<json format>

- 작성목적 : 네트워크 로그수집 분석 결과를 가지고 리포트 작성할 예정임으로 json파일의 구조를 알아보자

1. 수리카타에 pcap파일을 분석하면 자동생성되는 파일이다 설명

-> (추후) 피캣 검출할 때 어떻게 설정하고 실행했는지 방법작성

2. 기본 내용만 충실 (일단 여기까지!!)

3. 이후에 피캣 검출이 필요할 시 내용 추가될 수 있음

1. 들어가며

Suricata는 네트워크 트래픽 분석 및 침입 탐지 시스템으로, pcap 파일을 분석하여 다양한 포맷으로 결과를 출력할 수 있다.

### **Suricata의 분석 결과 출력**

**1. EVE JSON**: Suricata의 가장 일반적인 출력 포맷 중 하나로, 네트워크 이벤트를 JSON 형식으로 저장한다. 여기에는 HTTP 요청, DNS 쿼리, TLS 정보, 침입 탐지 알림, 플로우 기록 등이 포함될 수 있으며 이 파일은 일반적으로 eve.json이라는 이름으로 저장된다.

**2. PCAP**: Suricata는 원본 패킷을 다시 pcap 파일로 저장할 수 있다.

**3. Unified2**, **Fast log**, **Syslog** 등: 이 외에도 다양한 로그 포맷을 지원하지만, JSON 포맷이 가장 유연하고 다양한 분석 도구와 통합하기 쉬운 형식이다.

따라서 Suricata로 pcap 파일을 분석하면, 설정에 따라 JSON 파일이 생성될 수 있고 이 파일은 네트워크 트래픽에 대한 상세한 정보를 포함하고 있어, 추가 분석이나 데이터 시각화 도구에서 활용할 수 있다.

### **JSON 파일 생성**

1) Suricata 설정 파일(suricata.yaml)에서 EVE(JSON) 로그 출력을 활성화

2) eve.json이라는 파일에 이벤트 저장

3) 세부적인 네트워크 트래픽 정보를 JSON 포맷으로 확인 가능

1. JSON 파일을 Elasticsearch에 적용하는 과정

### **1. Suricata 설정 및 JSON 파일 생성**

Suricata를 설정하여 pcap 파일을 분석하고, 결과를 JSON 형식으로 출력(eve.json 이름으로 저장됨)

### **2. Elasticsearch와 Kibana 준비**

### Suricata에서 생성된 데이터를 저장하고 시각화하는 데 사용

### **3. Filebeat 설정 (옵션)**

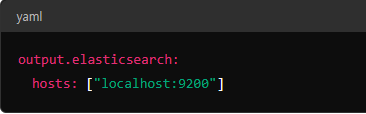
경량 로그 수집기로, JSON 로그 파일을 Elasticsearch로 쉽게 전송할 수 있다.

#### **Filebeat 설정 단계:**

1. **Filebeat 설치**:

공식 Elasticsearch 문서에 따라 설치 진행

1. **Suricata 모듈 활성화**:
   * Filebeat에 Suricata 모듈을 활성화하여, Suricata의 eve.json 파일을 자동으로 처리하고 Elasticsearch로 전송할 수 있다.
   * filebeat modules enable suricata 명령어로 Suricata 모듈을 활성화한다.
2. **Filebeat 설정 파일 수정**:
   * Filebeat 설정 파일(filebeat.yml)을 열고, Suricata 모듈이 eve.json 파일을 찾을 수 있도록 입력 경로를 지정
   * Elasticsearch와의 연결을 설정한다.



1. **Filebeat 시작**:
   * sudo service filebeat start 또는 filebeat -e 명령어를 사용해 Filebeat를 시작
   * Suricata에서 생성된 eve.json 파일을 읽고, 데이터를 Elasticsearch로 전송함

### **4. Kibana에서 데이터 시각화**

데이터가 Elasticsearch에 성공적으로 저장되면, Kibana에서 해당 인덱스를 탐색하고 대시보드를 만들 수 있다.

1. **Kibana에서 인덱스 패턴 생성**:

Kibana에 접속하여 Suricata 데이터에 대한 인덱스 패턴을 생성

2. **데이터 탐색 및 시각화**:

Discover 탭에서 Suricata 이벤트를 탐색하고, 시각화 도구를 사용해 데이터를 분석할 수 있다.

1. JSON 파일을 기반으로 한 정보보안 리포트 항목

JSON 파일(eve.json)에는 다양한 이벤트 유형과 관련된 데이터가 포함되어 있으며, 정보보안 리포트를 작성하기 위해서는 특정 이벤트와 필드를 분석해야 한다.

아래에 각각의 리포트 항목에 해당하는 JSON 파일의 부분과 그 필드들을 설명한다.

### **리포트 작성에 참고할 수 있는 필드**

* **요약(Summary)**: timestamp, alert 필드에서 탐지된 이벤트의 개요를 확인.
* **침입 탐지 이벤트(IDS)**: alert 필드의 signature, severity 및 category.
* **네트워크 트래픽 분석**: src\_ip, dest\_ip, proto, src\_port, dest\_port, flow\_id.
* **HTTP 및 웹 활동**: http 필드에서 웹 요청에 대한 정보를 추출.
* **DNS 활동**: dns 필드에서 DNS 쿼리를 분석.
* **TLS/SSL 활동**: tls 필드에서 인증서 및 암호화 정보 분석.
* **악성 파일 탐지**: fileinfo 필드에서 파일 관련 정보를 확인. 다운로드 된 파일이 악성인지 여부를 분석할 수 있음.

### **1. 요약(Summary)**

* **전체 분석 기간**: 분석이 수행된 시간 범위와 해당 기간 동안 수집된 데이터의 양.

-> JSON 파일의 timestamp 필드를 사용하여 처음과 마지막 이벤트의 시간을 확인

* **탐지된 보안 이벤트 총 수**: 특정 기간 동안 탐지된 전체 보안 이벤트의 수.

-> JSON 파일의 각 이벤트(레코드)를 카운트하여 총 이벤트 수를 계산

* **주요 보안 위협**: 가장 빈번하게 탐지된 위협 유형 요약 (예: 악성 IP, 의심스러운 도메인 등).

-> alert 필드를 분석하여 빈번하게 발생하는 위협 유형을 요약

### **2. 침입 탐지(IDS) 이벤트**

* **탐지된 위협 유형**: 악성 코드, DoS 공격, SQL 인젝션 등 탐지된 위협 유형과 그 발생 횟수.

-> alert 필드의 signature와 signature\_id 값을 사용해 탐지된 위협의 유형을 확인

* **탐지된 서명(Signatures)**: 탐지된 서명의 상세 목록 (예: 특정 서명 ID, 이름, 설명 등).

-> alert 필드의 signature\_id와 signature 필드를 참조하여 탐지된 서명 정보를 수집

* **심각도 레벨별 이벤트 분포**: 탐지된 이벤트를 심각도(critical, high, medium, low)별로 구분하여 분류.

-> alert 필드의 severity 값을 사용해 심각도에 따라 이벤트를 분류

* **공격 벡터**: 탐지된 공격이 어떤 경로를 통해 이루어졌는지 (예: 웹 애플리케이션, 네트워크 프로토콜, 포트 번호 등).

-> alert 필드의 category와 metadata 필드를 분석하여 공격 경로를 식별

### **3. 네트워크 트래픽 분석**

* **소스 및 목적지 IP 주소**: 네트워크에서 가장 빈번하게 사용된 소스 및 목적지 IP 주소.

-> src\_ip와 dest\_ip 필드를 사용해 빈번히 등장하는 IP 주소를 식별

* **프로토콜 사용량**: TCP, UDP, ICMP 등 프로토콜별 트래픽 분포.

-> proto 필드를 통해 트래픽에 사용된 프로토콜(TCP, UDP, ICMP 등)을 분석

* **가장 많이 사용된 포트**: 탐지된 이벤트에서 자주 사용된 포트 번호와 관련 트래픽.

-> src\_port와 dest\_port 필드를 분석하여 빈번히 사용된 포트를 식별

* **데이터 전송량**: 소스 및 목적지 간 전송된 데이터의 양 (예: 바이트, 패킷 수).

-> flow 필드의 bytes\_toserver, bytes\_toclient 값을 사용해 데이터 전송량을 계산

### **4. HTTP 및 웹 활동**

* **탐지된 HTTP 요청**: GET, POST, PUT 등의 요청 유형과 관련된 보안 이벤트.

-> http 필드의 http\_method, url, status, hostname 등을 분석하여 HTTP 요청을 평가

* **탐지된 악성 URL 또는 도메인**: 악성 활동과 관련된 URL 또는 도메인 목록.

-> http 필드의 url과 hostname을 통해 악성 URL이나 도메인을 탐지

* **사용된 브라우저 및 사용자 에이전트**: 탐지된 HTTP 트래픽에서 사용된 브라우저와 사용자 에이전트 정보.

-> http 필드의 user\_agent 값을 분석

### **5. DNS 활동**

* **탐지된 DNS 쿼리**: 수상한 DNS 요청, 의심스러운 도메인 이름 조회 등.

-> dns 필드의 query와 rrname 값을 분석해 의심스러운 DNS 요청을 확인

* **도메인 분포**: 요청된 도메인의 카테고리 (예: 정상, 의심스러움, 악성 등).

-> dns 필드의 rrname을 분석하여 도메인 요청을 카테고리별로 분류

* **DNS 터널링 가능성**: DNS를 이용한 터널링 공격 시도에 대한 분석.

-> dns 필드의 패턴을 분석하여 터널링 활동을 탐지

### **6. TLS/SSL 활동**

* **인증서 정보**: 사용된 인증서의 세부 정보, 만료된 인증서, 의심스러운 인증서 발행자.

-> tls 필드의 issuerdn, subject, fingerprint, notbefore, notafter 등을 분석해 인증서 정보를 수집

* **암호화 프로토콜 및 암호화 강도**: TLS 버전, 사용된 암호화 알고리즘의 유형과 강도.

-> tls 필드의 version과 cipher를 사용해 TLS 버전과 암호화 강도를 분석

* **암호화된 트래픽 분석**: 비정상적인 암호화된 트래픽 발생 여부.

-> tls 필드의 트래픽 패턴을 분석

### **7. 탐지된 악성 파일**

* **악성 파일 식별**: Suricata가 탐지한 악성 파일의 해시 값, 파일 이름, 전송 경로 등.

-> fileinfo 필드의 filename, md5, sha256, magic 등을 사용해 악성 파일을 탐지

* **파일 전송 프로토콜**: FTP, HTTP, SMB 등 어떤 프로토콜을 통해 파일이 전송되었는지.

-> fileinfo 필드의 protocol을 분석하여 파일 전송에 사용된 프로토콜을 확인

### **8. 사용자 활동 모니터링**

* **비정상적인 사용자 활동**: 네트워크 사용 패턴에서 비정상적인 사용자가 탐지되었는지 (예: 자주 사용되지 않는 시간에 로그인 시도 등).

-> http 또는 ssh 필드의 username이나 http\_user\_agent 등을 사용하여 비정상적인 활동을 분석

* **내부 위협 탐지**: 내부 네트워크에서 의심스러운 활동을 한 사용자 목록.

-> 네트워크 내부 IP (src\_ip, dest\_ip)와 관련된 이벤트를 분석

### **9. 경고 및 알림 요약**

* **발생한 경고**: 특정 기간 동안 발생한 경고의 개요와 그 심각도.

-> alert 필드의 이벤트를 기반으로 생성된 경고의 유형과 수를 요약

* **미해결 경고**: 아직 조치되지 않은 경고와 그에 대한 대응 계획.

-> 경고 이벤트 중 아직 대응이 이루어지지 않은 항목을 식별

### **10. 조치 및 권장 사항**

* **발견된 문제에 대한 조치 사항**: 탐지된 위협에 대해 이미 취해진 조치 또는 계획 중인 대응.

-> alert 필드와 그에 따른 후속 조치를 문서화

* **보안 개선 권장 사항**: 시스템 보안을 강화하기 위한 권장 조치 (예: 패치 업데이트, 방화벽 규칙 수정 등).

-> 탐지된 위협과 취약점에 기반한 권장 조치를 정리

### **11. 로그 및 데이터 저장소 관리**

* **로그 보존 정책**: 수집된 로그 데이터의 보존 기간 및 보안 관리 정책.

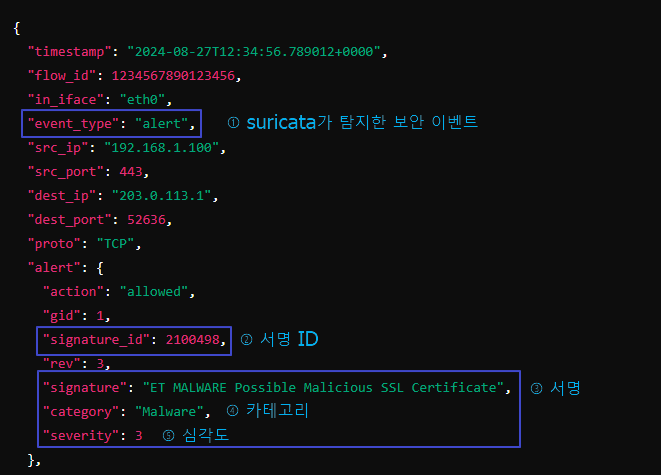
-> timestamp 필드를 통해 로그의 생성 시점과 보존 기간을 평가

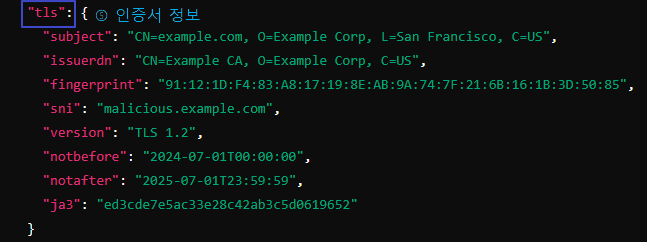
* **데이터 저장소 사용량**: JSON 로그 데이터로 인해 사용된 스토리지의 양과 향후 필요 용량 예측.

-> JSON 파일의 전체 크기와 이벤트 수를 계산하여 로그 데이터의 저장소 사용량을 추정

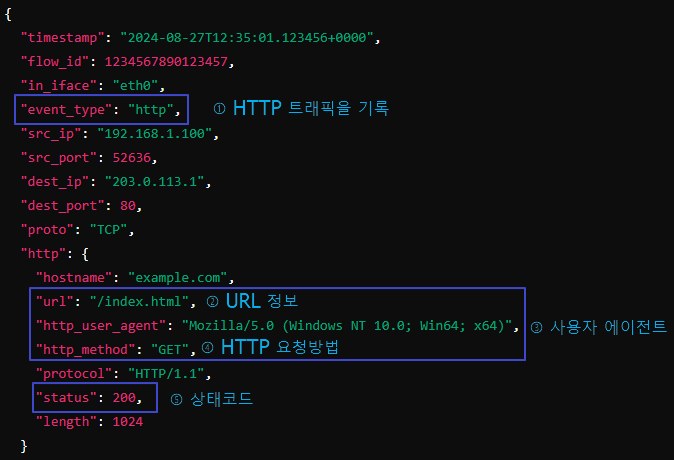
예시)

1. Alert 이벤트

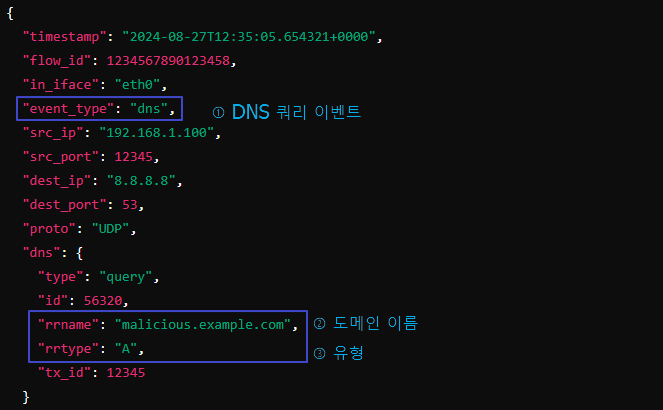




2. HTTP 이벤트



3. DNS 이벤트



4. Fileinfo 이벤트

