**SIEM RULE DETECTION – MITRE ATT&CK**

# 0. Tóm tắt tiến độ

## 0.1. Logs

*Log src:*

- winndows security logs src => ***xong***

- Linux logs src => ***xong***

- Services cơ bản: liên quan web, IIS, nginx -> khi tấn công vào -> xuất ra gì? => ***xong được dịch vụ Nginx bên Ubuntu***

*Log src security agent:*

- Suricata => ***xong***

- Dung them 1 pfsense tuong lua => Gia lap tan cong LAN/WAN => day log ve sime => ***chưa xong***

## 0.2. Detection Rule => *xong*

- Kha nang capabilti

- Gia lap cac kieu tan cong

- hien ket qua

## 0.3. Giả lập một số cuộc tấn công (ví dụ Ransomware mã hoá dữ liệu). Xem Suricata có phát hiện được không?

*=> chưa thực hiện được Ransomware, mới thực hiện được DDoS*

## 0.4. Giả lập tấn công C&C Server. Xem Suricata có phát hiện được không?

*=> Mới tìm hiểu được cơ bản, chưa đi sâu*

## 0.5. Khả năng Auto Respond của Elastic Search như nào?

*=> chưa làm*

# 1. Mô hình triển khai

A diagram of a computer network

AI-generated content may be incorrect.

Mô hình triển khai

## 1.1. Router

- Sử dụng 1 máy pfSense làm bộ định tuyến (Router) kiêm tường lửa (Firewall) trung tâm cho toàn bộ hạ tầng.

- Phân tách và quản lý lưu lượng giữa các mạng (NAT, Host-Only, DMZ…) thông qua các rule định tuyến và NAT.

- Áp dụng chính sách truy cập (firewall rules) để cho phép hoặc chặn dịch vụ theo lớp mạng, IP, port.

- Thu thập và chuyển tiếp tất cả log hệ thống (firewall, DHCP, OpenVPN, DNS…) đến Logstash/SIEM để giám sát và phân tích an ninh.

## 1.2. IDS/IPS

- Triển khai 1 máy Ubuntu Server cài đặt Suricata hoạt động ở chế độ Inline IDS/IPS.

- Suricata kiểm tra lưu lượng mạng theo thời gian thực, phát hiện và ngăn chặn các hành vi xâm nhập hoặc bất thường dựa trên rule signatures.

- Gửi các sự kiện cảnh báo (alert logs) về hệ thống SIEM thông qua Filebeat để phân tích và điều tra chi tiết.

- Định kỳ cập nhật rule từ cộng đồng (ET Open Rules) để đảm bảo khả năng phát hiện các mối đe dọa mới.

## 1.3. Clients

- 1 máy Ubuntu Desktop bình thường, không cài thêm IDS gì thêm, chỉ để Filebeat lấy log hệ thống

- 1 máy Metasploitable2 để khai thác lỗ hỏng

- 1 máy Windows 10, không cài thêm IDS gì thêm, chỉ để Filebeat lấy log hệ thống

## 1.4. Admin

- 1 máy làm ELK Server dùng để lọc, phân tích, lưu trữ log

- 1 máy bên ngoài đăng nhập vào giao diện web của Kibana để quản lý

# 2. Triển khai pfSense

Mục tiêu của việc triển khai pfSense trong mô hình trên là thiết lập một hệ thống firewall mạnh mẽ để bảo vệ các máy chủ và dịch vụ trong môi trường mạng. pfSense sẽ đóng vai trò là bộ lọc chính, kiểm tra và phân tích lưu lượng mạng để ngăn chặn các cuộc tấn công mạng trước khi chúng tiếp cận các tài nguyên mạng quan trọng.

## 2.1. Cấu hình

A screenshot of a computer program

AI-generated content may be incorrect.

- **WAN Interface**: Kết nối với internet hoặc mạng bên ngoài -> 192.168.71.250/24

- **LAN Interface**: gồm 3 interfaces

+ LAN – em1: 10.81.85.2/24 => là gateway cho lớp mạng 10.81.85.0/24

+ OPT1 - em2: 192.168.85.2/24 => là gateway cho lớp mạng 192.168.85.0/24

+ OPT2 – em3: 192.168.100.2/24 => là gateway cho lớp mạng 192.168.100.0/24

- Gửi logs về Filebeat: *<đang hoàn thiện, còn lỗi>*

## 2.2. Một số rules cơ bản

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# 3. Triển khai Suricata Inline IDS/IPS

- Theo mô hình trên, nhóm triển khai Suricata làm Inline IDS/IPS trên máy Ubuntu Server 192.168.71.171/24 để giám sát lớp mạng 192.168.85.0/24 và đẩy Suricata Logs về máy ELK là 192.168.100.128

- Cấu hình inline IDS/IPS trong file /etc/suricata/suricata.yml

af-packet:

  - interface: ens37

    threads: 1

    defrag: no

    cluster-type: cluster\_flow

    cluster-id: 100

    copy-mode: ips

    copy-iface: ens38

    buffer-size: 64535

    use-mmap: yes

    tpacket-v3: no

  - interface: ens38

    threads: 1

    cluster-id: 101

    defrag: no

    cluster-type: cluster\_flow

    copy-mode: ips

    copy-iface: ens37

    buffer-size: 64535

    use-mmap: yes

    tpacket-v3: no

- Ngoài bộ Rules khổng lồ của cộng đồng, nhóm có viết thêm 1 bộ rule ở local nhằm mục đích kiểm thử hệ thống hoạt động ổn không:

|  |
| --- |
| wanthinnn@ubuntuserver:~$ sudo cat /var/lib/suricata/rules/local.rules  #alert tls any any -> any any (msg:"[TEST] TLS traffic detected"; sid:9999999; rev:1;)  # 1.Initial Access  # Ping detected  alert icmp any any -> any any (msg:"[ALERT] ICMP Ping detected"; sid:1000001; rev:1;)  # TCP SYN Flood Detected  alert tcp any any -> any any (msg:"[ALERT] TCP SYN Packet Detected"; flags:S; threshold:type threshold, track by\_src, count 20, seconds 10; sid:1000002; rev:1;)  # UDP Flood Detected  alert udp any any -> any any (msg:"[ALERT] UDP Packet Detected"; threshold:type threshold, track by\_src, count 20, seconds 10; sid:1000003; rev:1;)  # SSH Brute Force Detection  alert tcp any any -> any 22 (msg:"[ALERT] SSH Brute Force Attempt"; flow:to\_server,established; threshold:type threshold, track by\_src, count 5, seconds 60; content:"SSH"; nocase; sid:1000004; rev:1;)  # Path Travel  #drop tcp any any -> 192.168.71.0/24 80 (msg:"[DROP] Potential Path Traversal Attack Detected"; flow:established; content:"GET"; content:"HTTP"; fast\_pattern; content:"../"; nocase; session:all; sid:1000005; rev:1;)  # 2. Resource Development  # Suspicious Tool Download  alert http any any -> any any (msg:"[ALERT] Suspicious Tool Download"; flow:to\_server,established; content:"/\*.exe"; nocase; sid:2000001; rev:1;)  # Suspicious SNI Detected  alert tls any any -> any any (msg:"[ALERT] Suspicious SNI detected in TLS handshake"; tls.sni; content:"suspicious.example.com"; nocase; sid:2000002; rev:1;)  # 3. Reconnaissance  # Banner Grabbing Detection  alert tcp any any -> any 80 (msg:"[ALERT] Banner Grabbing Attempt"; flow:to\_server,established; content:"HTTP/1.1"; nocase; detection\_filter: track by\_src, count 3, seconds 5; sid:3000001; rev:1;)  # Port Scan Detection  alert ip any any -> 192.168.71.129 any (msg:"[ALERT] Port scan detected from other device"; sid:3000002; rev:1;)  # 4. Execution  # Command Injection Attempt  #alert http any any -> any any (msg:"[ALERT] Potential Command Injection Attempt"; flow:to\_server,established; content:"cmd="; nocase; pcre:"/[\;\|\&]\s\*cmd/"; sid:4000001; rev:1;)  alert http any any -> any any (msg:"[ALERT] Potential Command Injection Attempt"; flow:to\_server,established; uricontent:"cmd="; nocase; sid:4000001; rev:1;)  # 5. Persistence  # Suspicious SMB Traffic  alert tcp any any -> any 135 (msg:"[ALERT] Potential RPC/SMB Exploit Attempt"; flow:to\_server,established; content:"|90 90 90|"; offset:0; depth:10; sid:5000001; rev:1;)  # 6. Privilege Escalation  alert tcp any any -> any 135 (msg:"[ALERT] Potential RPC/SMB Exploit Attempt"; flow:to\_server,established; content:"|90 90 90|"; offset:0; depth:10; sid:6000001; rev:1;)  alert tls any any -> any 443 (msg:"[ALERT] TA0004 T1548: Possible abuse of elevation control mechanism detected"; tls.sni; pcre:"/cmd\.exe|powershell\.exe|wscript\.exe|cscript\.exe|regedit\.exe|mshta\.exe|bash\b|sudo\b|su\b|pkexec|gksudo|kdesudo/i"; sid:6000002; rev:5;)  # 7. Defense Evasion  # Suspicious TLS Version  alert tls any any -> any any (msg:"[ALERT] Suspicious TLS Version Used"; tls.version:0x0301; sid:7000001; rev:1;)  # 8. Credential Access  # FTP Login Attempt  alert ftp any any -> any 21 (msg:"[ALERT] FTP Login Attempt"; flow:to\_server,established; content:"USER "; sid:8000001; rev:1;)  alert http any any -> any 443 (msg:"[ALERT] TA0006 Credential Access: Potential credential exposure in Elasticsearch query over HTTPS"; flow:established,to\_server; content:"/elasticsearch/"; nocase; file\_data; content:"password="; nocase; sid:8000002; rev:2;)  # 9. Discovery  # ICMP Ping Sweep  alert icmp any any -> any any (msg:"[ALERT] ICMP Ping Sweep Detected"; threshold:type threshold, track by\_src, count 10, seconds 5; sid:9000001; rev:1;)  # 10. Lateral Movement  # SMB Null Session Scan  alert smb any any -> any any (msg:"[ALERT] SMB Null Session Scan Detected"; content:"|00 00 00|"; depth:4; sid:10000001; rev:1;)  # 11. Collection  # Large FTP File Transfer  alert ftp any any -> any 21 (msg:"[ALERT] Large FTP File Transfer Detected"; flow:to\_server,established; content:"SIZE "; sid:11000001; rev:1;)  # 12. Command and Control  # DNS Tunneling Detection  alert dns any any -> any any (msg:"[ALERT] Potential DNS Tunneling Detected"; dns.query; pcre:"/([a-z0-9]{20,}\.)/i"; sid:12000001; rev:1;)  # 13. Exfiltration  # Unusual HTTP POST Large Data Transfer  alert http any any -> any any (msg:"[ALERT] Unusual HTTP POST Large Data Transfer Detected"; flow:to\_server,established; content:"POST"; content:"Content-Length:"; threshold:type threshold, track by\_src, count 3, seconds 30; sid:13000001; rev:1;)  # 14. Impact  # TCP SYN Flood Detection  alert tcp any any -> any any (msg:"[ALERT] Potential TCP SYN Flood Detected"; flags:S; threshold:type threshold, track by\_src, count 100, seconds 10; sid:14000001; rev:1;) |

- Cấu hình Filebeat.yml:

|  |
| --- |
| wanthinnn@ubuntuserver:~$ sudo cat /etc/filebeat/filebeat.yml  # Filebeat inputs  filebeat.inputs:  # filestream is an input for collecting log messages from files.  - type: filestream  # Unique ID among all inputs, an ID is required.  id: my-filestream-id  # Change to true to enable this input configuration.  enabled: true  - /var/log/suricata/eve.json  # Filebeat modules  filebeat.config.modules:  # Glob pattern for configuration loading  path: ${path.config}/modules.d/\*.yml  # Set to true to enable config reloading  reload.enabled: false  # Elasticsearch template setting  setup.template.settings:  index.number\_of\_shards: 1  # Logstash Output  output.logstash:  # The Logstash hosts  hosts: ["siem-dacn.local:5044"]  # Optional SSL. By default is off.  # List of root certificates for HTTPS server verifications  ssl.certificate\_authorities: ["/etc/filebeat/certs/rootCA.crt"] |

- Chạy Suricata ở chế độ Inline mode:

A computer screen with white text

AI-generated content may be incorrect.

- Sau cấu hình trên, Logs từ Suricata sẽ được đẩy về Filebeat, rồi từ Filebeat sẽ đẩy về Logstash trên máy SIEM Server.

# 4. Triển khai SIEM Server (có cập nhật so với trước đây)

## 4.1. Sơ lược

- Trước đây: sử dụng card NAT với địa chỉ 192.168.71.128

- Bây giờ: Sử dụng Host-only với địa chỉ 192.168.100.128/24

- Mục tiêu: cô lập máy SIEM Server vào 1 LAN riêng, firewall chỉ cho phép giao tiếp TCP/5044, còn lại inbound và outbound khác đều bị chặn và chỉ cho phép truy cập vào Kibana/Logstash/Elasticsearch thông qua tên miền local: siem-dacn.local với giao thức HTTPS.

## 4.2. Luồng dữ liệu

- **Thu thập và gửi log**

+ Trên các máy source (Windows/Linux/Suricata/etc.) đều chạy Filebeat, cấu hình gửi tới https://siem-dacn.local:5044.

+ Filebeat transport qua HTTPS (443) đến Nginx trên SIEM.

- **Reverse proxy bởi Nginx**: Nginx lắng nghe trên 443, route tất cả request /logstash → backend Beats input của Logstash trên localhost:5044.

- **Xử lý bởi Logstash**

+ Logstash input beats { port => 5045 host => "0.0.0.0" } nhận dữ liệu thô.

+ Chạy qua các filter (grok, json, prune, mutate…) để chuẩn hóa và gắn nhãn.

+ file conf và parser logs:

input {

  beats {

    port => 5045

    ssl => false

    host => "0.0.0.0"

  }

}

filter {

  ################################

  # Log Suricata (eve.json)

  ################################

  if [log][file][path] =~ "suricata/eve.json" {

    # Gắn nhãn nguồn log

    mutate { add\_field => { "log\_source" => "suricata" } }

    # Parse JSON nội dung log

    json { source => "message" }

    # Chỉ giữ sự kiện có event\_type = alert hoặc drop

    if [event\_type] != "alert" and [event\_type] != "drop" {

      drop { }

    }

    # Đưa các trường trong [alert] lên cấp trên

    mutate {

      add\_field => { "signature\_id"     => "%{[alert][signature\_id]}" }

      add\_field => { "alert\_signature"  => "%{[alert][signature]}" }

      add\_field => { "alert\_action"     => "%{[alert][action]}" }

    }

    # Xóa các trường không cần thiết

    mutate { remove\_field => [ "alert", "message" ] }

  }

  ################################

  # Log Nginx - access.log

  ################################

  if [log][file][path] =~ "nginx/access.log" {

    mutate { add\_field => { "log\_source" => "nginx\_access" } }

    # Grok parse theo định dạng access log: client\_ip, method, request, response\_code, bytes, referrer, agent

    grok {

      match => {

        "message" => [

          # 1) Chuẩn HTTP request

          "%{IPORHOST:client\_ip} %{DATA:ident} %{DATA:auth} \[%{HTTPDATE:timestamp}\] \"%{WORD:method} %{DATA:request} HTTP/%{NUMBER:httpversion}\" %{INT:response\_code} %{INT:bytes} \"%{DATA:referrer}\" \"%{DATA:agent}\"",

          # 2) Fallback: bất kỳ chuỗi trong dấu "..."

          "%{IPORHOST:client\_ip} %{DATA:ident} %{DATA:auth} \[%{HTTPDATE:timestamp}\] \"%{DATA:raw\_request}\" %{INT:response\_code} %{INT:bytes} \"%{DATA:referrer}\" \"%{DATA:agent}\""

        ]

       }

    }

    # Nếu chỉ có raw\_request, gán thành method cho dễ xài

    if [raw\_request] and method] {

      mutate {

       add\_field => { "method" => "%{raw\_request}" }

        remove\_field => [ "raw\_request" ]

        }

      }

    # Loại bỏ các trường phụ và message gốc

    mutate { remove\_field => [ "ident", "auth", "message" ] }

  }

  ################################

  # Log Nginx - error.log

  ################################

  else if [log][file][path] =~ "nginx/error.log" {

    mutate { add\_field => { "log\_source" => "nginx\_error" } }

    # Grok parse error log: host, program và thông điệp lỗi

    grok {

      match => { "message" => "%{DATA:host} %{DATA:error\_program}: %{GREEDYDATA:error\_message}" }

    }

    mutate { remove\_field => [ "message" ] }

  }

  ################################

  # Log Hệ thống (auth.log, syslog, messages)

  ################################

  else if [log][file][path] =~ "(auth\\.log|syslog|messages)" {

    mutate { add\_field => { "log\_source" => "system" } }

    # Grok parse Syslog tiêu chuẩn: host, chương trình, thông điệp

    grok {

      match => { "message" => "%{SYSLOGHOST:syslog\_host} %{SYSLOGPROG:sys\_program}: %{GREEDYDATA:sys\_message}" }

    }

    mutate { remove\_field => [ "message" ] }

  }

  ################################

  # Windows Security Logs

  ################################

  else if [log][file][path] =~ "Security" {

    mutate { add\_field => { "log\_source" => "windows\_security" } }

    # Không cần parse thêm; giữ các trường cơ bản như winlog.event\_id, event\_data.SubjectUserName, event.action, message

  }

  ################################

  # Các log khác

  ################################

  else {

    mutate { add\_field => { "log\_source" => "other" } }

    # Không parse thêm; sẽ giữ lại timestamp và message gốc

  }

  mutate {

    remove\_field => ["host", "agent", "ecs", "input", "log", "fileset"]

  }

  ################################

  # Prune filter: chỉ giữ các trường cần thiết

  ################################

  prune {

    whitelist\_names => [

      "@timestamp", "log\_source",

      # Trường Suricata

      "event\_type", "src\_ip", "dest\_ip", "proto", "flow\_id",

      "signature\_id", "alert\_signature", "alert\_action", "community\_id",

      # Trường Nginx access

      "client\_ip", "method", "request", "response\_code", "bytes", "referrer", "agent",

      # Trường Nginx error

      "host", "error\_program", "error\_message",

      # Trường hệ thống

      "syslog\_host", "sys\_program", "sys\_message",

      # Trường Windows

      "host.name", "winlog.event\_id", "winlog.event\_data.SubjectUserName", "event.action",

      # Luôn giữ message để hiển thị (nếu cần)

      "message"

    ]

  }

}

output {

  elasticsearch {

    index => "system\_logs-2"

    hosts => ["https://localhost:9200"]

    user => "wanthinnn"

    password => "Thienlai0941841870@#$"

    ssl => true

    ssl\_certificate\_verification => false

  }

  # Thêm output debug

  stdout { codec => rubydebug }

}

- **Lưu trữ vào Elasticsearch**: Logstash output gửi event đã parse → Elasticsearch trên localhost:9200.

- **Hiển thị trên Kibana**

+ Người dùng truy cập https://siem-dacn.local/kibana → Nginx reverse proxy → Kibana UI.

+ Kibana query Elasticsearch, dựng dashboard và bảng điều khiển từ index system\_logs\*.

# 5. Giả lập tấn công

Mục tiêu: Kiểm tra Suricata phát hiện xâm nhập và khả năng hiển thị logs-parser của Kibana, đồng thời xây dựng các detection rule & alert trong Kibana SIEM, mapping sang MITRE ATT&CK.

## 5.1. ICMP Flood Attack – DdoS [ALERT]

- Trên máy Kali, ta tấn công vào máy Ubuntu:

![A screen shot of a computer

AI-generated content may be incorrect.

- Lúc này trên máy Ubuntu không thể vào internet được:

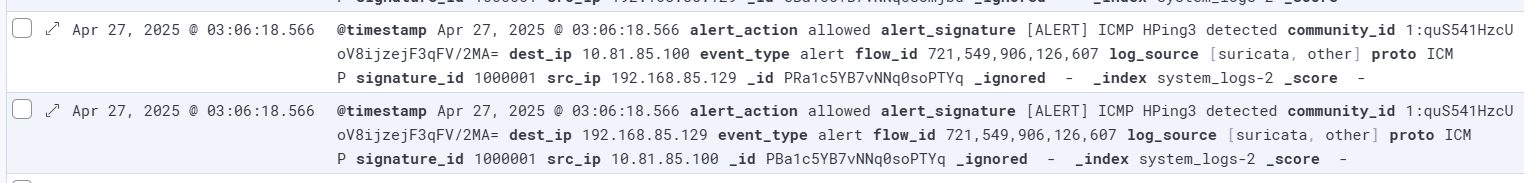
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

- Kiểm tra logs đã được parser:



=> Ta thấy các thông tin hiện lên đầy đủ và chi tiết, dễ nắm bắt được.

## 5.2. UDP Flood Attack – DdoS [DROP]

Tiếp theo, ta tấn công UDP Flood, nhưng mà lần này Suricata sẽ Drop nếu phát hiện, đảm bảo hệ thống an toàn, không bị sập:

A computer screen with white text

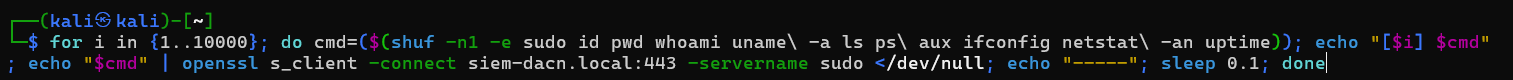
AI-generated content may be incorrect.

Kiểm tra:

A screenshot of a computer

AI-generated content may be incorrect.

## 5.3. Command-and-Control (C2) qua HTTPS/TLS [Bản đơn giản]



- Lặp 10000 lần, mỗi lần chọn ngẫu nhiên một lệnh hệ thống (ví dụ id, whoami, ls, ifconfig…), rồi

- Mở một kết nối TLS tới siem-dacn.local:443 bằng openssl s\_client,

- Đưa tên lệnh đó vào stream TLS (tức gửi “command” đã chọn qua kênh mã hóa),

- Tạm dừng 0.1 giây rồi tiếp tục => Máy chủ SIEM Server bị quá tải

- Lúc này, máy chủ Nginx sẽ gửi liên tục logs về cho Kibana:

A screenshot of a computer

AI-generated content may be incorrect.

# 6. Kibana Detection rules

Dựa vào những logs ta thu về, tiến hành viết Rules cho Kibana để sau này nó detected và alert lên. Dưới đây, nhóm dựa vào signature\_id của Suricata để viết rule và đồng thời mapping với MITTRE ATT&CK:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.