

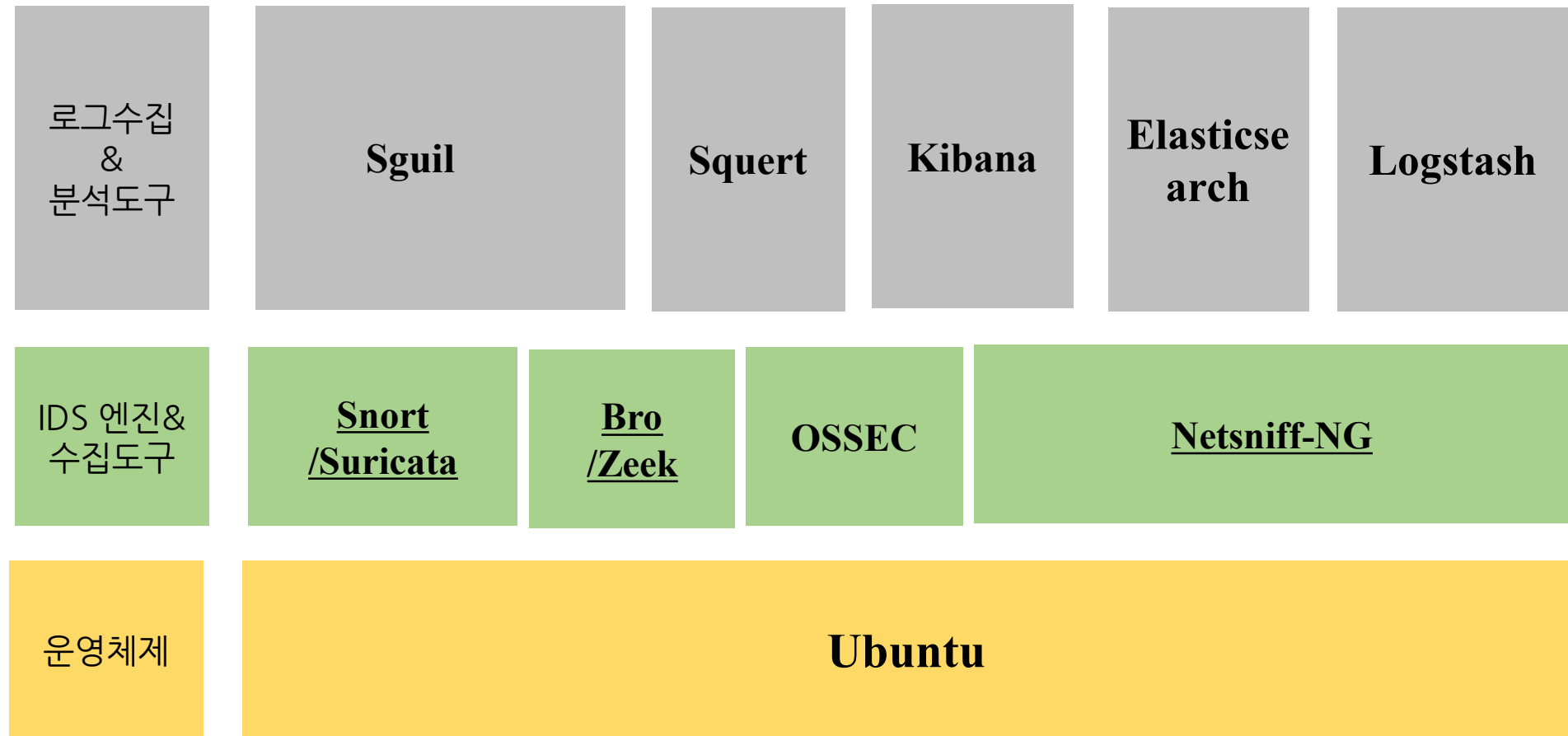
NSM(Network Security Monitoring)

- 네트워크 보안 모니터링을 위한 다양한 도구와 인프라를 제공하여 네트워크 상의 이상 징후를 탐지하고 대응하는 데 도움을 줌
- IDS 기능을 수행할 뿐만 아니라 효율적으로 감시하고 분석하는 역할 수행
- 수집도구, 분석도구, 침입 탐지 시스템이 함께 동작

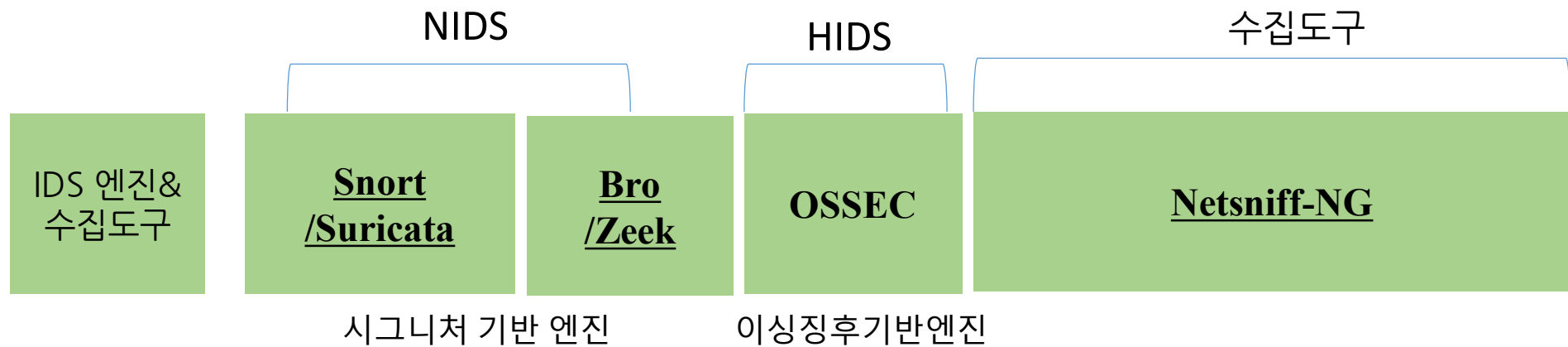
Security Onion

- 더그 벅스(Doug Burks)가 개발
- 네트워크 보안 모니터링 및 이상 징후 탐지를 위한 오픈 소스 플랫폼
 - 오픈 소스로 구현되어 있어 모두 무료
- 리눅스 기반 네트워크 보안 모니터링(NSM)과 침입탐지시스템(IDS) 역할 수행
 - NSM은 IDS 기능을 수행 할 뿐 아니라 효율적으로 감시와 분석 제공
 - NSM은 큰 영역이며 수집도구, 분석도구, IDS이 함께 동작
- 침입탐지 테스트를 위한 교육용 또는 소규모 네트워크 감시로 적합

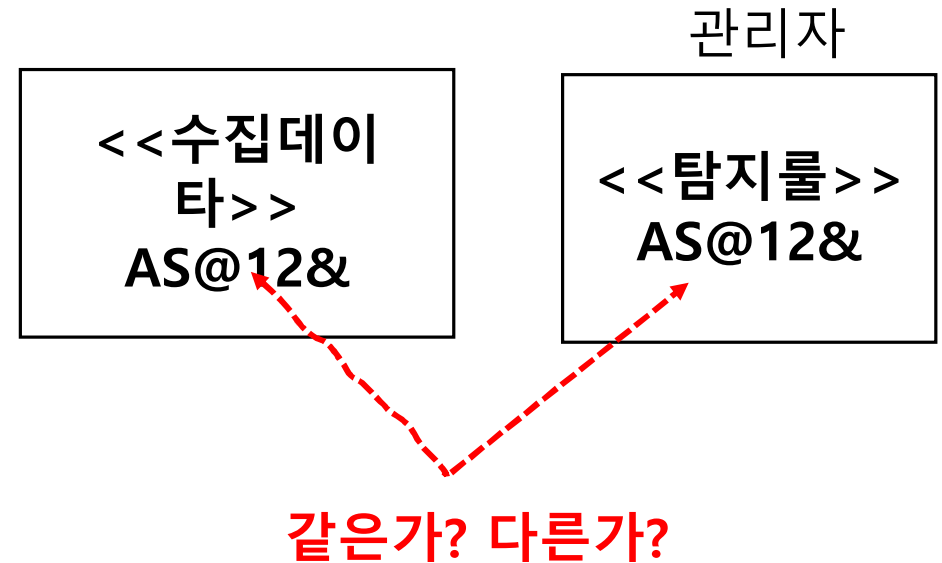
Security Onion Structure



Security Onion Structure



Snort(스노트)



- 1998년 마틴로쉬에 의해 개발
- 오픈소스로 시그니처 기반 NIDS
 - 네트워크 패킷을 수집하여 트래픽을 모니터링
 - 준비된 규칙과 비교하여 침입탐지 및 경로를 발생
- * 시그니처(signature) 기반이란 : 침입탐지를 문자열로 판단하는것
(패킷 데이터에서 악의적인 문자열을 탐지하여 침입여부를 결정)
- 오늘날 침입탐지시스템의 대명사로 사용

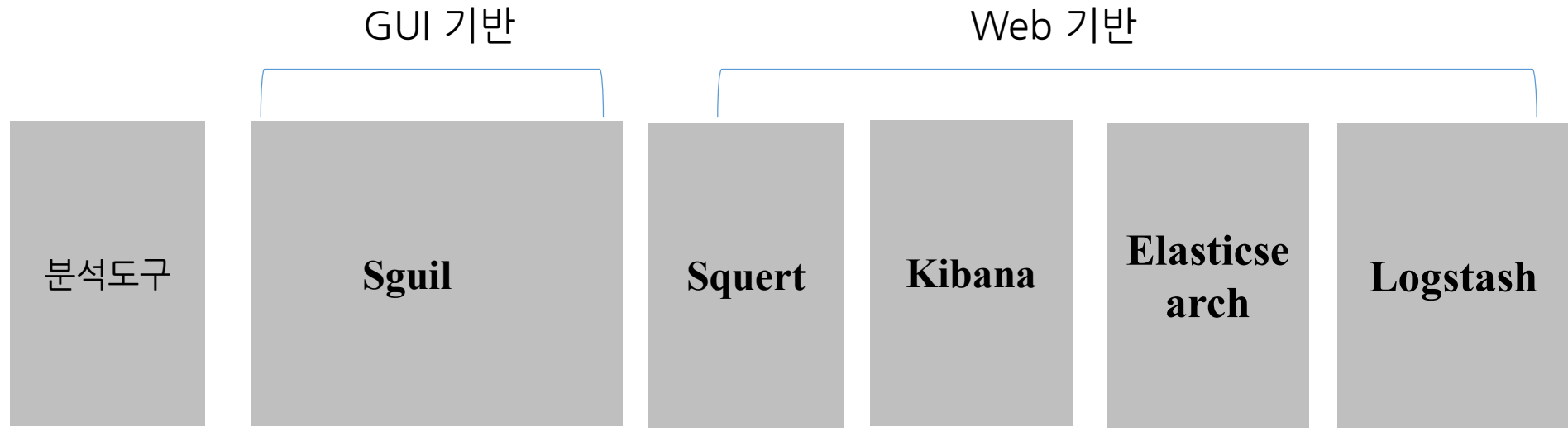
Suricata(수리카타)

- 2010년 OISF 단체에서 오픈 소스 프로젝트로 개발한 NIDS/IPS
- Snort의 단점을 개선하고 장점을 수용
 - 멀티 코어 및 멀티 스레드 지원 : 대용량 트래픽 실시간 처리(성능향상)
 - Snort Rule 완전 호환 및 대부분의 기능 지원
 - 하드웨어 벤더의 개발 지원으로 하드웨어 가속 지원
 - 스크립트 언어(Lua) 지원

Zeek

- 네트워크 침입탐지시스템(NIDS)
 - Bro 또는 프로토콜 분석
 - 네트워크를 모니터링 할 수 있는 오픈 소스 프로그램
- IP헤더와 TCP 헤더를 분석하여 로그 생성
- 응용 프로토콜의 헤더를 분석하여 로그 생성
 - FTP, HTTP, SMTP, X.509 ..

Security Onion Structure



- 로그 수집 및 분석

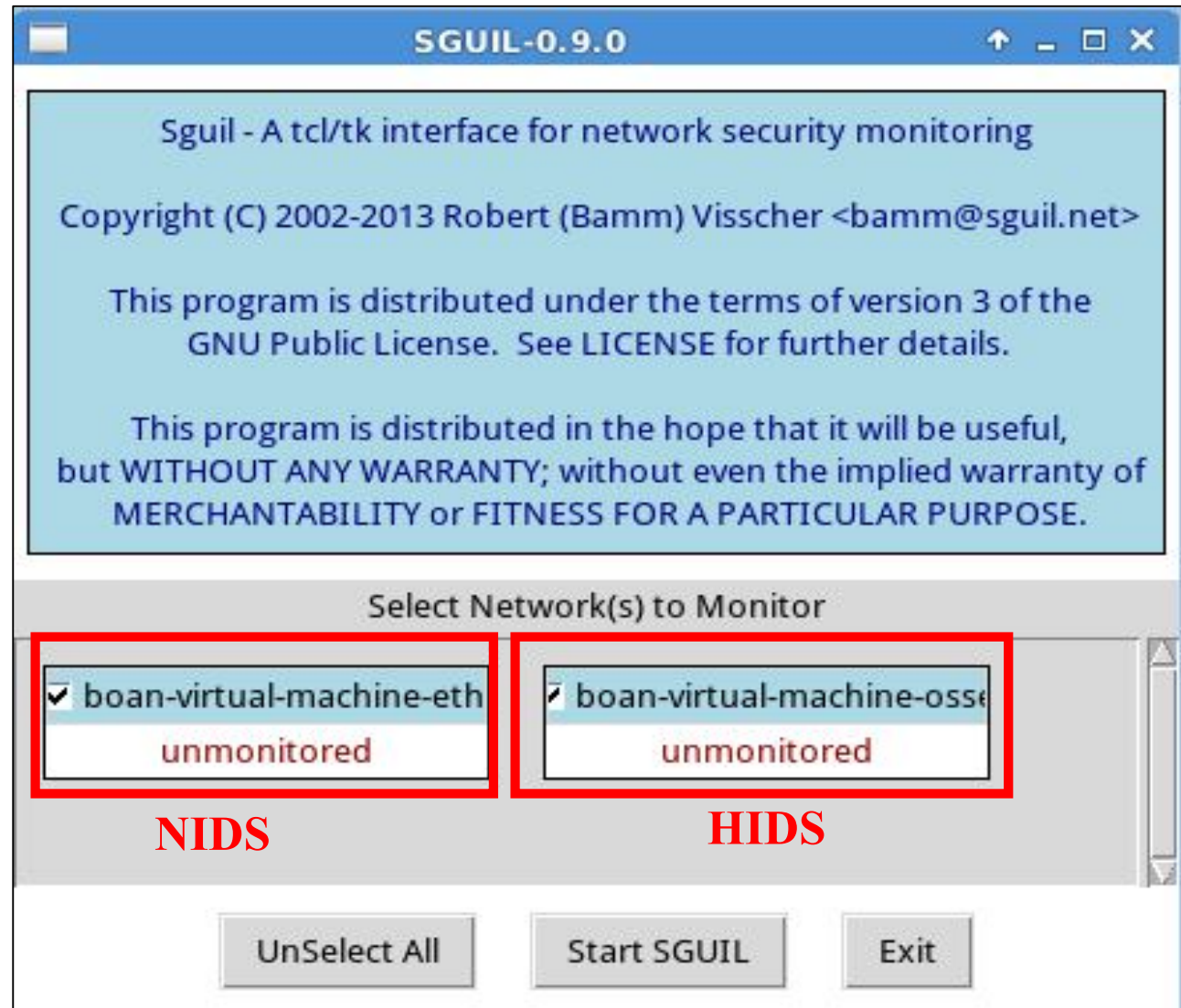
- Elastic Stack (Elasticsearch, Logstash, Kibana)을 통해 로그를 수집하고 분석
- 시각화된 로그 데이터를 통해 보안 이벤트를 식별할 수 있음

Sguil(스구일)

- 네트워크 보안 모니터링(NSM) 도구
- NSM 장비가 수집한 세션 데이터 분석 와 패킷 데이터 또는 탐지한 경고 데이터에 접근과 출력을 제공하는 직관적인 GUI
- 침입 발생 시 경고를 출력하고 이벤트 확인과 검색, 패킷 분석 기회를 제공
- 스크립트는 스구일의 데이터베이스에 저장된 데이터를 시각적으로 표현



Sguil(스구일)



모니터링 대상 지정

- 네트워크 침입 탐지를 경고하는 컴퓨터 명 boanproject-VM-eth0
- 호스트 침입탐지를 경고하는 boanproject-VM-eth0

Sguil 화면 구성

상단메뉴

File Query Reports Sound: Off ServerName: localhost UserName: boan UserID: 2

2017-09-11 20:43:10 GMT

RealTime Events Escalated Events

메인창

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	38	boan-virt...	3.26326	2017-09-11 12:32:00	192.168.10.10		192.168.10.20		1	ICMP Ping TEST
RT	12087	boan-virt...	3.26438	2017-09-11 13:04:47	192.168.10.10	38734	222.97.86.10	80	6	HTTP Packet
RT	5	boan-virt...	3.38527	2017-09-11 14:41:45	192.168.10.10	43405	192.168.10.2	3306	6	ET POLICY Suspicious in...
RT	3	boan-virt...	3.38537	2017-09-11 14:41:45	192.168.10.10	43406	192.168.10.20	5900	6	ET SCAN Potential VNC S...
RT	17	boan-virt...	3.38541	2017-09-11 14:41:45	192.168.10.10	43406	192.168.10.20	22	6	ET SCAN Potential SSH S...
RT	1	boan-virt...	3.38542	2017-09-11 14:41:45	192.168.10.10	43406	192.168.10.20	22	6	ET SCAN Potential SSH S...
RT	3	boan-virt...	3.38551	2017-09-11 14:41:50	192.168.10.10	43405	192.168.10.2	5815	6	ET SCAN Potential VNC S...
RT	5	boan-virt...	3.38552	2017-09-11 14:41:57	192.168.10.10	43405	192.168.10.20	5432	6	ET POLICY Suspicious in...
RT	10	boan-virt...	3.38566	2017-09-11 14:42:52	192.168.10.10	43405	192.168.10.200	1433	6	ET POLICY Suspicious in...
RT	9	boan-virt...	3.38572	2017-09-11 14:42:59	192.168.10.10	43405	192.168.10.200	1521	6	ET POLICY Suspicious in...

IP Resolution Agent Status Snort Statistics System Ms

☐ Reverse DNS ☒ Enable External DNS

Src IP:

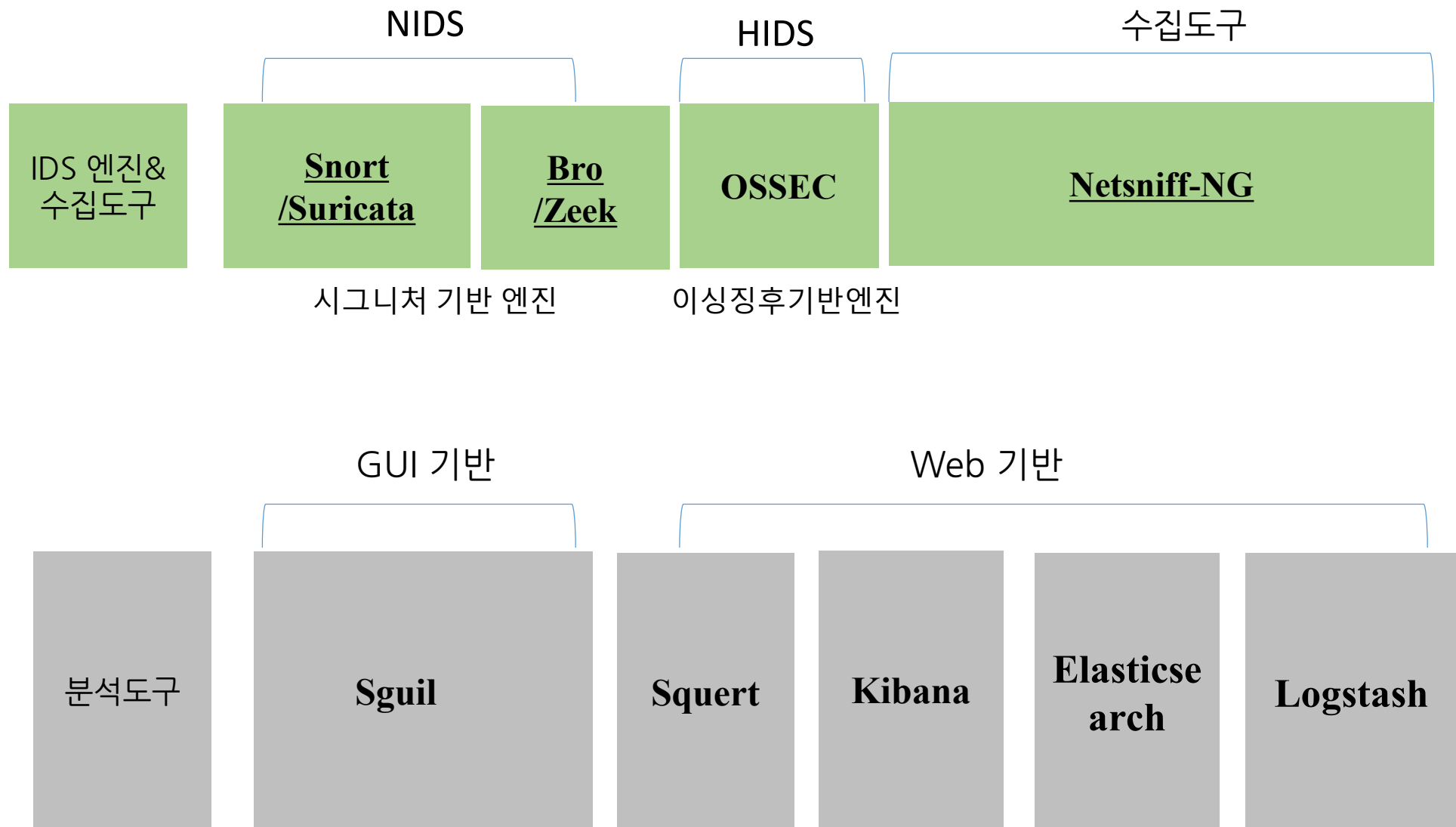
Src Name:

☐ Show Packet Data ☐ Show Rule

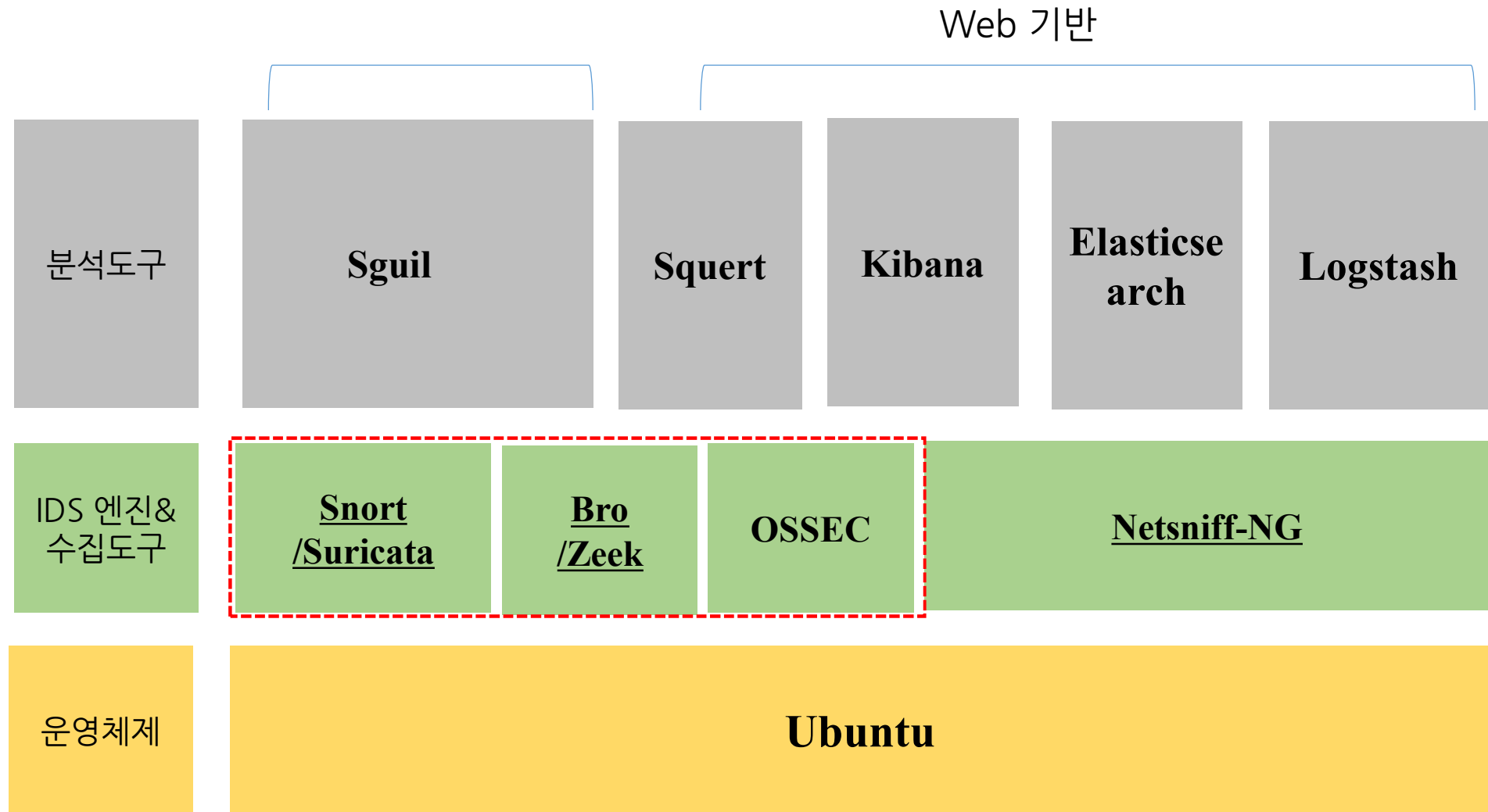
IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	hkSu

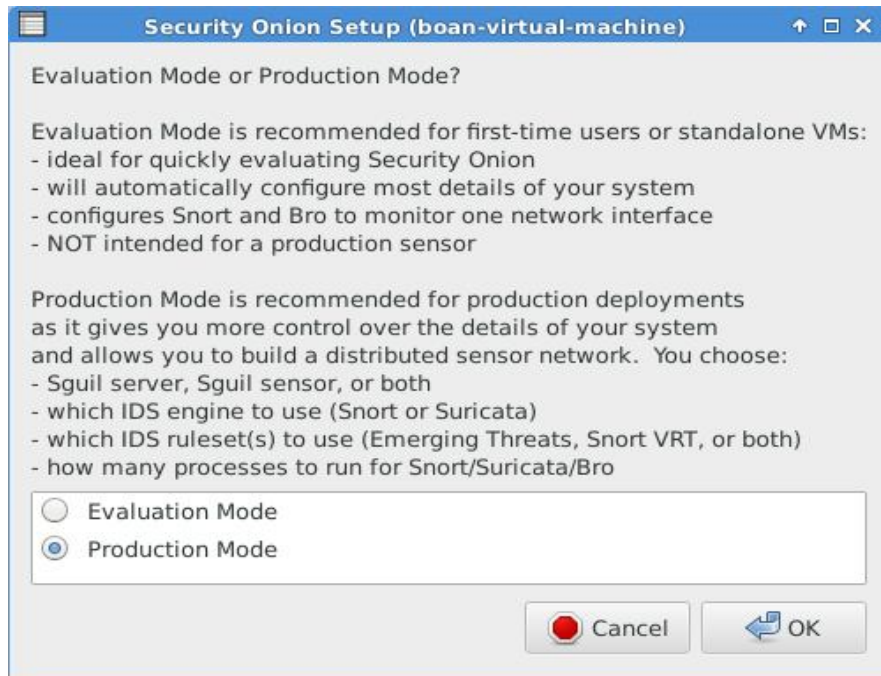
서브창

Security Onion Structure

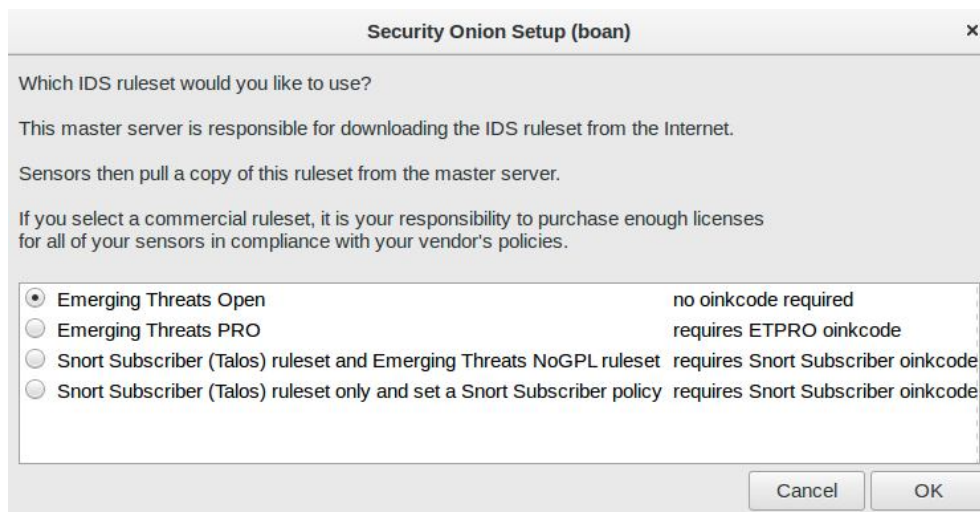


Security Onion Structure

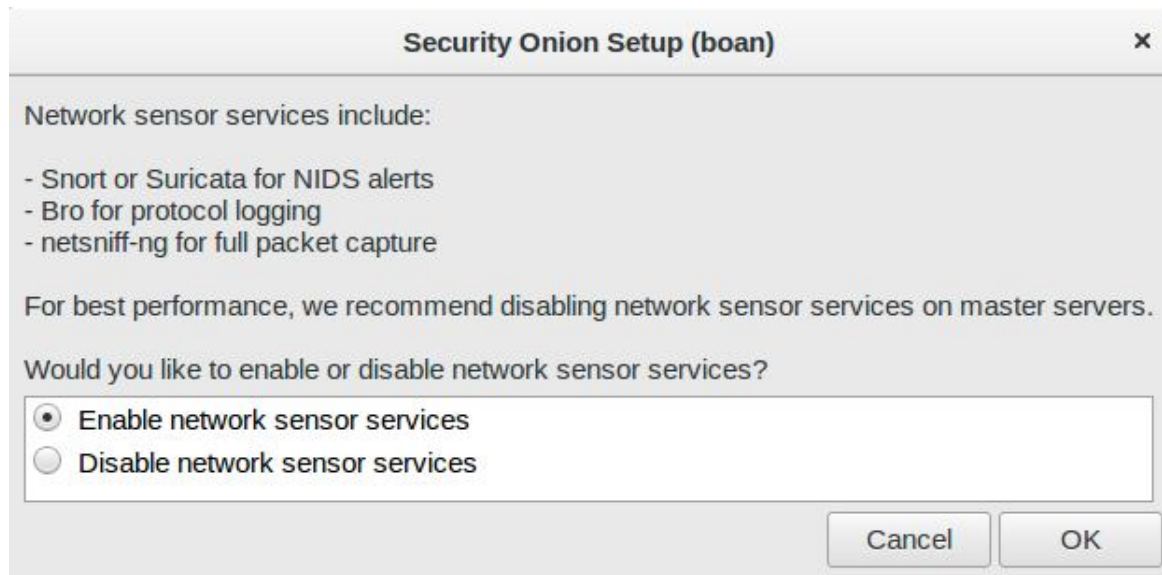




Evaluation	사용자 시스템에 맞게 자동으로 설정 처음 설정 시 권고
production	시스템을 조정 하면서 NSM 설치

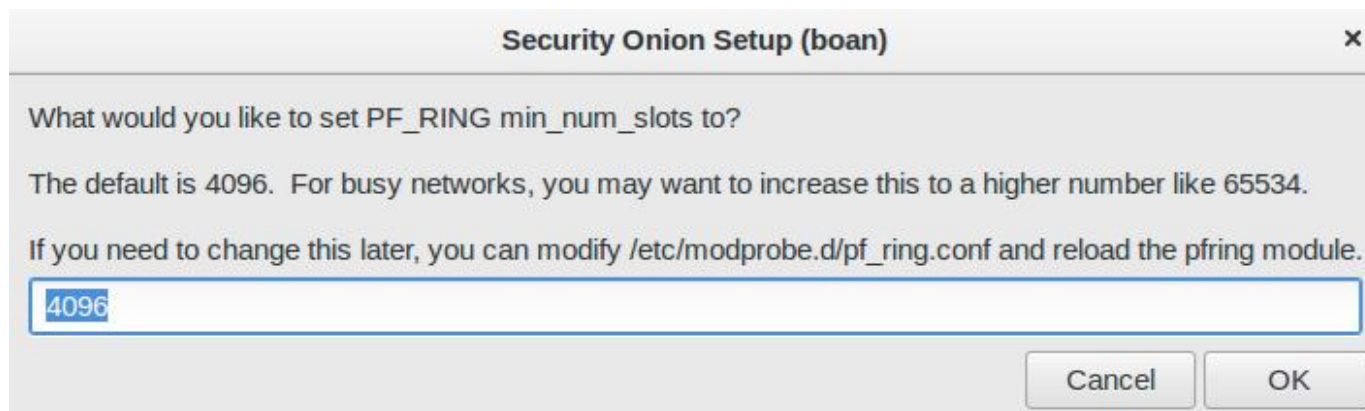


IDS 룰셋 선택: [Emerging Threats GPL] 선택



server


- 분산시스템 구축 시 사용
- 이벤트 분석용으로 사용시 선택



PF_RING

- 수신받은 패킷을 사용자 영역으로 전달 할 수 있는 패킷 캡처를 위한 소켓 트래픽이 많으면 65534와 같은 큰값지정
- PF_RING 의 슬롯 수 지정 [4096] 입력

Security Onion Setup (boan) ✕

 By default, the master server stores logs in its own local Elasticsearch database via a local Logstash instance.

If you want to forward logs from multiple nodes to this master server, then you may overwhelm those single instances of Logstash and Elasticsearch. You may want to consider load balancing these forwarded logs to additional storage nodes.

Would you like to store logs locally on boan?

Security Onion Setup (boan) ✕

How much disk space (in GigaBytes) should be allocated for Elasticsearch to store logs?

Please enter an integer greater than 0.

Please make sure that the value you set here is less than the size of your disk!

If you need to adjust this later, you can modify LOG_SIZE_LIMIT in /etc/nsm/securityonion.conf.

Log_Size_Limit

- Elasticsearch가 로그를 저장하기 위해 할당해야 하는 디스크 공간의 크기 지정 (Gigabyte 단위)

Snort Rule

Snort

- 시그니처 기반 네트워크 침입탐지 시스템 (NIDS)
- 네트워크 패킷을 수집하여 트래픽을 모니터링하고 준비된 규칙과 비교하여 침입 탐지 및 경고를 발생
- 시그니처 기반이란 침입탐지를 문자열로 판단한다는 의미
 - 악의적인 문자열을 탐색하여 침입여부를 결정



- 스니퍼 : 네트워크 패킷을 수집
- 패킷 디코더 : 수집된 패킷은 디코더로 전처리기와 탐지 엔진이 파싱할 수 있도록 정규화
- 전처리기 : 특정 행위가 발결된 패키을 탐색 엔진에 전송
- 탐색엔진 : 해당 패킷이 스노트 규칙에 매칭 되는 지 확인
- 경고/로그 : 스노트 규칙에 매칭 된다면 콘솔 창이라 분석 도구에 경고를 출력하고 기록

Snort Rule

- NIDS나 IPS는 Pattern으로 정의된 rule을 기반으로 탐지한다.
 - Packet의 Payload를 검사하는 방식으로 공격을 탐지한다.
 - 기본적으로 signature(혹은 pattern)을 비교 혹은 검색하는 방식으로 탐지한다.
 - 정규표현식(Regular expression)으로 탐지 규칙을 규정할 수 있다.
 - DDoS 공격은 단위 시간 동안의 '발생량'을 기반으로 탐지한다.

Snort 시그니처 기반의 IDS 의 Detection Rule 기본 구조

```
alert tcp 192.168.10.30/32 40090 -> 192.168.10.20/32 23 (msg: "Hacker Detection";  
content:"hacker";  
nocase;  
sid:3000001;)
```

← 규칙헤더(Rule Header) | 규칙옵션(Rule Option) →

Snort Rule Header

		③트래픽 흐름 방향				
① Action	② Protocol	Src IP	Src Port	Direction	Dst IP	Dst Port
alert	TCP	any	any	→	any	80

❶ Snort Rule Header – Action

Option	Role
alert	경고를 발생한다.
log	패킷을 로그로 저장한다.
pass	패킷을 무시한다.
active	경고를 발생시킨 다음 다른 동적 규칙을 활성화 한다.
dynamic	Active 옵션으로 활성화 된다.

*In-line 모드로 IDS가 배치된 경우

Action	Description
drop	패킷을 차단 한 후 로그로 저장한다.
reject	TCP의 경우, 차단 및 로그 저장 후 세션을 리셋(RST 전송)한다. UDP의 경우 차단 및 로그 저장 후 ICMP port unreachable 메시지를 전송한다.
sdrop	패킷을 차단하지만 로그는 남기지 않는다.

② Snort Rule Header – Protocol

Option	Role
tcp	TCP 프로토콜에 적용
udp	UDP 프로토콜에 적용
icmp	ICMP 프로토콜에 적용
ip	IP 프로토콜에 적용

③ Snort Rule Header – IP, Port

Option	Role	
IP	any	모든 IP 주소
	1.1.1.1	특정 IP 주소
	[1.1.1.1, 2.2.2.2]	여러 IP 주소
	[1.1.1.1/24]	특정 IP 주소
PORT	Any	모든 포트 번호
	80	특정 포트 번호
	1:1024	1~1024 번 포트 범위
	80:	80 번 이상 범위
	:1024	1024번 이하 범위
	!80	80번을 뺀 나머지
→	단방향	
<>	양방향	

④ 규칙 옵션

- 규칙 헤더에 해당하는 패킷 중 특정 패턴(문자열)을 정의해 놓은 영역
- 옵션 종류
 - 일반옵션
 - 흐름 옵션
 - 페이로드
 - HTTP 관련 옵션 등
- 옵션들은 ';' (세미콜론)으로 구분

일반 옵션

Action	Protocol	Src IP	Src Port	Direction	Dst IP	Dst Port
alert	TCP	any	any	→	any	80

msg: "TestAttack";

sid:12345;

classtype:attempted-admin;

rev:1;

)

④ 일반 옵션

일반 옵션

- 규칙에 대한 정보를 제공하는 옵션
- 검색하는 동안 어떤 영향도 미치지 않음

msg	<ul style="list-style-type: none">• 규칙이 탐지될 경우 출력되는 메시지• 공격유형과 정보를 기록
sid	<ul style="list-style-type: none">• 규칙 식별자로 모든 규칙은 반드시 식별 번호를 가짐• 예약된 식별자 : 0~2,999,999• Local.rules에는 3,000,000이상부터 사용
rev	<ul style="list-style-type: none">• 규칙의 수정 버전을 나타냄• 규칙이 수정 시 1 씩 증가
classtype	<ul style="list-style-type: none">• 규칙을 분류하는 옵션• 클래스 명은 classfication.config 파일에 정의
priority	<ul style="list-style-type: none">• 규칙의 우선순위 지정• 1 ~10까지의 수 사용, 숫자가 작을수록 높은 우선순위를 가짐

Payload 옵션

- 악성 패킷을 탐지하는 옵션

content	매칭할 문자열 지정
pcre	문자열로 표현하기 어려운 것들을 정규 표현식을 이용하여 정의 할 경우 사용

- 문자열 지정 **content: “administrator”;**
- 숫자 지정 **content: “|121212|”;**
- 정규표현식 지정 **pcre: “/^select/”;**

“/^select/”; 검색할 문자열 중 가장 앞에 위치한 select 문자가 있는 경우 매치

Payload 옵션

content	매칭할 문자열 지정
nocase	대소문자 구별하지 않고 매칭
offset	매칭할 문자열의 위치 지정
depth	문자열의 범위 지정
distance	Content 옵션값 이후 탐색할 위치 지정
within	Content 옵션값 이후의 탐색할 범위를 지정
pcre	문자열로 표현하기 어려운 것들을 정규 표현식을 이용하여 정의 할 경우 사용

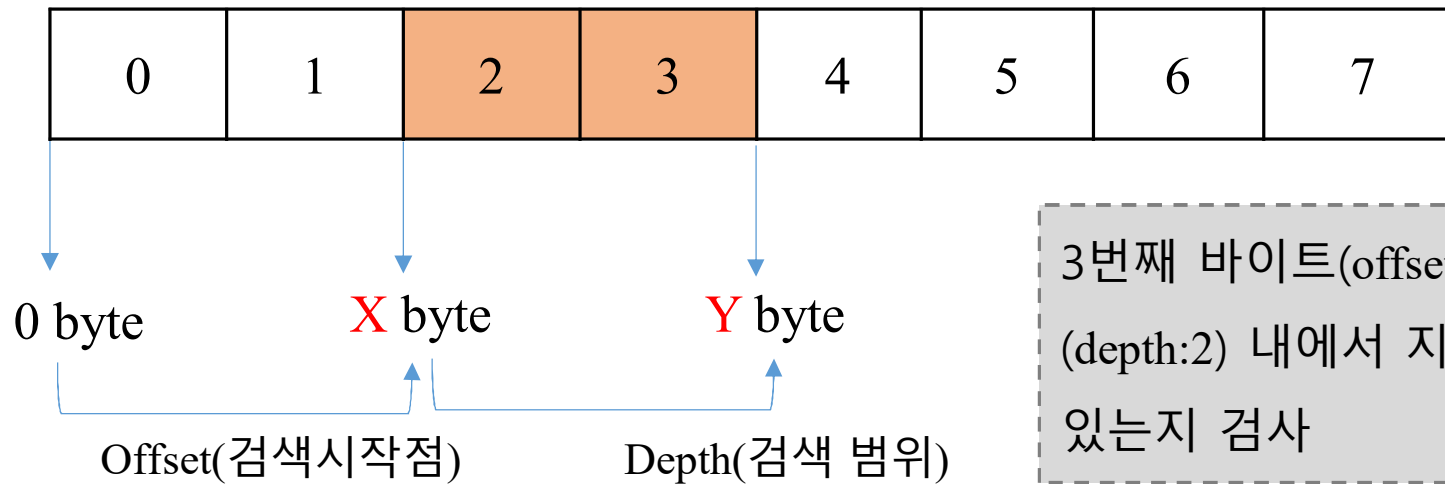
Option	Role	Example
nocase	대/소문자 구별하지 않음	
offset	패킷의 데이터 영역이 시작되는 지점을 기준으로 검사 시작 위치 지정	(content: "a"; offset:1)
depth	Offset으로 시작된 검사의 종료 위치 지정(검사 종료 절대 위치)	(content: "a"; offset:1 ; depth:1)

- **offset**

- content 패턴을 검사 할 시작 위치
- 첫 번째 바이트 위치가 0부터 시작

- **depth**

- offset부터 몇 바이트까지 검사할 것인지 지정



3번째 바이트(offset:2)부터 2바이트 범위
(depth:2) 내에서 지정된 문자열 패턴이
있는지 검사

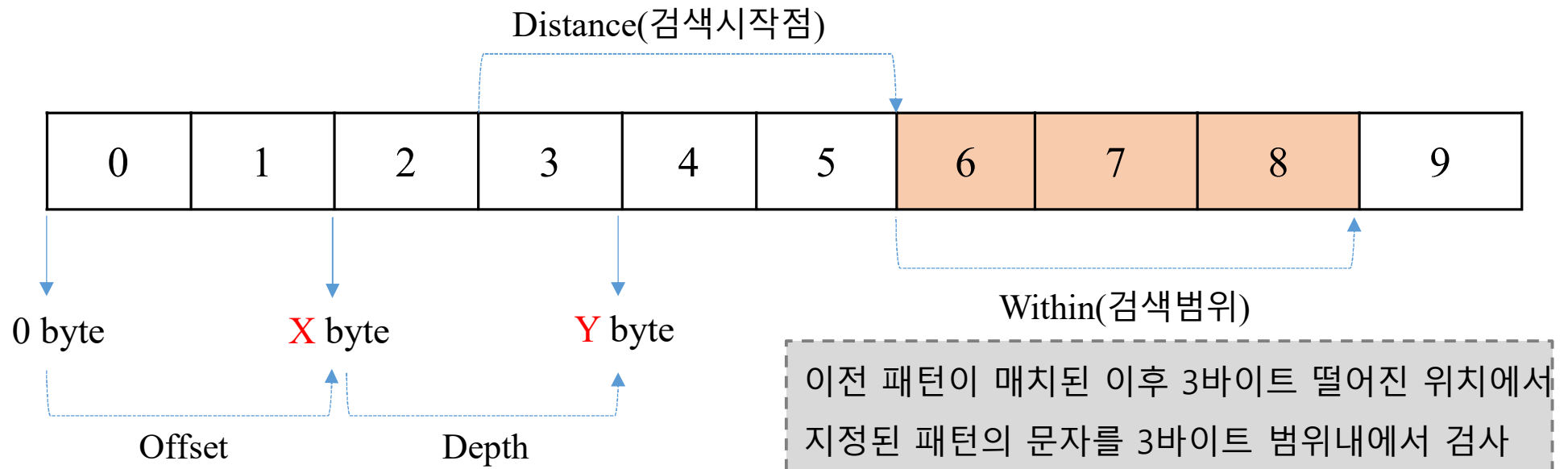
Option	Role	Example
distance	이전 패턴 검사가 종료된 시점을 기준으로 검사 시작 위치 지정	(content:“a”; content:“b”; distance:1;)
within	Distance로 시작된 검사의 종료 위치 지정(검사 종료 상대 위치)	(content:“a”; content:“b”; distance:1; within:1;)
rawbytes	인코딩 된 문자열 패턴의 디코딩 전처리기와 관계없이 Hexa 코드로 검사	(content: 3B 20 ; rawbytes;)

- distance

- 이전 content 패턴이 매치된 경우, 매치된 바이트로부터 몇 바이트 떨어진 위치에서 다음 content를 검사 할 것인지 지정

- within

- distance부터 몇 바이트 범위 내에서 지정된 패턴을 검사할 것인지 지정



Option	Role	Example
http_client_body	패턴 검사 범위를 웹요청(POST) 메시지 본문(body)으로 제한	(content: “a”; http_client_body;)
http_cookie	검사 범위를 cookie로 제한	(content: “a”; http_cookie;)
http_header	검사 범위를 HTTP 헤더 영역으로 제한	(content: “Host”; http_header;)
http_method	검사 범위를 웹 요청 메소드로 제한	(content: “GET”; http_method;)
http_uri	검사 범위를 웹 요청 URI로 제한	(content: “a”; http_uri;)

Option	Role	Example
http_state_code	검사 범위를 웹 응답 코드 번호 영역으로 제한	(content: “200”; http_state_code;)
http_stat_msg	검사범위를 웹 응답 코드 메시지 영역으로 제한	(content: “OK”; http_state_msg;)
fast_pattern	<p>Contents 옵션이 두 개 이상 사용될 때 검색 우선 순위를 조정</p> <p>fast_pattern:only 중복 검사 방지</p> <p>fast_pattern:offset값 우선 검사할 문자열의 검색위치</p> <p>fast_pattern:length값 우선 검사할 문자열의 검색 범위</p>	<p>(content: “aa”; content: “b”; fast_pattern;)</p> <p>aa에서 b 문자열을 먼저 검사</p>

Option	Role	Example
i	Content의 nocase와 동일	(content: “a”; pcre: “/(B C)/i”);)
S	메타문자 '.'과 달리 공백문자까지 포함	(content: “a”; pcre: “/\b./s”);)
m	줄 바꿈 문자를 무시하고 여러 행을 한 행으로 이어진 문자열로 처리	(content: “a”; pcre: “/^bc/m”);)
R	Content의 distance와 동일	
B	Content의 rawbytes와 동일	(content: “a”; pcre: “/\x3B\x20/B”);)

Header Rule Option

Option	Role	Example
fragbits	단편화 여부 검사 M(More fragment) D(Don't fragment), R(Reserved bit)	fragbits:M;
fragoffset	단편화된 패킷의 위치 검사	fragbits:M fragoffset:0;
ttl	ttl값 검사	ttl:=128
tos	TOS값 검사	tos:4
id	IP 헤더 ID 값 검사	id:12345;

Header Rule Option

Option	Role	Example
Ipopts	IP헤더 옵션 값 검사	
dsize	패킷 데이터 영역 길이(byte) 검사	dsize:<1024
flow	TCP stream 전 처리로 패킷 방향 정의	flow:from_client; from_client(to_server), from_server(to_client) established
seq	순서번호	
ack	응답값	
window	TCP 헤더 윈도우값	window:55555 or window:!33333
sameip	출발지 목적지가 동일한 IP인지 조사	

Threshold

로그 발생 타입	로그 발생 기준	로그 발생 예시
threshold:type threshold , count 100, seconds 2;	패킷양	2초내에 패킷 100개 : 로그 1개 2초내에 패킷 200개 : 로그 2개 4초내에 패킷 400개 : 로그 4개
threshold:type limit , count 100, seconds 2;	임계시간	2초내에 패킷 100개 : 로그 1개 2초내에 패킷 200개 : 로그 1개 4초내에 패킷 400개 : 로그 2개
threshold:type both , count 100, seconds 2;	IP	2초내에 패킷 100개 : 로그 1개 2초내에 패킷 200개 : 로그 1개 4초내에 패킷 400개 : 로그 1개

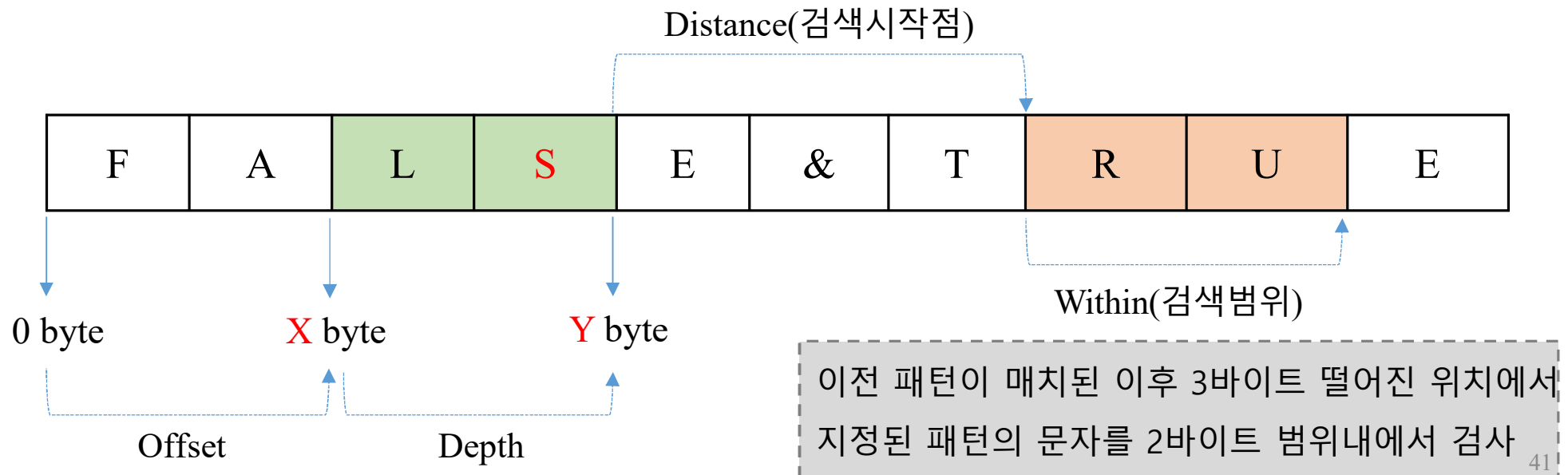
2) Detection Rule Example

① Detection Rule

```
alert tcp $EXTERNAL_NET any → $HOME_NET any
```

```
(msg: "TEST"; content: "S"; offset:2; depth:2; content: "R"; distance:3; within:2;  
sid:1000001;)
```

- 3번째 byte 부터 2byte 범위에 S 패턴이 있는 지 검사, 패턴이 검사된 이 후 3byte 떨어진 위치에서 2byte 범위 내에서 R이라는 문자가 있는지 검색

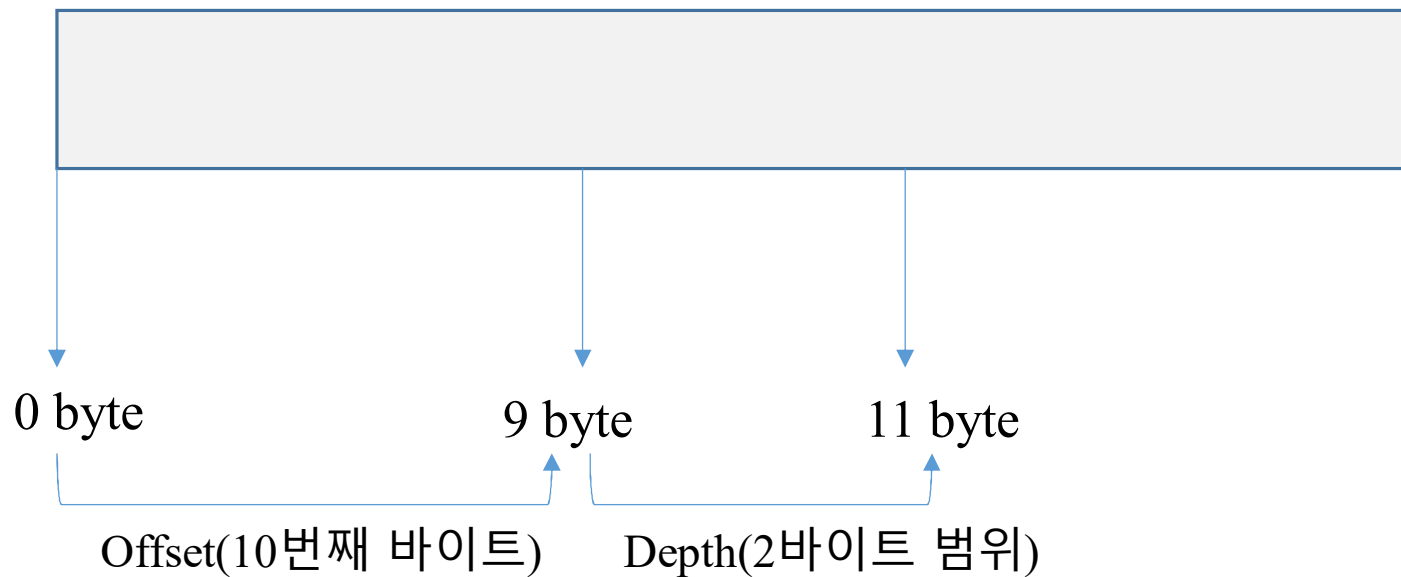


② Detection Rule

```
alert tcp $EXTERNAL_NET any → $HOME_NET any
```

```
(msg: "TEST"; content:|FFFF|; offset:9; depth:2; sid:1000001;)
```

- 10번째 byte 부터 2byte 범위에 FFFF 바이너리 패턴이 있는지 검사



③ Detection Rule

```
alert tcp any any → any 80
```

```
(msg: “ Web Scan Detected”; content: “/administrator”);
```

- 전송되는 패킷의 내용을 검사하여 “/administrator”란 문자열이 포함 된 경우 “Web Scan Detected”란 메시지로 로깅

④ Detection Rule

```
alert tcp any any → any 80 (content: “root”; nocase;)
```

- 목적지 포트가 80인 모든 TCP 패킷에 대하여 대/소문자 구분없이 페이로드에 root문자열이 포함한 경우 alert 발생

⑤ Detection Rule

```
alert tcp any any → any 22 (content: “login”; depth:10;)
```

- 목적지 포트가 22인 모두 TCP 패킷에 대하여 페이로드의 첫번째 byte부터 10byte 범위 내에 소문자 login 문자열이 포함한 경우 alert 발생

⑥ Detection Rule

HTTP Flooding 공격 특징은 대부분 웹서버 공격 트래픽에서 최초 웹페이지에 대해 웹 접속 요청을 폭주시켜 세션자원을 소진검색 탐지 룰

```
alert tcp any any → any any (msg: "Get Flooding"; content:"Get / HTTP1."; nocase; depth:13; threshold:type threshold, track by_dst, count 10, seconds 1; sid:1000999)
```

- 첫 번째 바이트부터 13번째 바이트 범위 내에서 검색
(offset을 명시하지 않으면 첫번째 byte부터 검색)
- Get : http request line, / : 호스트의 default page , HTTP1. : HTTP 버전
- 목적지 IP주소를 기준으로 1초마다 10번째 이벤트마다 alert action을 수행시켜 과도하게 많은 alert event가 발생하는 것을 방지

⑦ Detection Rule

```
drop tcp any any → any any (msg: “ SYN/FIN Drop”; flags:SF;)
```

- 제어 플래그 중 SYN와 FIN이 동시에 설정되어 있는 TCP 패킷 차단한다.
- SYN은 연결 요청, FIN는 연결 종료를 위한 플래그이므로 동시에 설정될 수 없는 비정상 패킷이다. 비정상 패킷은 IDS/IPS의 탐지를 우회하여 공격 또는 스캐닝을 위한 목적으로 사용되므로 이를 탐지 및 차단해야 한다.

⑧ Detection Rule

```
alert tcp any any → any any (pcre: “ /POST.*Content\x2dLength\x3a\x20evilstring/”);)
```

- 목적지 주소 및 포트가 모두 any로 설정으로 모든 패킷을 검사한다.
이것은 장비에 많은 부하를 발생시킨다.
- HTTP 서비스를 제공하는 IP주소 및 PORT 정보를 파악해서 목적지 IP주소와 Port에 대한 검사만 검사률을 적용시켜 장비의 부하를 줄일 수 있다.

⑨ Detection Rule

```
alert tcp any any → any 80
( msg:“XSS Detect”;
  content:“GET”; offset:0; depth:3;
  content:“/login.php?id=%3Cscript%3E”; distance:1; sid:1000500 );
```

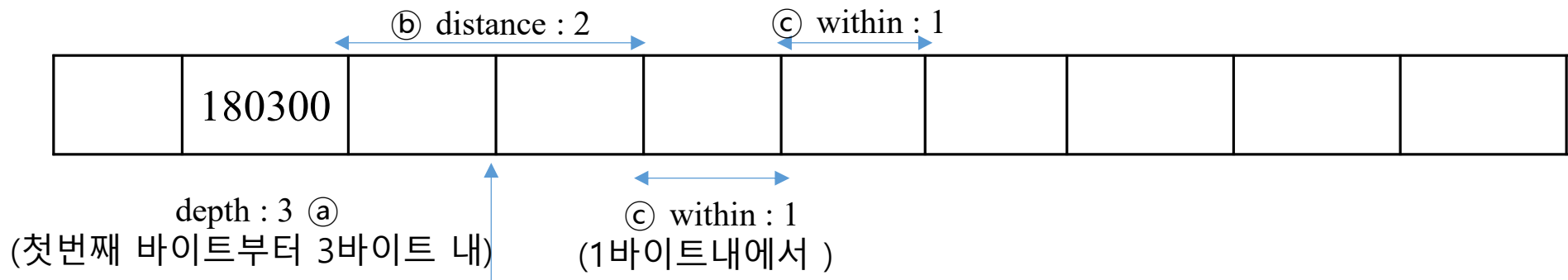
- 첫바이트~ 세번째 바이트 내에서 GET이라는 문자 검색
 - 첫번째 매치된 문자열에서 1바이트 떨어진 곳에서부터 해당 문자 검색
- ➔ Detection 이 안 될 경우 '대소문자' 옵션 첨부

⑩ Detection Rule

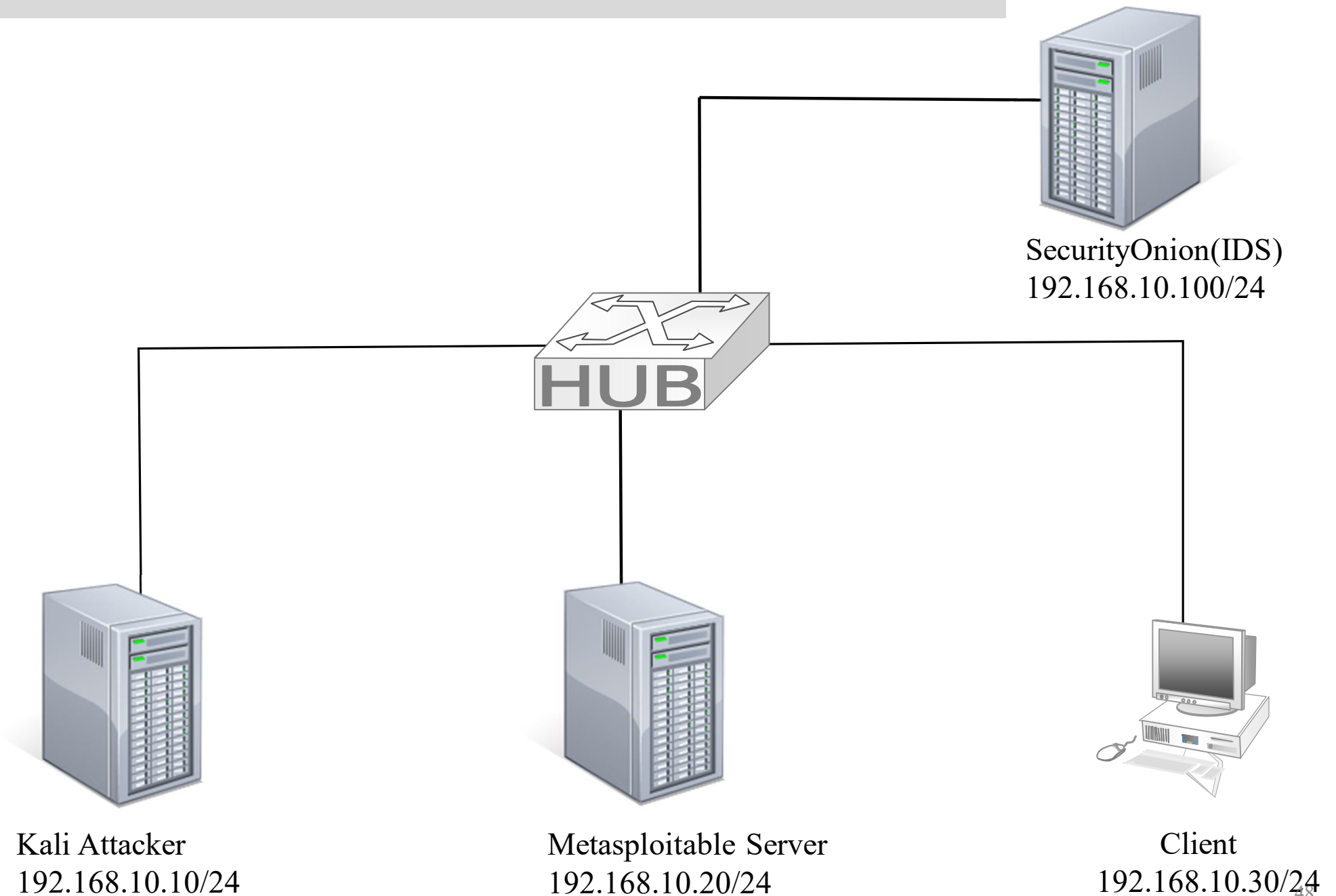
- OpenSSL 라이브러리의 하트비트(HeartBeat) 확장 모듈의 버그로 인해 발생하는 하트블리드(heartbleed) 취약점을 이용한 공격 탐지

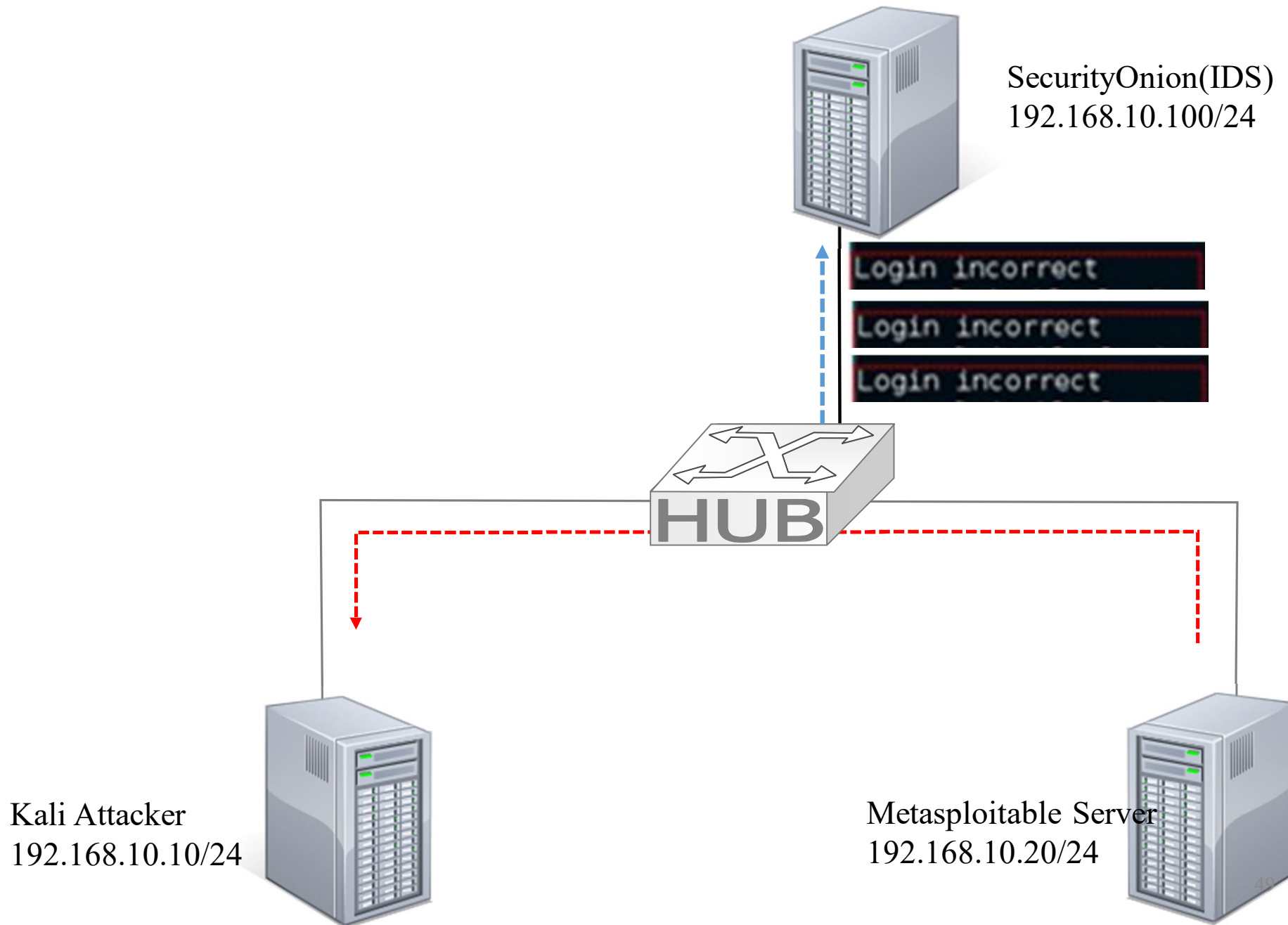
```
alert tcp any any <> any [443,465,563]  
(msg:"SSLv3 Malicious Heartbleed Request V2";  
content: "|18 03 00|"; depth:3;  
content: "|01|"; distance:2, within:1;  
content: "!|00|"; within:1; sid:100300;)
```

- ㉠ 첫 바이트부터 3바이트 범위 내에서 패턴 검사
- ㉡ 첫 번째 content가 매치 된 이후 2바이트 떨어진 위치에서 1바이트 내에서 지정된 패턴 검사
- ㉢ 두 번째 content가 매치된 이후 1바이트 떨어진 위치에서 지정된 패턴 검사



[실습] Password cracking 공격 탐지





① 공격 시그니처 찾기

SecurityOnion(IDS)
192.168.10.200/24



Filter: telnet && ip.src == 192.168.10.20 Expression... Clear Apply 저장

No.	Time	Source	Destination	Protocol	Length	Info
69	41.10283300	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...
72	41.27116100	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...
75	41.43159200	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...
78	41.54224900	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...
81	41.78936700	192.168.10.20	192.168.10.10	TELNET	68	Telnet Data ...
83	41.78974600	192.168.10.20	192.168.10.10	TELNET	76	Telnet Data ...
99	43.12356500	192.168.10.20	192.168.10.10	TELNET	68	Telnet Data ...
103	45.50181000	192.168.10.20	192.168.10.10	TELNET	85	Telnet Data ...
105	45.51516200	192.168.10.20	192.168.10.10	TELNET	88	Telnet Data ...
111	48.79905100	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...
114	48.91509700	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...
118	49.06396700	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...
121	49.11146100	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...
124	49.28038500	192.168.10.20	192.168.10.10	TELNET	67	Telnet Data ...

▶ Frame 103: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface 0

▶ Ethernet II, Src: Vmware_0b:16:25 (00:0c:29:0b:16:25), Dst: Vmware_5c:c3:e0 (00:0c:29:5c:c3:e0)

▶ Internet Protocol Version 4, Src: 192.168.10.20 (192.168.10.20), Dst: 192.168.10.10 (192.168.10.10)

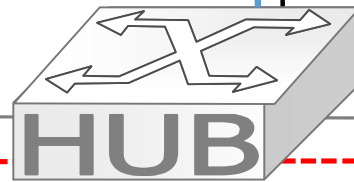
▶ Transmission Control Protocol, Src Port: 23 (23), Dst Port: 36529 (36529), Seq: 696, Ack: 133, Len: 19

Telnet

Data: \r\n

Data: Login incorrect\r\n

패스워드 크래킹의 시그니처 : login incorrect



Kali Attacker
192.168.10.10/24

Metasploitable
Server
192.168.10.20/24



② Detection Rule 생성

/etc/nsm/templates/snort

```
root@boan-virtual-machine:/etc/nsm/templates/snort# pwd
/etc/nsm/templates/snort
root@boan-virtual-machine:/etc/nsm/templates/snort# ls -l
total 292
-rw-r--r-- 1 root root 13941 Apr 12 2016 Makefile
-rw-r--r-- 1 root root 190 Nov 18 2015 Makefile.am
-rw-r--r-- 1 root root 12306 Mar 18 2016 Makefile.in
-rw-r--r-- 1 root root 1281 Aug 20 2007 attribute_table.dtd
-rw-r--r-- 1 root root 3757 Nov 18 2015 classification.config
-rw-r--r-- 1 root root 23058 Jun 10 2014 file_magic.conf
-rw-r--r-- 1 root root 31971 Nov 18 2015 gen-msg.map
-rw-r--r-- 1 root root 687 Nov 18 2015 reference.config
-rw-r--r-- 1 root root 23609 Apr 12 2016 snort.conf
-rw-r--r-- 1 root root 2335 Jul 7 2009 threshold.conf
-rw-r--r-- 1 root root 160606 Jul 13 2011 unicode.map
```

Snort 규칙 파일

/etc/nsm/rules

```
root@boan-virtual-machine:/etc/nsm/rules# pwd
/etc/nsm/rules
root@boan-virtual-machine:/etc/nsm/rules# ls -l
total 17316
-rw-r--r-- 1 sguil sguil      169 Sep 11 12:57 1
-rw-r--r-- 1 sguil sguil    1295 Dec  8  2015 app-layer-events.rules
drwxr-xr-x 2 sguil sguil    4096 Sep 11 14:32 backup
-rw-r--r-- 1 sguil sguil       0 Jun  6  2016 black_list.rules
-rw-r--r-- 1 sguil sguil       0 Sep  9 02:48 bpf.conf
-rw-r--r-- 1 sguil sguil    2855 Sep  9 02:48 classification.config
-rw-r--r-- 1 sguil sguil   15404 Apr  4  2016 decoder-events.rules
-rw-r--r-- 1 sguil sguil    1498 Dec  8  2015 dns-events.rules
-rw-r--r-- 1 sguil sguil 13947294 Sep 11 14:32 downloaded.rules
-rw-r--r-- 1 sguil sguil    3004 Dec  8  2015 files.rules
-rw-r--r-- 1 sguil sguil   31971 Sep  9 02:48 gen-msg.map
-rw-r--r-- 1 sguil sguil    8637 Dec  8  2015 http-events.rules
-rw-r--r-- 1 sguil sguil     377 Sep 11 14:27 local.rules
-rw-r--r-- 1 sguil sguil    1763 Dec  8  2015 modbus-events.rules
-rw-r--r-- 1 sguil sguil    1455 Sep  9 02:48 reference.config
-rw-r--r-- 1 sguil sguil 3647361 Sep 11 14:32 sid-msg.map
-rw-r--r-- 1 sguil sguil    4939 Apr  4  2016 smtp-events.rules
-rw-r--r-- 1 sguil sguil       0 Jun  6  2016 so_rules.rules
-rw-r--r-- 1 sguil sguil   11879 Dec  8  2015 stream-events.rules
-rw-r--r-- 1 sguil sguil    2335 Sep  9 02:48 threshold.conf
-rw-r--r-- 1 sguil sguil    4761 Apr  4  2016 tls-events.rules
-rw-r--r-- 1 sguil sguil       0 Jun  6  2016 white_list.rules
```

① [Kali → Meta]

Connected to 192.168.10.20.

```
Escape character is '^['.
```

$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ -1 & i \end{pmatrix}$

Contact: [msfdev\[at\]metasploit.com](mailto:msfdev[at]metasploit.com)

```
metasploitable login: test ② [ Meta → Kali]
```

③ [Kali → Meta]

④ [Meta \rightarrow Kali]

```
metasploitable login:
```

[SecurityOnion-snort] 탐지 정책 생성

❶ alert tcp 192.168.10.20/32 23 -> 192.168.10.10/32 any

(msg: "Telnet Fail"; content:"login incorrect"; nocase;
sid:3000001;)

❷ alert tcp 192.168.10.20/32 23 -> any any

(msg: "Telnet Fail"; content:"login incorrect"; nocase;
id:3000001;)

④ Snort 규칙 업데이트

rule-update

```
Generating sid-msg.map....
    Done
Writing v1 /etc/nsm/rules/sid-msg.map....
    Done
Writing /var/log/nsm/sid_changes.log....
    Done
Rule Stats...
    New:-----0
    Deleted:---0
    Enabled Rules:----20172
    Dropped Rules:----0
    Disabled Rules:---5491
    Total Rules:-----25663
No IP Blacklist Changes
Done
Please review /var/log/nsm/sid_changes.log for additional details
Fly Piggy Fly!
Restarting Barnyard2.
Restarting: boan-virtual-machine-eth0
    * stopping: barnyard2-1 (spooler, unified2 format)      [ OK ]
    * starting: barnyard2-1 (spooler, unified2 format)      [ OK ]
Restarting IDS Engine.
Restarting: boan-virtual-machine-eth0
    * stopping: snort-1 (alert data)                        [ OK ]
    * starting: snort-1 (alert data)                        [ OK ]
```

<<문법 오류 확인 >>

#cat /var/log/nsm/boanproject-VM-eth0/snortu-x.log 확인

```
# cat /var/log/nsm/boan-virtual-machine-eth0/snortu-1.log
```

<<Snort만 재실행 >>

#nsm --sensor --restart --only-snort-alert

```
# nsm --sensor --restart --only-snort-alert
```

#rule-update

SGUIL-0.9.0



Sguil Host:

Sguil Port:

Username:

Password:

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	3	boan-vir...	311735	2021-11-02 -8:23	192.168.10.20	23	192.168.10.10	36350	6	Telnet Fail

IP Resolution Agent Status Snort Statistics System Msgs Us

☒ Reverse DNS ☒ Enable External DNS

Src IP:

Src Name:

Dst IP:

Dst Name:

Whois Query: ☒ None ☐ Src IP ☐ Dst IP

☒ Show Packet Data ☒ Show Rule

alert tcp 192.168.10.20/32 23 -> 192.168.10.10/32 any (msg:"Telnet Fail"; content:"login incorrect"; nocase; sid:3000001;)

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	192.168.10.20	192.168.10.10	4	5	16	71	19171	2	0	64	23119

TCP	Source Port	Dest Port	R	R	R	C	S	S	S	Y	I	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	23	36350	.	.	.	X	X	324651920	455528992	8	0	181	0	55214

DATA

0D 0A 4C 6F 67 69 6E 20 69 6E 63 6F 72 72 65 63 ..Login incorrec
74 0D 0A t..

Search Packet Payload ☐ Hex ☒ Text ☐ NoCase

[SecurityOnion-snort] 탐지 정책 생성

❸ alert tcp 192.168.10.20/32 23 -> 192.168.10.10/32 any

(msg: "Telnet Attack";

threshold:type both, track by_src, count 3, seconds 20;

content:"login incorrect"; nocase; sid:30000004;)

- Threshold : 동일한 특정 패킷이 관리자 설정한 시간안에 일정 수가 발견이 되면 경고 알림을 출력해주는 것.
- **threshold:type [limit,threshold,both], track [by_src, by_dst], count [몇초], seconds [횟수]**

limit : count 동안 횟 수번째 트래픽까지 탐지

threshold : 횟수 마다 계속 탐지

both : count 동안 횟수 만큼 트래픽이 탐지 될 시1번 만 탐지

by_src : 출발지 패킷만 해당

by_dst : 도착지 패킷만 해당

Threshold

로그 발생 타입	로그 발생 기준	로그 발생 예시
threshold:type threshold , count 100, seconds 2;	패킷양	2초내에 패킷 100개 : 로그 1개 2초내에 패킷 200개 : 로그 2개 4초내에 패킷 400개 : 로그 4개
threshold:type limit , count 100, seconds 2;	임계시간	2초내에 패킷 100개 : 로그 1개 2초내에 패킷 200개 : 로그 1개 4초내에 패킷 400개 : 로그 2개
threshold:type both , count 100, seconds 2;	IP	2초내에 패킷 100개 : 로그 1개 2초내에 패킷 200개 : 로그 1개 4초내에 패킷 400개 : 로그 1개

XSS (Cross-Site Scripting) Attack

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Hello~

Message *

Hello~
<script>alert('This is TEST')</script>

Sign Guestbook

Name: test

Message: This is a test

http

No.	Time	Source	Destination	Protocol	Length
4	0.000687575	192.168.10.10	192.168.10.20	HTTP	72
10	0.017271085	192.168.10.20	192.168.10.10	HTTP	248

File Data: 121 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

Form item: "txtName" = "Hello~"

Key: txtName

Value: Hello~

Form item: "mtxMessage" = "Hello~
<script>alert('This is TEST')</script>"

Key: mtxMessage

Value: Hello~\r\n<script>alert('This is TEST')</script>

Form item: "btnSign" = "Sign Guestbook"

0260	65 3d 48 65 6c 6c 6f 25 37 45 26 6d 74 78 4d 65	e=Hello% 7E&mtxMe
0270	73 73 61 67 65 3d 48 65 6c 6c 6f 25 37 45 25 30	ssage=He llo%7E%0
0280	44 25 30 41 25 33 43 73 63 72 69 70 74 25 33 45	D%0A%3Cs cript%3E
0290	61 6c 65 72 74 25 32 38 25 32 37 54 68 69 73 2b	alert%28 %27This+
02a0	69 73 2b 54 45 53 54 25 32 37 25 32 39 25 33 43	is+TEST% 27%29%3C
02b0	25 32 46 73 63 72 69 70 74 25 33 45 26 62 74 6e	%2Fscrip t%3E&btn
02c0	53 69 67 6e 3d 53 69 67 6e 2b 47 75 65 73 74 62	Sign=Sig n+Guestb
02d0	6f 6f 6b	ook

XSS Attack Pattern Detection Rule

Transmission Control Protocol, Src Port: 46204, Dst Port: 80, Seq: 1, Ack: 1, Len: 502

Hypertext Transfer Protocol

GET /xss_test_server.php?xss_test=%3Cscript%3Ealert%28%27XSS+Test%27%29%3B%3C%2Fscript%3E

[Expert Info (Chat/Sequence): GET /xss_test_server.php?xss_test=%3Cscript%3Ealert%28%27XSS+Test%27%29%3B%3C%2Fscript%3E
Request Method: GET
Request URI: /xss_test_server.php?xss_test=%3Cscript%3Ealert%28%27XSS+Test%27%29%3B%3C%2Fscript%3E
Request URI Path: /xss_test_server.php
Request URI Query: xss_test=%3Cscript%3Ealert%28%27XSS+Test%27%29%3B%3C%2Fscript%3E
Request URI Query Parameter: xss_test=%3Cscript%3Ealert%28%27XSS+Test%27%29%3B%3C%2Fscript%3E

0050 73 65 72 76 65 72 2e 70 68 70 3f 78 73 73 5f 74 server.p hp?xss_t
0060 65 73 74 3d 25 33 43 73 63 72 69 70 74 25 33 45 est:%3C: cript%3E
0070 61 6c 65 72 74 25 32 38 25 32 37 58 53 53 2b 54 alert%28 %27XSS+T
0080 65 73 74 25 32 37 25 32 39 25 33 42 25 33 43 25 est%27%2 9%3B%3C%
0090 32 46 73 63 72 69 70 74 25 33 45 20 48 54 54 50 2Fscript %3E HTTP
00a0 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 39 32 2e /1.1..Ho st: 192.

<script>

① % 3C ~ %3E (공격자)

② 25 33 43 ~ 25 33 45 (16진수)

</script>

① % 3C % 2F (공격자)

② 25 33 43 25 32 46 (16진수)

문자	16진수
%	25
<	3C
/	2F
>	3E
2	32

문자	16진수
3	33
s	73
C	43
F	46
E	45

10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60	
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g
8	0x08	BS	40	0x28	(72	0x48	H	104	0x68	h
9	0x09	HT	41	0x29)	73	0x49	I	105	0x69	i
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	SUB	58	0x3A	:	90	0x5A	Z	122	0x7A	z
27	0x1B	ESC	59	0x3B	;	91	0x5B	[123	0x7B	{
28	0x1C	FS	60	0x3C	<	92	0x5C	\	124	0x7C	
29	0x1D	GS	61	0x3D	=	93	0x5D]	125	0x7D	}
30	0x1E	RS	62	0x3E	>	94	0x5E	~	126	0x7E	~
31	0x1F	US	63	0x3F	?	95	0x5F	_	127	0x7F	DEL

문자	16진수
%	25
<	3C
/	2F
>	3E
2	32
3	33
s	73
C	43
F	46
E	45

문자	문자+16진수	16진수
<script>	%3C script %3E	25 33 43 ~ 25 33 45
</script>	%3C2Fscript%3E	25 33 43 32 46

[Detection Rule 생성 예제]

alert tcp any any → any 80

(msg:“XSS Detect”;

content:“GET”; offset:0; depth:3; ❶

content:“/login.php?id=%3Cscript%3E”; distance:1; sid:1000500); ❷

❶ 첫바이트~ 세번째 바이트 내에서 GET이라는 문자 검색

❷ 첫번째 매치된 문자열에서 1바이트 떨어진 곳에서부터 해당 문자 검색

➔ Detection 이 안 될 경우 '대소문자' 옵션 첨부

[Detection Rule 생성 예제]

- 다음 조건을 만족하는 텔넷 접속 패킷들을 검사하는 snort rule을 작성
 - ❶ 텔넷 서비스 포트는 기본 포트(well-known port)를 사용
 - ❷ 탐지 시 alert를 발생시키고 이벤트 명으로 "Dangerous"를 사용
 - ❸ 첫 번째 바이트부터 14번째 바이트 범위 내에서 anonymous 문자열 패턴 검사

alert tcp any any → any 23

(msg:"Dangerous";

content:"anonymous"; offset:0; depth:14;)

[Detection Rule 생성 예제]

- OpenSSL 라이브러리의 하드비트(HeartBeat) 확장 모듈의 버그로 인해 발생하는 하트블리드(heartbleed) 취약점을 이용한 공격 패킷을 탐지하기 위한 snort 탐지룰이다.

```
alert tcp any any <> any [443,465,563]
```

```
(msg:“SSLv3 Malicious Heartbleed Request V2”;
```

```
content: “|18 03 00|”; depth:3;
```

```
content: “|01|”; distance:2, within:1;
```

```
content: ! “|00|”; within:1; sid:100300;)
```

- ❶ 첫 바이트부터 3바이트 범위 내에서 패턴 검사
- ❷ 첫 번째 content가 매치 된 이후 2바이트 떨어진 위치에서 1바이트 내에서 지정된 패턴 검사
- ❸ 두 번째 content가 매치된 이후 1바이트 떨어진 위치에서 지정된 패턴 검사

