. Giới thiệu chung

Bài thực hành "Phát hiện và phản ứng C2 Beacon qua cổng bất thường (non-standard port)" được xây dựng nhằm giúp sinh viên hiểu cách attacker che giấu C2 traffic, và cách triển khai hệ thống phát hiện dựa trên Suricata + SIEM, đồng thời mô phỏng phản ứng khi bị tấn công như block port, kill process.

1.1.2. Mục đích

Sau khi hoàn thành bài lab, sinh viên sẽ nắm được:

- Hiểu cách attacker sử dụng cổng non-standard để né firewall.
- Hiểu cơ chế beacon C2 hoạt động định kỳ để duy trì kết nối với máy bị nhiễm.
- Nắm được dấu hiệu nhận diện traffic bất thường: payload nhỏ, đều thời gian, kết nối lặp đến cùng IP.
- Chạy Suricata ở chế độ sensor để giám sát traffic.
- Phân tích log từ Suricata & SIEM để phát hiện hoạt động C2.
- Tự viết Suricata rule để phát hiện C2 beacon trên cổng lạ.
- Phân tích hành vi beacon dựa trên:
    - Port bất thường
    - Chu kỳ gửi đều đặn
    - Payload nhỏ, base64
    - Destination IP cố định
- Nhận biết đường tấn công C2 (adversary kill-chain).
- Thực hiện chặn port / C2 traffic bằng iptables
- Có thể kill process beacon hoặc cô lập máy victim.
- Ghi lại incident timeline phục vụ điều tra số.

1.1.3. Yêu cầu đối với sinh viên

Sinh viên cần:

- Hiểu về HTTP request, TCP, netstat.
- Biết cơ bản Python (đọc hiểu được beacon script).
- Biết tường lửa cơ bản (iptables).
- Có kiến thức SIEM/sensor cơ bản (Suricata, log file).

1.1.4. Nội dung thực hành

- Tải bài lab:

  imodule https://raw.githubusercontent.com/anhdnmit/do_an_tot_nghiep/main/idr_suricata_c2port/idr_suricata_c2port.tar

- Khởi động bài lab:

  labtainer -r idr_suricata_c2port

(chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)

- Sau khi chạy, sẽ xuất hiện 4 container:

attacker	Máy C2 server lắng nghe beacon
victim	Máy bị cài beacon gửi tín hiệu
gateway	Máy Suricata giám sát & phân tích
siem	Máy đọc log từ gateway

- Để kiểm tra địa chỉ IP của 4 máy, gõ lệnh sau trên từng máy:

  ifconfig

1.1.4.1. *Cài đặt suricata*

- Tại container gateway, sinh viên thực hiện các lệnh sau để tiến hành cài đặt Suricata – công cụ IDS dùng để giám sát và phát hiện beacon C2.

  sudo apt-get update

  sudo apt-get install software-properties-common

- Thêm repository Suricata chính thức

  sudo add-apt-repository ppa:oisf/suricata-stable

  Sau khi chạy lệnh này, hệ thống sẽ yêu cầu nhấn Enter để xác nhận.

  Sinh viên nhấn Enter để tiếp tục cài đặt repository.

- Cập nhật lại danh sách package

  sudo apt-get update

- Cài đặt Suricata

sudo apt-get install suricata

Sau khi lệnh hoàn tất, Suricata sẽ được cài đặt vào hệ thống.

- Sinh viên có thể kiểm tra phiên bản bằng lệnh:

suricata -V

1.1.4.2. *Tạo C2 server (attacker)*

- Trong bài lab này, container attacker sẽ đóng vai trò máy chủ Command & Control (C2). Đây là nơi beacon từ victim gửi về để báo rằng máy đã bị nhiễm và vẫn đang hoạt động.
- Sau khi vào container attacker, sinh viên thực hiện các bước sau:
- Di chuyển đến đúng thư mục chứa mã nguồn

cd /home/ubuntu

ls

- Sinh viên sẽ nhìn thấy file c2_server.py – đây chính là mã Python mô phỏng C2 server.
- Chạy lệnh sau để bật C2 server ở chế độ nền (background):

python3 c2_server.py &

ps aux | grep c2_server > /tmp/ps.stdout

- Sau khi chạy server, sinh viên theo dõi log bằng lệnh:

tail -f c2.log

- Nếu beacon hoạt động đúng, log sẽ xuất hiện các dòng như:

Beacon received: b'eyJpZCI6ICJ2aWN0aW0wMSIsICJ2IjogIjEuMCJ9'

Đây là tín hiệu cho thấy beacon từ victim đang cố kết nối tới C2 server.

1.1.4.3. *Khởi chạy Beacon (Trên máy victim)*

- Sau khi vào container victim, sinh viên nhập lệnh sau để di chuyển về đúng thư mục:

cd /home/ubuntu

ls

Sinh viên sẽ nhìn thấy tệp: beacon.py

Đây là mã nguồn Python đã được cung cấp sẵn, mô phỏng malware duy trì kết nối với máy C2.

- Sinh viên có thể kiểm tra nội dung beacon.py (tùy chọn nhưng khuyến khích)

cat beacon.py

- Khởi chạy beacon ở chế độ nền (background process)

sudo apt update

sudo apt install python3-requests -y

python3 beacon.py &

Lệnh này sẽ chạy beacon.py dưới nền, tiếp tục gửi beacon sau mỗi 30 giây.

- Kiểm tra beacon có đang chạy hay không

ps aux | grep beacon > ps.stdout

Nếu beacon chạy đúng, sẽ xuất hiện một dòng tương tự:

root 120 0.0 python3 /home/ubuntu/beacon.py

- Kiểm tra victim có kết nối đến C2 hay không

netstat -ant | grep 45678 > netstat.stdout

Nếu beacon hoạt động, sẽ có kết nối tới C2 (attacker):

tcp 0 0 172.20.0.30:54210 172.20.0.10:45678 ESTABLISHED

1.1.4.4. Khởi động suricata

Suricata được triển khai trên container gateway đóng vai trò IDS – hệ thống phát hiện xâm nhập.

Sinh viên khởi động Suricata bằng lệnh:

sudo suricata -c /etc/suricata/suricata.yaml -i eth0 &

Sau đó, theo dõi log của Suricata bằng:

sudo tail -f /var/log/suricata/fast.log

- Tại thời điểm này, Suricata chưa có rule phát hiện C2, nên log có thể chưa hiển thị alert.

1.1.4.5. *Viết rule Suricata phát hiện C2 Beacon (bắt buộc – tự thực hiện)*

- Tạo thư mục chứa rule (nếu chưa có)

sudo mkdir -p /etc/suricata/rules

- Suricata cho phép viết rule thủ công để phát hiện các dấu hiệu tấn công.

Sinh viên mở file rule bằng lệnh:

sudo nano /etc/suricata/rules/local.rules

- Thêm rule sau vào file (Thêm rule DƯỚI ĐÂY – CHỈ ĐƯỢC VIẾT TRÊN 1 DÒNG DUY NHẤT**:**)

alert http any any -> 172.20.0.10 45678 (msg:"C2 Beacon detected"; flow:established,to_server; sid:900001; rev:1;)

Ý nghĩa rule:

alert http any any -> 172.20.0.10 45678	Giám sát tất cả HTTP traffic tới IP C2 và port 45678
flow:established,to_server	Chỉ bắt các kết nối đã thiết lập tới server
msg:	Thông báo khi phát hiện beacon
sid:900001	Mã định danh rule (bắt buộc phải là số duy nhất)

Sau khi lưu file (Ctrl + O, Enter, Ctrl + X), Suricata sẽ dùng rule này để phát hiện beacon.

- Kiểm tra cấu hình Suricata đang load đúng rule chưa

sudo nano /etc/suricata/suricata.yaml

Tìm đến phần `rule-files:`

Nếu thấy dòng:

rule-files:

- *suricata.rules*

→ Sửa lại như sau :

rule-files:

- */etc/suricata/rules/local.rules*

Sau đó lưu file rồi thoát

- Kiểm tra rule có load được không

sudo suricata -T -c /etc/suricata/suricata.yaml

Nếu rule đúng → sẽ hiện:

Configuration provided was successfully loaded.

Nếu sai, sẽ báo lỗi rất rõ → sinh viên cần sửa lại dựa trên thông báo.

- Khởi động Suricata để áp dụng rule

sudo suricata -c /etc/suricata/suricata.yaml -i eth0 &

- Kiểm tra Suricata phát hiện beacon hay chưa

sudo tail -f /var/log/suricata/fast.log

1.1.4.6. *Khởi chạy SIEM giả lập (Trên máy siem)*

- Sau khi vào container siem, sinh viên nhập lệnh sau để di chuyển về đúng thư mục:

cd /home/ubuntu

ls

Sinh viên sẽ nhìn thấy tệp: *siem.py*.

- Sinh viên có thể kiểm tra nội dung siem.py (tùy chọn nhưng khuyến khích)

cat siem.py

Sinh viên sẽ thấy đoạn mã này:

```
import time
print("SIEM started... monitoring logs.")
while True:
    try:
        with open("/var/log/suricata/fast.log", "r") as f:
            lines = f.readlines()
```

```
            for l in lines:
                if "900001" in l:
                    print("[ALERT] Beacon detected: ", l.strip())
        except:
            pass
        time.sleep(5)
```

Ý nghĩa:

Hành vi	Ý nghĩ
Đọc file log của Suricata	Thu thập log từ IDS
Tìm rule SID = 900001	Kiểm tra dấu hiệu tấn công
Nếu có -> In ra ALERT	SIEM phát hiện beacon

- Sinh viên sử dụng lệnh sau để chạy SIEM giả lập:

  python3 siem.py > siem_output.txt &

- Theo dõi output của SIEM:

  tail -f /home/ubuntu/siem_output.txt

- Khi Suricata phát hiện beacon, SIEM sẽ hiển thị:

  [ALERT] Beacon detected: 11/27/2025-10:44:12 172.20.0.30 -> 172.20.0.10:45678

- Điều này chứng minh log từ Suricata đã được phân tích và SIEM đã phát hiện hành vi đáng ngờ.

1.1.4.7. *Phản ứng thủ công (Trên máy gateway)*

- Sau khi xác định beacon là traffic độc, sinh viên thực hiện biện pháp phản ứng bằng cách chặn port C2 trên gateway:

  sudo iptables -A INPUT -p tcp --dport 45678 -j DROP

- Kiểm tra rule firewall vừa thêm:

  iptables -L INPUT -n

- Nếu xuất hiện dòng như:

DROP tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:45678

- Chứng tỏ C2 traffic đã bị chặn thành công.
- Đây là bước mô phỏng ứng phó tấn công trong thực tế

1.1.4.8. *Kết thúc bài lab*

Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab:

stoplab idr_suricata_c2port

Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.

Khởi động lại bài lab:

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

labtainer -r idr_suricata_c2port