

수업 내용 정리 보고서

작성일: 2026년 7월 10일

챕터 구성

1. Network Security Monitoring (NSM) and Security Onion Architecture
2. Data Processing Workflow: From Detection Engines to ELK Stack
3. Network Routing (RIP) and Security System Fundamentals

Network Security Monitoring (NSM) and Security Onion Architecture

네트워크 보안은 단순히 외부 침입을 막는 방화벽(Firewall)만으로 완성되지 않습니다. 방화벽을 통과하거나 내부에서 발생하는 비정상적인 움직임을 포착하기 위해서는 네트워크 전체를 조망하고 기록하는 체계가 필요합니다. 본 장에서는 네트워크 보안 모니터링의 핵심 개념인 **NSM(Network Security Monitoring)**과 이를 구현하기 위한 대표적인 오픈소스 플랫폼인 **Security Onion**의 아키텍처에 대해 심도 있게 학습합니다.

1. NSM(Network Security Monitoring)의 이해

NSM(네트워크 보안 모니터링)이란 네트워크에서 발생하는 모든 트래픽을 지속적으로 수집, 분석하여 침해 사고를 탐지하고 대응하는 전반적인 체계나 솔루션을 의미합니다.

- 핵심 역할:** 단순히 "공격이 들어왔다"는 경고를 올리는 것에 그치지 않고, 어떤 트래픽이 어떤 경로로 이동했는지, 어떤 데이터가 유출되었는지에 대한 가시성을 제공합니다.
- 주요 프로세스:** 트래픽 수집(Collection) → 분석 및 탐지(Analysis/Detection) → 경고 생성(Alerting) → 사고 대응(Response).

2. Security Onion 플랫폼 아키텍처

Security Onion은 위에서 설명한 NSM 기능을 수행하기 위해 필요한 다양한 도구들을 하나로 묶어 놓은 **리눅스 기반의 보안 전문 OS(플랫폼)**입니다. 사용자가 개별 도구들의 의존성이나 연계 문제를 직접 해결할 필요 없이, 설치 즉시 통합된 관제 환경을 구축할 수 있도록 최적화되어 있습니다.

2.1 인터페이스 분리 (Interface Separation)

Security Onion은 물리적 또는 논리적으로 두 종류의 인터페이스를 분리하여 운영하는 것을 원칙으로 합니다. 이는 보안성을 극대화하기 위한 설계입니다.

인터페이스 종류	주요 역할	보안 특징 및 운영 방식
Management (관리)	관리자 접속 및 장비 제어	관리자가 SSH나 웹 UI(Kibana 등)를 통해 접속하는 통로입니다. 보안이 유지된 안전한 경로를 통해 장비를 설정하고 상태를 확인합니다.
Monitor (모니터링)	네트워크 트래픽 수집	IP 주소가 할당되지 않습니다. 오직 네트워크 패킷을 받아 적기(Sniffing)만 합니다. IP가 없기 때문에 해커가 이 인터페이스의 존재를 알아채거나 직접적으로 공격(Targeting)하는 것이 불가능합니다.

2.2 데이터 수집 및 분석 파이프라인

Security Onion 내부에서는 데이터가 수집되어 분석가에게 전달되기까지 다음과 같은 흐름을 거칩니다. 이 흐름을 이해하는 것이 Security Onion 아키텍처의 핵심입니다.

- [탐지 및 기록] 단계:** Monitor 인터페이스를 통해 들어온 패킷을 엔진이 처리합니다.
 - Suricata / Snort (NIDS):** 침입 탐지 엔진입니다. 미리 정의된 규칙(Rule)을 바탕으로 "이 패턴은 공격이다!"라고 판단되면 즉시 **경고(Alert)**를 생성합니다.
 - Zeek (네트워크 분석 프레임워크):** 네트워크의 모든 행동을 육원칙에 따라 기록하는 **CCTV** 역할을 합니다. 공격 여부와 상관없이 "누가, 언제, 어떤 프로토콜로, 어떤 파일을 주고받았는지"에 대한 메타데이터를 상세히 남깁니다. 침해 사고 발생 시 사후 추적(Forensics)에 필수적입니다.
- [로그 정제 및 이동] 단계:** 엔진들이 뱉어낸 다양한 형태의 로그를 표준화합니다.
 - Logstash / Filebeat:** 흩어져 있는 다양한 로그들을 깔끔하게 다듬고(Parsing), 분석하기 좋은 형태로 변환하여 저장소로 배달하는 **청소부이자 배달원** 역할을 수행합니다.
- [저장 및 검색] 단계:** 방대한 데이터를 안전하게 보관합니다.
 - Elasticsearch:** 수많은 네트워크 로그와 경고 데이터를 빛의 속도로 저장하고 검색할 수 있게 해주는 고성능 **분산 데이터베이스(DB)**입니다.
- [모니터링 및 시각화] 단계:** 분석가가 데이터를 확인합니다.
 - Kibana:** 저장된 데이터를 그래프, 차트, 대시보드 형태로 보여주는 **웹 인터페이스**입니다. 분석가는 복잡한 로그 데이터 대신 직관적인 화면을 보며 관제 업무를 수행합니다.

3. 기술의 진화: Legacy 도구 vs Modern Stack

Security Onion은 기술 발전에 따라 분석 환경을 지속적으로 개선해 왔습니다. 과거에는 개별적인 도구들을 조합하여 사용했으나, 현재는 **ELK 스택**을 중심으로 통합되었습니다.

구분	과거 (Legacy Tools)	현재 (Modern ELK Stack)
로그 전달	Barnyard2: 탐지 엔진의 로그를 DB로 전달하는 징검다리 역할	Logstash / Filebeat: 고도화된 정제 및 배달 기능
이벤트 모니터링	sguil: 투박한 디자인의 실시간 이벤트 콘솔 squert: sguil을 보완한 웹 기반 시각화 도구	Kibana: 강력하고 유연한 통합 대시보드 및 시각화 도구

[참고] IDS vs IPS

- IDS (Intrusion Detection System): 침입을 탐지하고 경고를 올리는 시스템 (Security Onion의 주 역할).
- IPS (Intrusion Prevention System): 탐지를 넘어 실시간으로 해당 트래픽을 차단하는 시스템.

4. 학습 마무리 및 점검 사항

실무/시험 관점의 핵심 포인트

- 인터페이스 분리의 목적: 관리 편의성이 아니라, 모니터링 인터페이스를 "Invisible(비가시적)"하게 만들어 공격 대상에서 제외하기 위함임을 명심해야 합니다.
- Suricata와 Zeek의 차이점: "공격인가?"를 묻는 것은 Suricata(Alert 중심)이고, "무슨 일이 있었나?"를 묻는 것은 Zeek(Metadata/CCTV 중심)입니다.
- 데이터 흐름의 이해: Monitor → Engine(Suricata/ZeeK) → Logstash → Elasticsearch → Kibana로 이어지는 파이프라인 순서를 숙지하십시오.

자가 점검 질문

- Security Onion에서 Monitor 인터페이스에 IP를 할당하지 않는 이유는 무엇인가?
- 침해 사고 발생 후, 공격자가 유출한 파일의 이름을 확인하기 위해 가장 먼저 확인해야 할 도구는 무엇인가? (Hint: CCTV 역할)
- 과거의 sgul과 현재의 Kibana는 기능적으로 어떤 관계에 있는가?
- ELK 스택 중 로그를 검색하고 저장하는 핵심 엔진의 명칭은 무엇인가?

Data Processing Workflow: From Detection Engines to ELK Stack

네트워크 보안 관제의 핵심은 네트워크에서 흐르는 방대한 양의 데이터를 어떻게 수집하고, 분석하여, 의미 있는 정보로 변환하느냐에 달려 있습니다. 본 장에서는 오픈소스 네트워크 보안 모니터링(NSM) 플랫폼인 **Security Onion**을 중심으로, 네트워크 트래픽이 탐지 엔진을 거쳐 최종적으로 분석가의 화면에 시각화되기까지의 전체 데이터 처리 흐름(Workflow)을 학습합니다.

1. 네트워크 보안 모니터링(NSM)과 플랫폼의 구조

데이터 흐름을 이해하기 전, 이를 가능하게 하는 기반 환경인 **NSM(Network Security Monitoring)**의 개념을 정립해야 합니다. NSM은 단순히 공격을 차단하는 것을 넘어, 네트워크 트래픽을 수집, 분석, 경고하여 침해 사고를 탐지하고 대응하는 전반적인 체계를 의미합니다.

Security Onion은 이러한 NSM 역할을 수행하기 위해 설계된 리눅스 기반의 보안 전문 운영체제(OS)입니다. 보안 관제 엔진들이 원활하게 작동할 수 있도록 필요한 의존성(Dependency)과 연계 설정을 미리 처리하여 이미지 형태로 제공되는 플랫폼입니다.

인터페이스 분리: 보안성을 위한 설계 원칙

Security Onion은 물리적 또는 논리적으로 두 가지 인터페이스를 엄격히 분리하여 운영합니다. 이는 관제 장비 자체가 공격의 대상이 되는 것을 방지하기 위한 핵심적인 보안 설계입니다.

- Management(관리) 인터페이스:** 관리자가 SSH(Secure Shell)나 웹 UI(User Interface)를 통해 장비에 접속하고 제어하는 통로입니다. 보안이 유지되는 안전한 네트워크 경로를 통해 접근합니다.
- Monitor(모니터링) 인터페이스:** 실제 네트워크 트래픽(Packet)을 수집하는 통로입니다. 이 인터페이스는 별도의 IP 주소를 할당하지 않고 오직 패킷을 받아 적기만 하는 '수동적(Passive)' 상태를 유지합니다. 이는 해커가 해당 인터페이스의 존재를 인지하거나 직접적인 공격을 시도하는 것을 원천적으로 차단하기 위함입니다.

2. 데이터 수집 및 분석 단계: 탐지 엔진의 역할

Monitor 인터페이스를 통해 들어온 데이터는 두 가지 서로 다른 성격의 엔진을 통해 처리됩니다. 하나는 '공격 여부'를 판단하는 엔진이고, 다른 하나는 '행동 기록'을 남기는 엔진입니다.

구분	도구 (Engine)	주요 역할 (Function)	비유
침입 탐지 (IDS)	Suricata, Snort	정해진 규칙(Signature)에 따라 유해한 패턴을 찾아내고 경고(Alert)를 발생시킴	도둑을 발견하면 올리는 경보음
네트워크 분석 (NSM)	Zeek	네트워크의 모든 통신을 분석하여 육원칙(누가, 언제, 어떤 프로토콜로 등)에 따른 메타데이터를 생성	사건의 모든 과정을 기록하는 CCTV

실무 포인트: 침해 사고가 발생했을 때, Suricata가 "공격이 발생했다"라는 신호를 준다면, 분석가는 Zeek가 남긴 기록을 추적하여 "공격자가 어떤 파일을 유출했는지", "어떤 경로를 통해 들어왔는지"에 대한 상세한 맥락(Context)을 파악하게 됩니다. 따라서 두 엔진의 상호보완적 활용이 필수적입니다.

3. 데이터 처리 파이프라인의 진화: Legacy에서 ELK Stack으로

과거의 Security Onion은 탐지 엔진이 생성한 로그를 저장하고 보여주기 위해 *Barneyard2*, *Sguil*, *Squert*라는 도구들을 사용했습니다. 하지만 데이터의 양이 폭

증함에 따라 더 빠르고 강력한 분석 능력이 요구되었고, 현재는 **ELK 스택**이 그 중심 역할을 수행하고 있습니다.

과거의 방식 (Legacy Workflow)

- **Barnyard2**: 탐지 엔진(Snort 등)이 뽑아내는 로그를 데이터베이스(DB)로 빠르게 전달하는 징검다리 역할을 수행했습니다.
- **Sguil/Squert**: 분석가가 이벤트를 확인하는 콘솔 및 UI였으나, 디자인이 투박하고 대용량 데이터 처리 및 시각화에 한계가 있었습니다.

현대의 방식 (Modern ELK Stack Workflow)

현재는 ELK 스택을 통해 데이터의 수집, 정제, 저장, 시각화가 하나의 유기적인 파이프라인으로 연결됩니다.

1. **Logstash / Filebeat (수집 및 정제)**: Suricata, Zeek 등이 생성한 다양한 형태의 원시 로그(Raw Log)를 수집합니다. 이후 데이터를 분석하기 좋은 형태로 깨끗하게 다듬고(Parsing), 목적지인 Elasticsearch로 안전하게 배달하는 '청소부이자 배달원' 역할을 합니다.
2. **Elasticsearch (저장 및 검색)**: 방대한 양의 네트워크 로그와 경고 데이터를 빛의 속도로 저장하고, 분석가가 원하는 조건으로 즉각 검색할 수 있게 해주는 고성능 데이터베이스 역할을 합니다.
3. **Kibana (시각화)**: 저장된 데이터를 그래프, 차트, 대시보드 형태로 변환하여 보여주는 웹 인터페이스입니다. 분석가는 직관적인 화면을 통해 네트워크의 상태를 한눈에 파악하고 관계 업무를 수행합니다.

4. 통합 데이터 처리 워크플로우 요약

전체 과정을 하나의 흐름도로 정리하면 다음과 같습니다.

```
[Monitor 인터페이스]
↓ (트래픽 수집)
[탐지 및 기록: Suricata/Snort & Zeek]
↓ (로그 생성)
[로그 정제 및 이동: Logstash / Beats]
↓ (데이터 전송)
[저장 및 검색: Elasticsearch]
↓ (데이터 요청)
[모니터링 및 시각화: Kibana]
```

[학습 확인 및 실무 점검 리스트]

- **보안 설계 확인**: 관계 장비의 관리용 IP가 모니터링 인터페이스(Monitor)에 노출되어 있지는 않은가? (공격자의 탐지 방지)
- **엔진 역할 구분**: "공격 알람이 울리지 않았는데 의심스러운 통신이 보인다"면, 어떤 엔진의 기록을 먼저 확인해야 하는가? (정답: Zeek의 메타데이터)
- **파이프라인 장애 판단**: Kibana에서 특정 로그가 보이지 않는다면, 어느 단계(Logstash의 정제 문제, Elasticsearch의 저장 문제, 혹은 엔진의 수집 문제)부터 점검해야 하는가?
- **기술적 차이 이해**: Sguil/Squert와 Kibana의 결정적인 차이점은 무엇인가? (정답: 대용량 데이터 처리 능력 및 시각화의 유연성)

Network Routing (RIP) and Security System Fundamentals

본 장에서는 네트워크의 경로를 결정하는 라우팅 프로토콜인 RIP(Routing Information Protocol)의 동작 원리와, 현대 네트워크 보안의 핵심인 네트워크 보안 모니터링(NSM) 플랫폼 및 침입 탐지/방지 시스템(IDS/IPS)의 기초 개념을 다룹니다.

1. 네트워크 보안 모니터링(NSM) 및 Security Onion

네트워크 보안을 유지하기 위해서는 단순히 방화벽을 세우는 것을 넘어, 네트워크 내부에서 어떤 일이 벌어지고 있는지 실시간으로 파악하는 것이 중요합니다. 이를 수행하는 체계를 **네트워크 보안 모니터링(NSM, Network Security Monitoring)**이라고 합니다.

NSM의 정의와 역할

NSM은 네트워크에서 발생하는 모든 트래픽을 수집, 분석, 경고하여 침해 사고를 탐지하고 대응하는 전반적인 체계나 솔루션을 의미합니다. 단순한 차단을 넘어 "누가, 언제, 어떻게" 공격했는지에 대한 증거를 확보하는 것이 핵심입니다.

Security Onion 플랫폼 아키텍처

Security Onion은 이러한 NSM 기능을 통합하여 제공하는 리눅스 기반의 보안 전문 OS(플랫폼)입니다. 사용자가 개별 도구를 일일이 설치하고 연동할 필요 없이, 보안 분석에 필요한 의존성과 연계 설정을 미리 처리하여 이미지 형태로 제공하는 것이 특징입니다.

Security Onion은 보안성과 관리 효율성을 위해 물리적/논리적으로 **인터페이스(Interface, 랜카드)를 두 가지로 분리하여** 운영합니다.

- **Management(관리) 인터페이스**: 관리자가 SSH(Secure Shell)나 웹 UI를 통해 장비에 접속하고 제어하는 안전한 통로입니다.
- **Monitor(모니터링) 인터페이스**: 오직 네트워크 트래픽(패킷)을 수집하기만 하는 통로입니다. 보안을 위해 이 인터페이스에는 IP 주소를 할당하지 않는 경우가 많습니다. 이렇게 하면 해커가 해당 인터페이스의 존재를 알아채거나 직접적인 공격을 가하는 것이 불가능해져, 공격자 모르게 트래픽을 관찰할 수 있습니다.

핵심 분석 도구: 탐지 엔진 vs 기록 엔진

Security Onion 내부에서는 트래픽을 분석하기 위해 서로 다른 목적을 가진 엔진들이 협업합니다.

구분	도구 명칭	주요 역할 (비유)	상세 설명
탐지 엔진 (IDS)	Snort, Suricata	경보 시스템 (Alert)	기존에 알려진 공격 패턴(Signature)을 기반으로 "이것은 공격이다!"라는 경고를 발생시킵니다.
기록 엔진 (NSM)	Zeek	CCTV (Metadata)	공격 여부와 상관없이 "누가, 언제, 어떤 프로토콜로, 어떤 파일을 주고받았는지" 모든 행동을 육하원칙에 따라 기록합니다. 침해 사고 발생 시 사후 추적에 필수적입니다.

로그 분석 플랫폼의 진화: Legacy에서 ELK Stack으로

과거에는 로그를 관리하기 위해 Barnyard2(전달자), Sguil(모니터링 콘솔), Squert(웹 시각화 UI)와 같은 도구들을 조합하여 사용했습니다. 그러나 현재는 대규모 데이터를 더 빠르고 효율적으로 처리하기 위해 **ELK 스택**으로 대체되었습니다.

- Logstash / Filebeat (수집 및 정제):** Snort, Suricata, Zeek 등이 생성한 다양한 형태의 로그를 수집하여, 분석하기 좋은 형태로 깔끔하게 다듬는 '청소부'이자 '배달원' 역할을 합니다.
- Elasticsearch (저장 및 검색):** 방대한 양의 로그와 경고 데이터를 초고속으로 저장하고, 분석가가 원하는 데이터를 즉시 찾아낼 수 있게 해주는 데이터베이스(DB) 역할을 합니다.
- Kibana (시각화):** 저장된 데이터를 그래프, 차트, 대시보드 형태로 보여주는 웹 인터페이스입니다. 보안 분석가는 이 화면을 통해 전체 네트워크의 상태를 한 눈에 파악합니다.

[데이터 흐름 요약]

Monitor 인터페이스(트래픽 수집) → Suricata/ZeeK(탐지 및 기록) → Logstash/Beats(정제) → Elasticsearch(저장) → Kibana(모니터링)

2. 네트워크 라우팅 프로토콜: RIP (Routing Information Protocol)

네트워크 내의 라우터들이 목적지까지 가장 효율적인 경로를 찾아가기 위해 사용하는 프로토콜 중 하나가 RIP입니다.

RIP의 주요 특징 및 동작 원리

- 거리 벡터(Distance Vector) 방식:** 목적지까지의 거리를 '홉 수(Hop Count)'라는 단위로 측정합니다.
- 최적 경로 결정:** 목적지까지 거치는 라우터의 개수(Hop)가 가장 적은 경로를 최적의 경로로 선택합니다.
- Autosummary (자동 요약):** RIP 설정 시 이 기능이 활성화되면, 네트워크 주소를 자동으로 축약하여 전송함으로써 라우팅 테이블의 크기를 줄이고 효율성을 높입니다.

주요 설정 및 명령어 활용

라우터에서 RIP를 설정하고 기본 경로를 광고할 때 다음과 같은 설정을 사용합니다.

```
router rip
! RIP 프로세스 활성화 및 설정 진행

default-information originate
! 자신이 디폴트 경로(0.0.0.0/0)를 가지고 있음을 모든 RIP 라우터에게 광고함
```

default-information originate 명령어를 사용하면, 해당 라우터는 네트워크의 외부(인터넷 등)로 나가는 통로 역할을 한다는 것을 주변 라우터들에게 알립니다. 설정이 성공하면 라우터의 라우팅 테이블에 "Gateway of last resort is [IP 주소] to network 0.0.0.0"와 같은 메시지가 나타나며, 이를 통해 하위 라우터들이 외부로 나가는 길을 찾을 수 있게 됩니다.

3. 침입 탐지 및 방지 시스템 (IDS/IPS) 기초

네트워크 보안의 핵심 구성 요소인 IDS와 IPS는 목적에 따라 다음과 같이 구분됩니다.

- IDS (Intrusion Detection System, 침입 탐지 시스템):** 네트워크 트래픽을 모니터링하며 비정상적인 패턴이나 공격 징후를 발견하면 관리자에게 **알람(Alert)**을 보냅니다. 즉, "침입이 발생했다"는 사실을 알려주는 역할에 집중합니다.
- IPS (Intrusion Prevention System, 침입 방지 시스템):** IDS의 기능을 포함하면서, 공격이 탐지되는 즉시 해당 트래픽을 **차단(Drop/Block)**하여 공격이 성공하지 못하도록 능동적으로 방어합니다.

학습 요약 및 실무 점검 리스트

핵심 개념 체크리스트

- Security Onion에서 Management와 Monitor 인터페이스를 분리하는 보안상의 이유는 무엇인가?
- Snort/Suricata와 Zeek의 결정적인 차이점(탐지 vs 기록)을 설명할 수 있는가?
- ELK 스택의 각 구성 요소(Elasticsearch, Logstash, Kibana)가 로그 분석 파이프라인에서 수행하는 역할은 무엇인가?
- RIP에서 최적 경로를 결정하는 기준은 무엇인가? (정답: Hop Count)
- `default-information originate` 명령어가 라우팅 테이블에 미치는 영향은 무엇인가?

△ 실무 주의사항 및 오답 포인트

- **인터페이스 설정 오류:** Monitor 인터페이스에 IP를 잘못 할당하여 관리용 인터페이스와 노출될 경우, 공격자의 직접적인 타겟이 될 수 있으므로 주의해야 합니다.
- **RIP의 한계:** RIP는 홉 수(Hop Count)만을 기준으로 경로를 결정하므로, 실제 대역폭(Bandwidth)이 훨씬 빠른 경로라도 홉 수가 많으면 최적 경로에서 제외될 수 있다는 점을 인지해야 합니다.
- **IDS vs IPS 선택:** 네트워크 서비스의 중단이 치명적인 환경에서는 오탐(False Positive)으로 인해 정상 트래픽이 차단될 위험이 있는 IPS보다, 먼저 IDS로 모니터링하며 정확도를 높이는 과정이 필요할 수 있습니다.