

보안관제 최종 프로젝트 보고서

가상 인프라 구축 및 보안관제 탐지/분석 결과 보고서

프로젝트 기간	2024.01.17 ~ 2024.03.03
이름	김희중

2024.03.04

목차

1.프로젝트 개요

2.프로젝트 배경

2.1 프로젝트명 및 기간.....	3
2.2 프로젝트 목적.....	3
2.3 수행 인력 및 장소	3
2.4 수행 단계별 방법.....	4

3.프로젝트 환경 분석

3.1 요구사항 정의서	4
3.2 가상 인프라 구성도 및 시스템 장비.....	5
3.2.1 ESXi 전체 네트워크 구성	6
가. ESXi 네트워크 대역대별 상세 구성	
a.DMZ-NET	6
b.Trust zone: WAS·DB·SOC	6
c.Untrust zone(외부망).....	7
나. 시스템 장비 구성도	7
3.2.2 OpenWRT 전체 네트워크 구성	8
가. OpenWRT 네트워크 대역대별 상세 구성.....	8
나. 시스템 장비 구성도	9
a. 공유기	10
b. 스위치 허브	11
c. 서버 노트북	12
3.3 피해 서버 구성	12
3.3.1 피해 서버 정보	
3.3.2 서비스 기능	
3.4 방화벽 정책 설정	14
가. FW1-WAN	14
나. FW1-DMZ	15
다. FW1-LAN	16
라. FW2-WAN	16
마. FW2-WAS-Net	17

바. FW2-DB-NET	17
사. FW2-SOC-NET	17

4. 공격 유형에 따른 Snort 정책 설정

4.1 Scanning	18
가. Port Scanning	18
나. TCP XMASS Scan	19
다. TCP NULL Scan	19
라. TCP FIN Scan	19
4.2 DoS	20
가. ICMP Flooding Attack	20
나. Ping of Death.....	21
다. Slowloris(Slow HTTP Header DoS).....	22
4.3 Injection	21
가. OS Command Injection	24
나. SQL Injection	24
4.4 XSS	27
가. reflected XSS	27
나. SVG XSS(Stored XSS using SVG file)	27
다. Stored XSS	28

5. 통합보안관제시스템 구축

5.1 통합보안관제시스템(SIEM)을 위한 Splunk	29
5.1.1 목적	29
5.1.2 환경 구성	29
5.1.3 Splunk 기본 설정	30
5.2 보안관제 고도화를 위한 Splunk DashBoard	31
5.2.1 전체 DashBoard	31
5.2.2 WSR Guard DashBoard	31
5.2.3 WSR Guard 1.....	38
5.2.4 WEB Dashboard	48
5.2.5 WAS Dashboard	50

6.프로젝트 결론 및 산출물

6.1 프로젝트 결론	54
6.2 프로젝트 산출물	55

1.프로젝트 개요

보안관제 일련의 업무 과정인 환경구성, 공격, 탐지, 분석, 대응 등의 프로세스를 본 프로젝트를 통해 이해하고 보안관제 업무 수행 및 기본 역량을 갖추기 위해 수행하는 프로젝트

2.프로젝트 배경

2.1 프로젝트명 및 기간

프로젝트명	가상 인프라 구축 및 보안관제 탐지/분석
프로젝트 기간	2024.01.17 ~ 2024.03.03

2.2 프로젝트 목적

- 가. 보안관제를 수행하기 위한 가상 인프라를 직접 구축 및 관리하는 역량 강화
- 나. 공격 유형에 대응하기 위한 방화벽 정책 및 Snort 룰을 직접 작성을 통한 공격 행위에 대한 대응/탐지/분석
- 다. 보안관제 활동 일련의 과정을 보고서화하는 능력 수양
- 라. 공격에 대한 수집된 로그 분석 및 대시보드 생성

2.3 수행 인력 및 장소

- 가. 수행 인력: ‘SeSAC 클라우드 기반의 모빌리티 융합과정’ 최종 프로젝트 수행인원
- 나. 수행 장소: SeSAC 교육장내 물리적 서버를 설치하여 수행



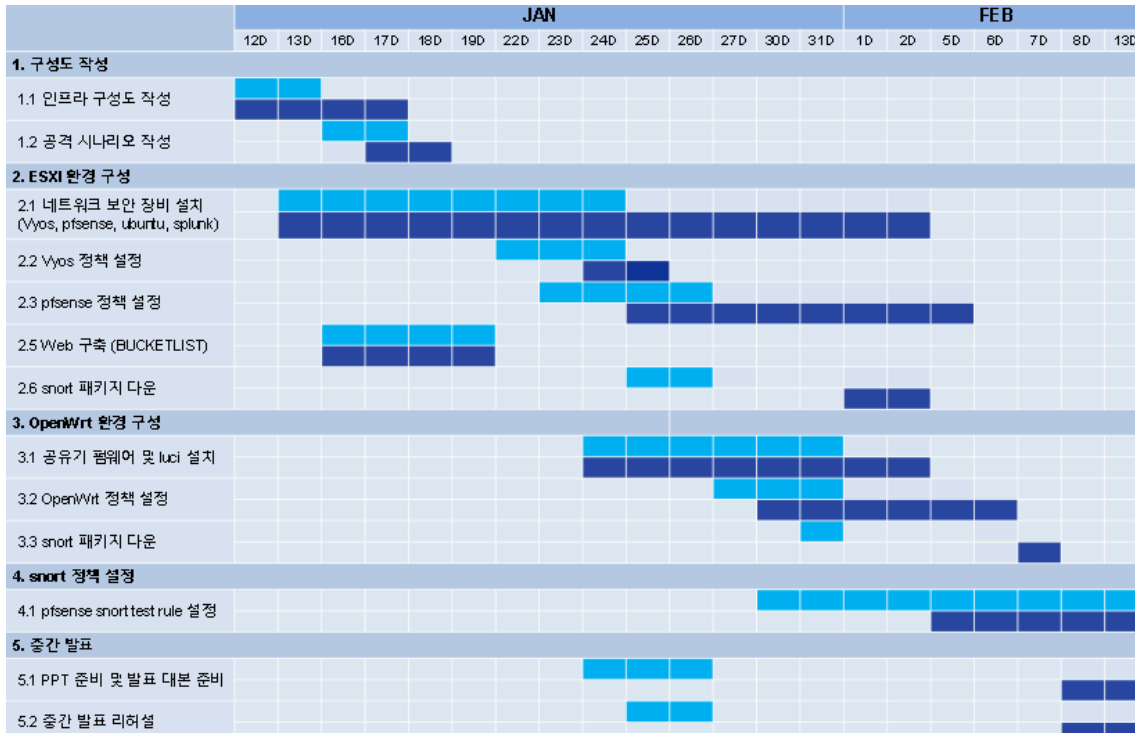
[그림1] 관제 서버



[그림2] 보안관제센터

2.4 수행 단계별 방법

프로젝트 수행 단계는 다음과 같은 WBS를 통해 프로젝트 작업 일정계획을 산정하고 전체일정 진행상황을 파악한다.



[그림3] 수행 단계

3. 프로젝트 환경 분석

3.1 요구사항 정의서

숙소 예약 시스템 웹사이트 'BucketList' 요구사항 정의서

작성자: 김희중 | 작성일: 2024.01.12

No	요구사항 ID	구분	요구사항명	상세 설명	수용여부
1	REQ_001	웹	공통	숙소 예약 서비스의 경우 아래와 같은 버전으로 구축 진행 -PC 사이트 버전(적용형 웹): Chrome, Safari	수용
2	REQ_002	웹	보안 가이드	관리자 사이트의 경우, 호스트 사용자용(숙박 예약 시스템 사업자용)과 내부 관리자용으로 분리가 필요하며 내부 관리자 사이트의 경우 접속 IP 제한이 필요	수용
3	REQ_003	웹	로그인/회원가입 OAuth	카카오톡, 네이버 계정으로 간편하게 회원가입 및 로그인 가능	수용
4	REQ_004	웹	로그인(일반 사용자)	아이디 및 비밀번호로 로그인 가능	수용
5	REQ_005	웹	비밀번호 재설정(일반 사용자)	일반 사용자의 경우, 비밀번호 재설정 기능 제공 a. 사용자 본인 이름, 이메일 주소, 전화번호를 입력하고 일치할 경우 비밀번호 재설정 인증 메일 발송 b. 인증 메일을 통한 비밀번호 재설정	수용
6	REQ_006	웹	GNB 메뉴-로그인/회원가입	로그인/회원가입을 위한 페이지로 구성	수용
7	REQ_007	웹	GNB 메뉴-예약	일반 사용자가 숙소 예약을 확인할 수 있는 메뉴 일반 사용자가 숙소 예약을 변경/취소 할 수 있는 기능 호스트 사용자가 숙소를 등록할 수 있는 기능 호스트 사용자가 숙소 예약건을 관리할 수 있는 기능 관리자가 모든 예약건을 관리하는 기능 관리자가 공지사항을 등록/수정할 수 있는 기능 관리자가 FAQ(자주 묻는 질문)를 등록/수정할 수 있는 기능 1:1 문의를 통해 질문/응답할 수 있는 기능 관리자가 리뷰 관리할 수 있는 기능	수용

[그림4] 요구사항 정의서1

숙박 예약 시스템 웹사이트(BucketList)를 구성하는 각 화면에서 필요한 기능을 모아 요구사항 정의서로 작성한다.

가상 인프라 구축 요구사항 정의서

작성자: 김희중 | 작성일: 2024.01.12

No	요구사항 ID	구분	요구사항명	상세 설명	수용여부
1	REQ_001	라우터	IP 대역 할당	WAN, LAN, DMZ 각 구역에 대해 NAT 룰을 활성화하여 IP 대역 할당	수용
2	REQ_002	라우터	우선 순위 적용	QoS(Quality of Service) 기능을 통해 트래픽 우선 순위 결정	수용
3	REQ_003	방화벽	정책 설정	내부망 및 외부망으로의 공격으로 부터 네트워크를 보호하기 위해 블랙리스트 보안조치를 통해 알려진 악성 IP 주소, 도메인, URL의 트래픽 차단	수용
4	REQ_004	방화벽	정책 설정	ACL(엑세스 제어 목록)을 통해 각 영역으로 송수신할 수 있는 트래픽 결정	수용
5	REQ_005	방화벽	정책 설정	Trust zone으로 들어오는 인터넷 회선을 방화벽을 통해 제어 가능	수용
6	REQ_006	방화벽	정책 설정	인터넷을 통한 서비스 사용을 위해 DMZ에 공개되는 서버(WEB)을 위치시킴	수용
7	REQ_007	방화벽	정책 설정	Trust zone 내부는 내부 방화벽을 통해 관제센터, DB-net, WAS-net 사이 통신 가능	수용
8	REQ_008	방화벽	정책 설정	외부 사용자는 인터넷을 통해 공개된 서버에 접근이 가능	수용
9	REQ_009	방화벽	정책 설정	방화벽 내부의 각 구역에 적합한 Snort 룰 작성	수용
10	REQ_010	방화벽	정책 설정	외부 사용자의 WEB의 웹사이트 접근에 대한 요청을 WAS 서버로 라우팅	수용
11	REQ_011	WAS	내/외부 통신	Tomcat을 통한 WEB-WAS, WAS-DB 로의 통신 가능	수용
12	REQ_012	DB	데이터 관리	Oracle을 통해 Data 구조화하여 DB서버에 데이터 저장/검색/수정/삭제 가능	수용
13	REQ_013	SOC	데이터 수집	방화벽에 올려진 가상머신, 네트워크 트래픽, 로그 데이터 등을 실시간 수집 및 통합해서 저장	수용
14	REQ_014	SOC	알람	공격 발생 시 대시보드에서 알람 생성	수용
15	REQ_015	SOC	모니터링	스플렁크 대시보드에 총 공격 발생건, 네트워크 구역별 공격 발생건, 최대 공격명 최대 공격 국가, 위험도 등을 시각화	수용

[그림5] 요구사항 정의서2

가상 인프라 환경 구축을 필요한 기능을 모아 요구사항 정의서로 작성한다.

3.2 가상 인프라 구성도

다음 두가지 관점에서 인프라 구성도를 작성한다.

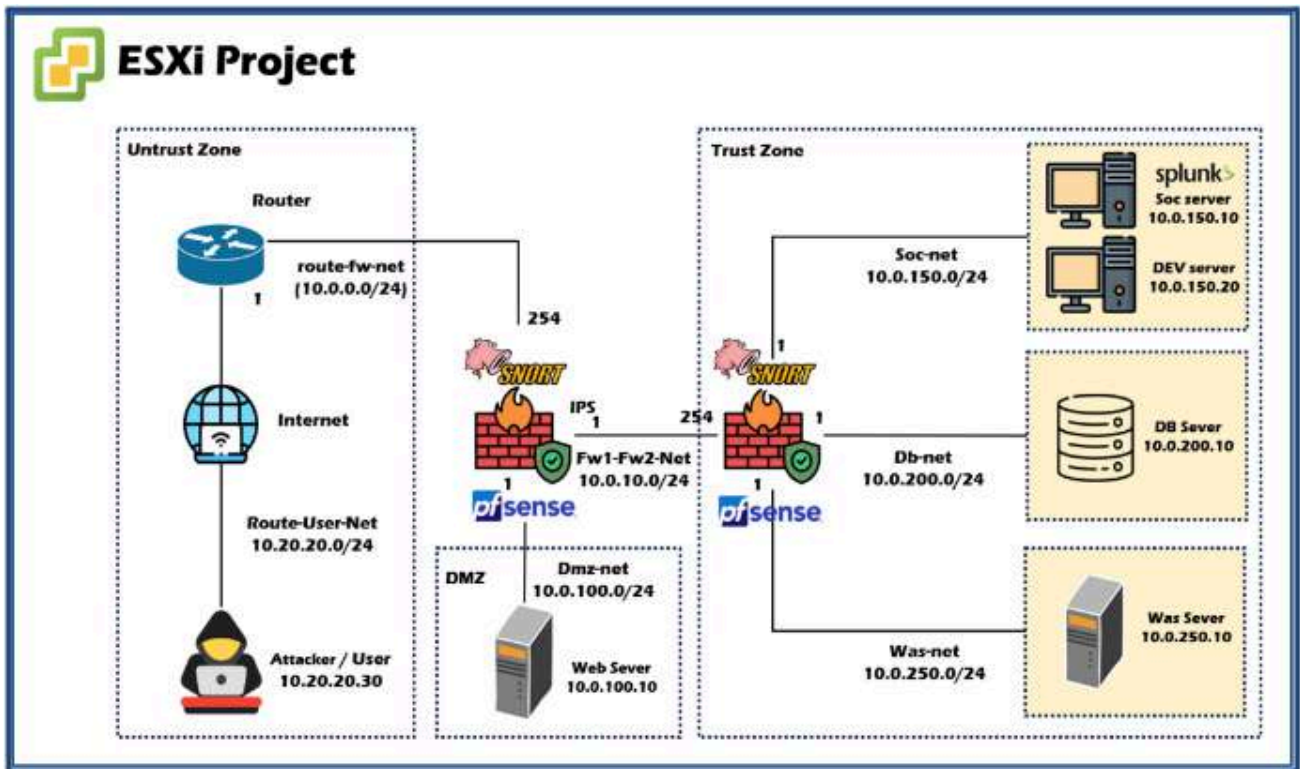
3.2.1 VMware에서 개발한 Hypervisor인 ESXi를 사용하여 OS에 종속되지 않고

하드웨어에 직접 설치되는 환경에서 프로젝트를 진행한다.

3.2.2 공유기에 리눅스 기반의 오픈 소스 운영체제인 OpenWrt를 올려 물리적으로

IDS/IPS 장비를 구축하여 프로젝트를 진행한다.

3.2.1 ESXi 전체 네트워크 구성



[그림6] ESXi 네트워크 구성

가. ESXi 네트워크 대역대별 상세 구성

a. DMZ-NET

- 외부에서 접근 가능한 웹 서버 및 애플리케이션 서버를 호스팅함으로써 외부 사용자는 접근 가능하지만 내부 네트워크는 노출되지 않는다.
- 외부 및 내부 네트워크 사이에 위치하기에 외부에서 내부로의 접근에 대해 인증 및 액세스 제어를 구현한다.

b. Trust zone: WAS·DB·SOC

앞서 ‘[그림5] 요구사항 정의서’에서 작성한 기능이 반영되도록 구현한다.

구성 요소	상세 설명
WAS	설명 Tomcat을 통한 WEB-WAS와 WAS-DB 간의 통신 데이터 저장, 검색, 수정, 삭제 등의 비즈니스 로직 처리
DB	Oracle을 통해 Data 구조화하여 저장 쿼리를 사용해 데이터저장, 검색, 수정, 삭제 등의 비즈니스 로직 처리 데이터베이스 관리, 백업 등 수행

SOC	가상머신 로그, 네트워크 트래픽 등 다양한 로그 데이터를 실시간 수집 및 통합, 저장하며 Splunk에서 통합적으로 수집 이벤트 발생 시 경고 및 로그파일에 로그 축적 이를 통한 대시보드 구축을 통해 가시적으로 모니터링할 수 있도록 함
-----	--

c. Untrust zone(외부망)

외부 네트워크(인터넷)와 직접적으로 연결된 영역

나. 시스템 장비 구성

다음은 ESXi에 올려진 가상 시스템 장비 목록이다.

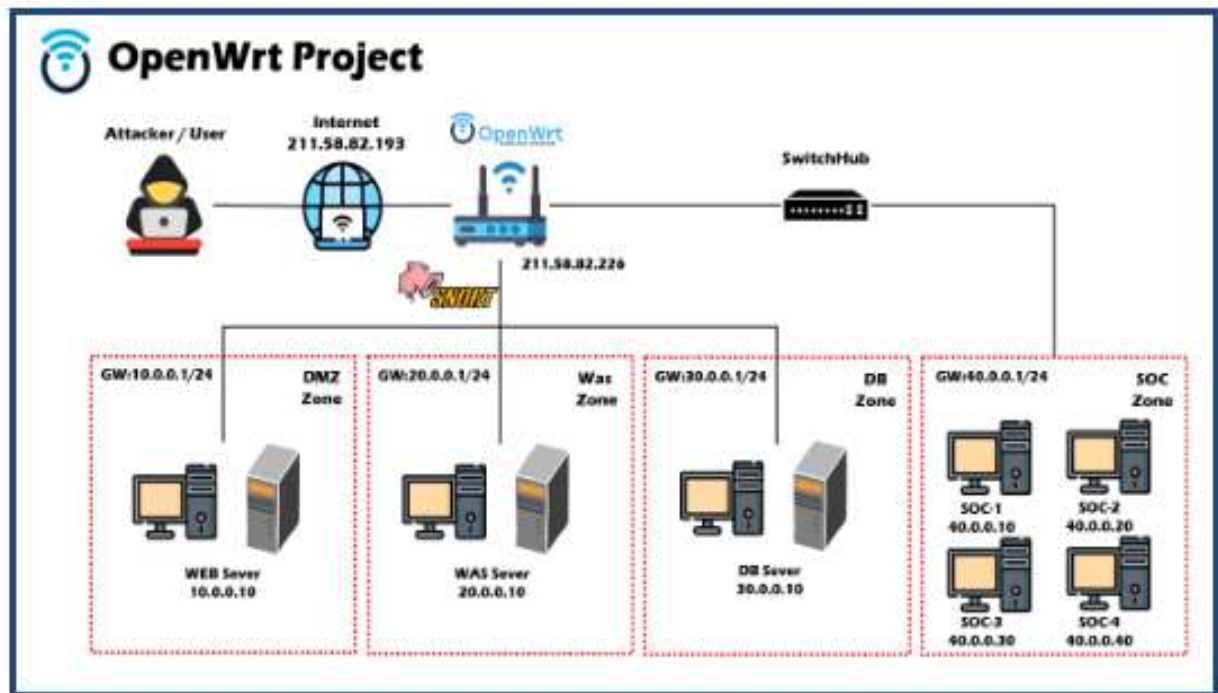
가상 시스템	상태	실행 상태
Router(VYOS)	▶ 전원 켜짐	✓ 정상
WAS1	▶ 전원 켜짐	✓ 정상
WEB1	▶ 전원 켜짐	✓ 정상
DB1	▶ 전원 켜짐	✓ 정상
SOC	▶ 전원 켜짐	✓ 정상
pfSense1	▶ 전원 켜짐	✓ 정상
attacker2	▶ 전원 켜짐	✓ 정상
pfSense2	▶ 전원 켜짐	✓ 정상
WAS2	▶ 전원 꺼짐	✓ 정상
WEB2	▶ 전원 꺼짐	✓ 정상
DB2	▶ 전원 꺼짐	✓ 정상
ATTACK	▶ 전원 꺼짐	✓ 정상

[그림7] ESXi 가상 시스템 장비

가상 시스템	장비	구성요소	IP 정보
Router	라우터	Vyos-1.1.8 os: Debian GNU/Linux 64bit	eth0: 211.58.82.203 eth1: 10.0.0.1 eth2: 10.20.20.1
pfSense1	방화벽	pfSense-CE-2.7.2 os: FreeBSD 14(64bit)	eth0: 10.0.0.254 eth1: 10.0.10.1 eth2: 10.0.100.1
pfSense2	방화벽	pfSense-CE-2.7.2 os: FreeBSD 14(64bit)	eth0: 10.0.10.254 eth1: 10.0.150.1 eth2: 10.0.200.1 eth3: 10.0.250.1
WEB	웹 서버	os: Ubuntu-22.04.3-desktop-amd64	10.0.100.10
WAS	WAS 서버	Tomcat os: Ubuntu-22.04.3-desktop-amd64)	10.0.250.10
DB	DB 서버	Oracle os: Ubuntu-22.04.3-desktop-amd64)	10.0.200.10
SOC	Splunk	Splunk os: Ubuntu-22.04.3-desktop-amd64)	10.0.150.10
Attacker	공격자VM	os: Ubuntu-22.04.3-desktop-amd64	10.20.20.70

[그림8] 장비 목록

3.2.2 OpenWrt 전체 네트워크 구성



[그림9] OpenWrt 네트워크 구성

가. OpenWrt 네트워크 대역대별 상세 구성

가. DMZ zone

Interfaces » DMZ

General Settings | Advanced Settings | Firewall Settings | DHCP Server

Status: Device: br-lan.1
Uptime: 8d 8h 3m 29s
MAC: 90:9F:33:17:F0:F3
RX: 363.81 MB (628837 Pkts.)
TX: 616.76 MB (594573 Pkts.)
IPv4: 10.0.0.1/24

Protocol: Static address

Device: br-lan.1

Disable this interface: ☐

Bring up on boot: ☒

IPv4 address: 10.0.0.1/24


[그림10] OpenWrt DMZ

b.WAS zone

Interfaces » WAS

General Settings Advanced Settings Firewall Settings DHCP Server

Status


Device: br-lan.2
Uptime: 8d 8h 28m 20s
MAC: 90:9F:33:17:F0:F3
RX: 405.56 MB (897028 Pkts.)
TX: 547.64 MB (777863 Pkts.)
IPv4: 20.0.0.1/24

Protocol

Static address ▼

Device

br-lan.2 ▼


Disable this interface

☐

Bring up on boot

☒

IPv4 address

20.0.0.1/24 

[그림 11] OpenWrt WAS

c.DB zone

Interfaces » DB

General Settings Advanced Settings Firewall Settings DHCP Server

Status


Device: br-lan.3
Uptime: 2d 20h 47m 3s
MAC: 90:9F:33:17:F0:F3
RX: 53.13 MB (807252 Pkts.)
TX: 54.39 MB (695868 Pkts.)
IPv4: 30.0.0.1/24

Protocol

Static address ▼


Device

br-lan.3 ▼

Bring up on boot

☒


IPv4 address

30.0.0.1/24 

[그림 12] OpenWrt DB

나. 시스템 장비 구성

a. 다음과 같이 장비를 구성한다.

장비	장비 사진	상세 정보
a. 공유기	 <p>[그림13] OpenWrt 공유기</p>	<p>[공유기 정보] Iptime A3004T</p> <p>[구성 요소] OpenWrt SNAPSHOT r25252-63f7ced2f0</p>

- 1) Iptime A3004T 공유기에 기존에 설치된 Windows OS를 제거하고 Linux 기반의 OpenWrt SNAPSHOT r25252-63f7ced2f0를 설치한다.
(최신 버전 3.25.0가 아닌 개발자 버전 사용으로 RAM 최적화를 고려)
- 2) 각 포트에 VLAN 1,2,3,4 대역 설정을 통해 포트 별 WEB,DB,WAS,SOC 서버 역할을 부여한다.

VLAN ID	Local	lan1	lan2	lan3	lan4	
		1000FD	1000FD	1000FD	1000FD	
1	✓	U	-	-	-	Delete
2	✓	-	U	-	-	Delete
3	✓	-	-	U	-	Delete
4	✓	-	-	-	U	Delete


[그림14] OpenWrt VLAN 설정

Devices		
Device	Type	MAC Address
br-lan	Bridge device	90:9F:33:17:F0:F3
br-lan.1	VLAN (802.1q)	-
br-lan.2	VLAN (802.1q)	-
br-lan.3	VLAN (802.1q)	-
br-lan.4	VLAN (802.1q)	-
eth0	Network device	90:9F:33:17:F0:F3
lan1	Network device	90:9F:33:17:F0:F3
lan2	Network device	90:9F:33:17:F0:F3
lan3	Network device	90:9F:33:17:F0:F3
lan4	Network device	90:9F:33:17:F0:F3
wan	Network device	90:9F:33:17:F0:F1


[그림15] OpenWrt VLAN 설정 후

3) Snort3 최신 버전 설치를 통해 이전 버전보다 많은 Snort community rule 적용

b. Iptime H6008 스위치 허브를 통해 네트워크 대역대를 설정하여 추가로 연결된 노트북SOC 서버로 사용하도록 설정한다.

장비	장비 사진	상세 정보
b.스위치 허브	 <p>[그림16] 스위치 허브</p>	<p>[공유기 정보] Iptime H6008</p>

- c. 기존에 존재하는 노트북 3대와 스위치 허브 연결을 통해 추가된 노트북 1대를 기준으로 물리적 보안관제센터를 운영하기 위해 필요한 서버를 구축한다.

서버	 <p>[그림17] WEB,DB,WAS Server(3 tier 구성)</p>	<p>[Hypervisor] VMware workstation 17.0.0 build-20800274</p> <p>[OS] Ubuntu 22.04 64bit</p>
----	--	---

3.3 피해 서버 구성

보안관제 프로젝트를 진행하기 위해서는 공격 대상 서버가 필요하다.

3.3.1 피해 서버 정보

[피해 서버 이름] 숙소 예약 서비스 웹사이트 'BucketList'

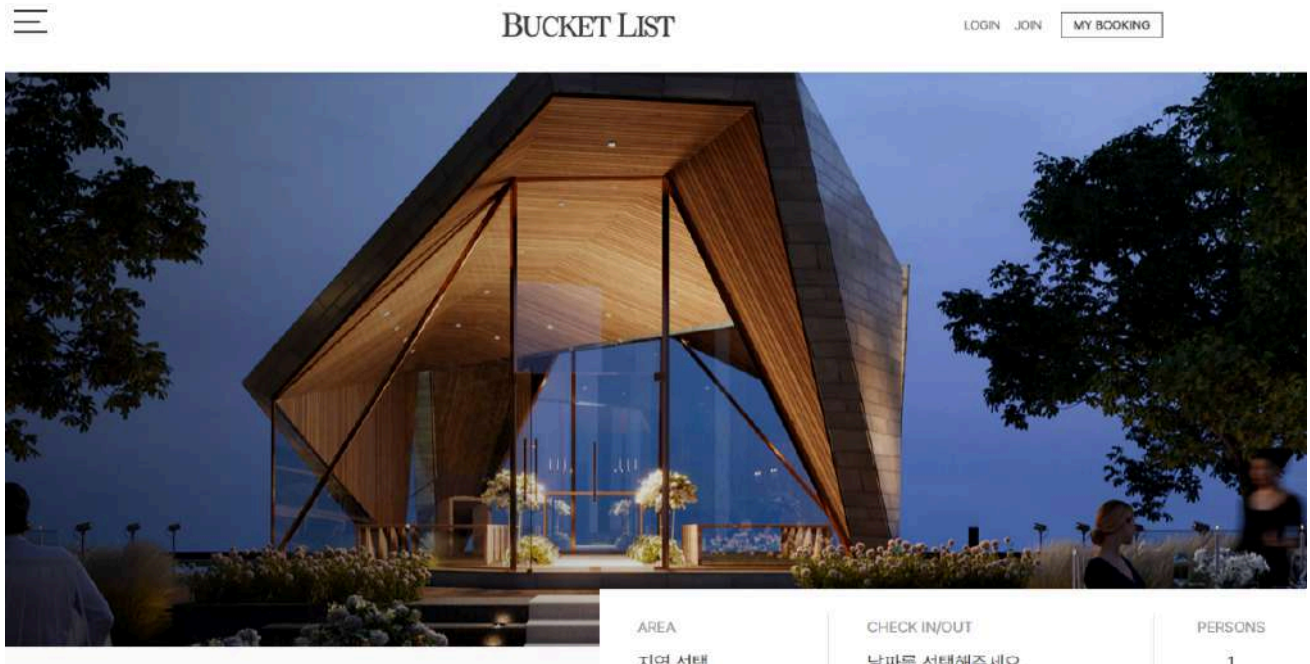
[URL] <http://10.0.100.10/bucketlist/>

구분	구성요소
WEB	nginx-1.20.2 docker container 4.25.0
WAS	Tomcat 8.5 docker container 4.25.0
DB	kasmweb/oracle-9-desktop docker container 4.25.0

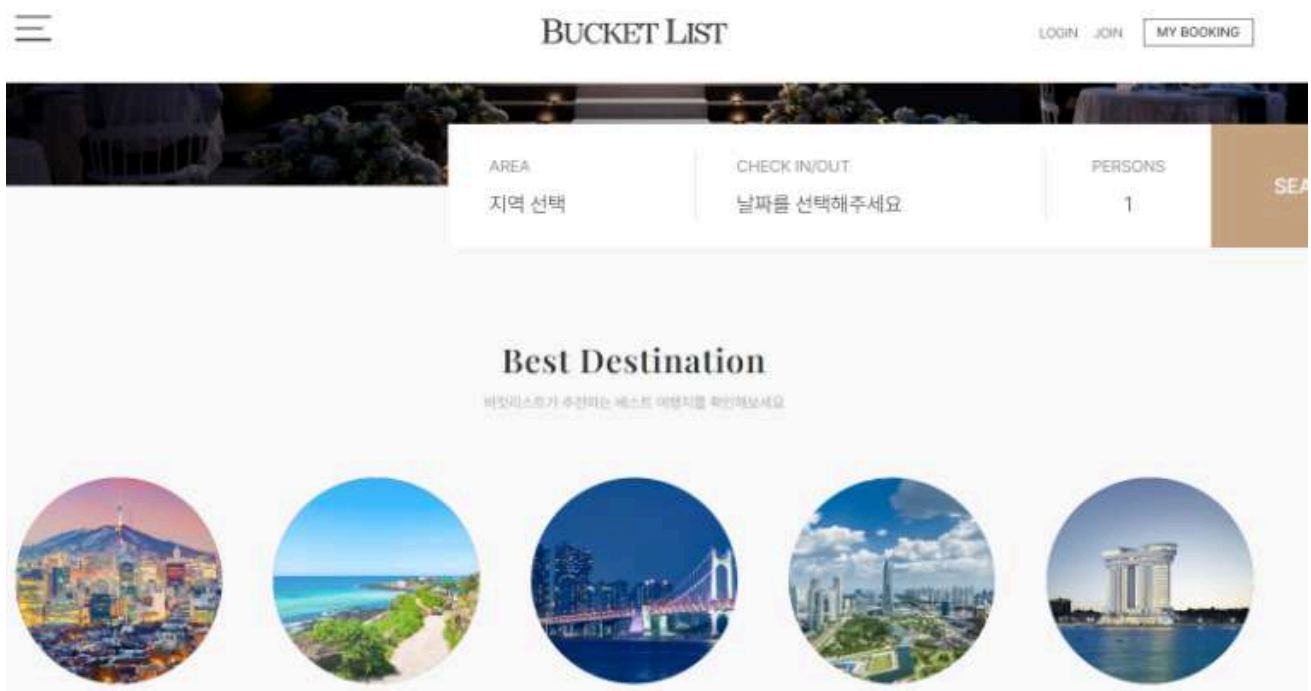
[그림18] 3-tier 구성요소

3.3.2 서비스 기능

‘[그림4] 요구사항 정의서1’ 에서 작성한 기능이 반영되도록 구현한다.



[그림19] 웹 사이트



[그림20] 웹 사이트

3.4 방화벽 정책 설정

가.FW1-WAN

Rules

<

[그림21] pfSense1-WAN 방화벽 정책

외부인터넷망과 직접적으로 연결되는 pfSense1(방화벽1)의 WAN 구간은 Default Deny로 모든 트래픽을 차단하는 룰을 가장 아래에 설정하고, 외부에서 WEB Server(10.0.100.10)으로 접근할 수 있도록 80/HTTP 포트만 열어준다.

나.FW1-DMZ

Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	<div><div>✓</div><div>☰</div></div>	0/0 B	IPv4 TCP	10.0.100.10	*	10.0.250.10	*	*	none	TCP Web -> Was	<div><div>📌</div><div>✎</div><div>📄</div><div>🔗</div><div>🗑</div><div>✕</div></div>
<input type="checkbox"/>	<div><div>✓</div><div>☰</div></div>	0/46.25 MiB	IPv4 TCP	DMZ subnets	*	10.0.10.254	9997	*	none	Allow splunk-forwarder	<div><div>📌</div><div>✎</div><div>📄</div><div>🔗</div><div>🗑</div><div>✕</div></div>
<input type="checkbox"/>	<div><div>✓</div><div>☰</div></div>	0/0 B	IPv4 ICMP any	10.0.100.10	*	10.0.250.10	*	*	none	ICMP Web -> Was	<div><div>📌</div><div>✎</div><div>📄</div><div>🔗</div><div>🗑</div><div>✕</div></div>
<input type="checkbox"/>	<div><div>✓</div><div>☰</div></div>	0/0 B	IPv4 TCP	DMZ subnets	*	10.0.100.1	443 (HTTPS)	*	none	Allow_Web_Site_Access	<div><div>📌</div><div>✎</div><div>📄</div><div>🔗</div><div>🗑</div><div>✕</div></div>
<input type="checkbox"/>	<div><div>✗</div><div>☰</div></div>	0/56 KiB	IPv4+6 *	*	*	*	*	*	none	Default Deny	<div><div>📌</div><div>✎</div><div>📄</div><div>🔗</div><div>🗑</div></div>

⬆️ Add

⬇️ Add

🗑 Delete

🔗 Toggle

📄 Copy

💾 Save

➕ Separator

[그림22] pfSense1-DMZ 방화벽 정책

pfSense1의 DMZ 구간은 Default Deny로 모든 트래픽을 차단하는 룰을 가장 아래에 설정하고, WEB Server에서 WAS Server(10.0.250.10)으로 TCP/ICMP 통신이 될 수 있게 구간을 열어 놓는다.

WEB Server에서 pfSense1 방화벽에 접속할 수 있도록 10.0.100.1:443을 열어준다. Web Server의 가상머신 로그를 SOC(Splunk) Server로 보내주기 위해 10.0.10.254:9997로 forwarder 설정한다.

다.FW1-LAN

Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/9.11 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	7/12.61 MiB	IPv4 *	LAN subnets	*	*	*	*	none		Default allow LAN to any rule	
<input type="checkbox"/>	0/0 B	IPv6 *	LAN subnets	*	*	*	*	none		Default allow LAN IPv6 to any rule	
<input type="checkbox"/>	0/324 B	IPv4+6 *	*	*	*	*	*	none		Default Deny	
Add Add Delete Toggle Copy Save Separator											

[그림23] pfSense1-LAN 방화벽 정책

pfSense1의 LAN 구간에는 Default Deny로 모든 트래픽을 차단하는 룰을 가장 아래에 적용하고, 내부망에서 오는 모든 요청을 전부 열어놓기 위해 LAN subnets의 IPv4, IPv6를 허용한다.

라.FW2-WAN

Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	0/4 KiB	IPv4 UDP	*	*	10.0.150.10	513	*	none		NAT PfSense_log	
<input type="checkbox"/>	0/0 B	IPv4 TCP	10.0.100.10	*	10.0.250.10	*	*	none		TCP_Web->Was	
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	10.0.150.10	9997	*	none		NAT PfSense_log	
<input type="checkbox"/>	0/4 KiB	IPv4+6 *	*	*	*	*	*	none			
<input type="checkbox"/>	0/0 B	IPv4+6 *	*	*	*	*	*	none		Default Deny	
Add Add Delete Toggle Copy Save Separator											

[그림24] pfSense2-WAN 방화벽 정책

pfSense2 방화벽의 DMZ 구간에는 Default Deny 룰로 모든 트래픽을 차단하는 룰을 가장 아래에 설정하고, WEB Server(10.0.100.10)에서 WAS Server(10.0.250.10)로 TCP 통신이 가능하게 하고, SOC(10.0.150.10) Server로 513,9997 포트를 허용시켜 pFSense2 방화벽 로그가 Splunk로 전송되는 것을 허용한다.

마.FW2-WAS-Net

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	0/64.43 MiB	IPv4 TCP	WAS subnets	*	10.0.150.10	9997	*	none		Allow splunk-forwarder	
<input type="checkbox"/>	0/1.81 MiB	IPv4 TCP	10.0.250.10	*	10.0.200.10	*	*	none		TCP was1 -> db1	
<input type="checkbox"/>	0/0 B	IPv4 TCP	10.0.250.10	*	10.0.100.10	*	*	none		TCP was1 -> web1	
<input type="checkbox"/>	0/8.86 MiB	IPv4+6 *	*	*	*	*	*	none			

Add
 Add
 Delete
 Toggle
 Copy
 Save
 Separator

[그림25] pfSense2-WAS 방화벽 정책

pfSense2 방화벽의 WAS 구간에는 Default Deny 룰로 모든 트래픽을 차단하는 룰을 가장 아래에 설정하고, WAS(10.0.250.10) Server와 DB(10.0.200.10) Server가 서로 TCP 통신이 가능하게 설정한다. WAS Server의 로그들이 SOC Server로 전송되도록 9997 포트를 허용해준다.

바.FW2-DB-NET

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	4/86 KiB	IPv4 TCP	DB subnets	*	10.0.150.10	9997	*	none		Allow Splunk-Forwarder	
<input type="checkbox"/>	0/0 B	IPv4 TCP	10.0.200.10	*	10.0.200.1	443 (HTTPS)	*	none		HTTPS DB -> PfSense	
<input type="checkbox"/>	0/0 B	IPv4 TCP	10.0.200.10	*	10.0.250.10	*	*	none		TCP DB -> WAS	
<input type="checkbox"/>	0/0 B	IPv4 ICMP any	10.0.200.10	*	10.0.250.10	*	*	none		ICMP DB -> WAS	
<input type="checkbox"/>	0/10.60 MiB	IPv4+6 *	*	*	*	*	*	none			

Add
 Add
 Delete
 Toggle
 Copy
 Save
 Separator

[그림26] pfSense2-DB 방화벽 정책

pfSense2 방화벽의 WAS 구간에는 Default Deny 룰로 모든 트래픽을 차단하는 룰을 가장 아래에 설정하고, DB(10.0.200.10) Server와 WAS(10.0.250.10) Server 간에 TCP 통신 설정을 한다. 두 서버가 정상적인 통신을 하는지 확인하기 위해 ICMP 통신도 허용한다. DB Server에서 pfSense에 접속할 수 있도록 443 포트 허용 및 DB Server에서 SOC Server로 로그 전송이 가능하게 9997 포트 허용해준다.

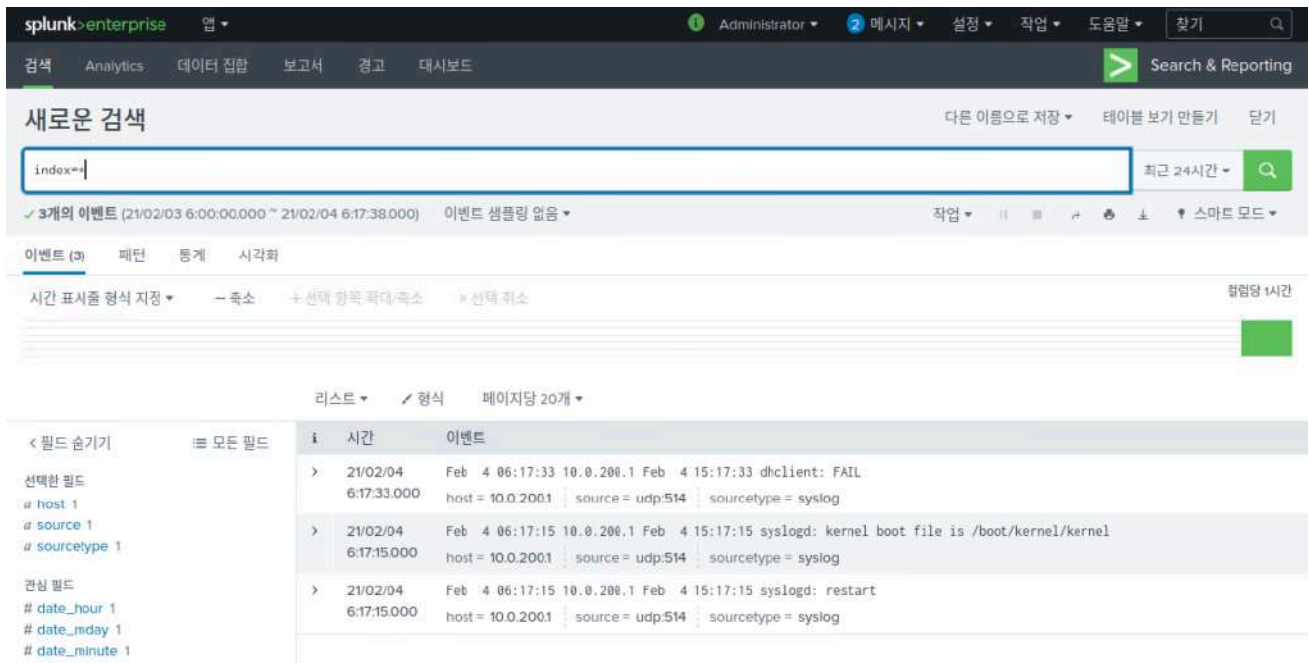
사.FW2-SOC-NET

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/0 B	*	*	*	SOC Address	443 80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	6/2 KiB	IPv4 UDP	*	*	*	*	*	none		UDP any pass	
<input type="checkbox"/>	0/0 B	IPv4 ICMP any	*	*	*	*	*	none		ICMP any Pass Soc	
<input type="checkbox"/>	4/14.22 MiB	IPv4 TCP	*	*	*	*	*	none		TCP any Pass SOC	
<input type="checkbox"/>	0/36 B	IPv4+6 *	*	*	*	*	*	none		Default Deny	

Add
 Add
 Delete
 Toggle
 Copy
 Save
 Separator

[그림27] pfSense2-SOC 방화벽 정책

pfSense2 방화벽의 SOC 구간에서는 TCP/ICMP/UDP 모든 통신이 양방향으로 가능하게 설정한다.



[그림28] Splunk에서 log 확인

4. 공격 유형에 따른 Snort 정책 설정

Network, Web을 기준으로 들어오는 공격을 분류하여 공격 구문을 집어넣고 공격을 시도한다. 공격 유형에 따른 Snort 룰 정책을 설정하여 해당 공격이 들어왔을때 즉각적으로 탐지가 가능하도록 한다.

4.1 Scanning

가. Port Scanning

공격명칭(중분류)	공격명칭(소분류)	공격 구문
스캐닝(sanning)	Port Scanning	nmap -sS 10.0.10.0/24 → FW1 방화벽을 통해 모든 대역에서 외부로 나가는 패킷 중 'FW1-FW2-Net' 대역대를 공격

Snort rule
alert tcp any any → any any (msg: "Port Scanning Possible"; flag:S; threshold: type both, track by_dst. count 50, seconds 5; sid:1000001; rev:1;)
[설명]

내부대역인 FW1-FW2-net 대역(10.0.10.0/24)/DMZ-net/SOC-net/DB-net/WAS-net 에서 외부로 나가는 패킷 중 각 대역에서 나가는 패킷의 모든 포트를 검출하고 검출하면 msg를 띄우고 5초간 50번 횟수인 도착지 패킷을 탐지한다.

나. TCP XMASS Scan

공격명칭(중분류)	공격명칭(소분류)´	공격 구문
스캐닝(sanning)	Port Scanning	nmap -sX -p 10.0.150.10 nmap -sX -p 10.0.200.10 namp -sX -p 10.0.250.10

Snort rule
alert tcp any any -> any any (msg:"TEST:XMAS SCAN Possible"; flags:FPU; sid:1000002; rev:1;)
[설명] TCP 헤더 flag에 FPU(FIN, PUT, URG) 설정을 해서 패킷을 보내는 스캔 공격으로, FPU flags가 설정된 패킷을 공격대상(SOC, DB, WAS)가 받으면 해당 포트가 오픈된 경우 에는 패킷을 DROP(버리기), 포트가 닫혀있는 경우에는 RST-ACK 패킷을 응답한다(response).

다. TCP NULL Scan

공격명칭(중분류)	공격명칭(소분류)´	공격 구문
스캐닝(sanning)	Port Scanning	nmap -sN 10.0.150.10 nmap -sN 10.0.200.10 namp -sN 10.0.250.10

Snort rule
alert tcp any any -> any any (msg:"TEST:XMAS SCAN Possible"; flags:FPU; sid:1000002; rev:1;)
[설명] tcp 헤더 flag에 어떤 flag도 설정하지 않고(0) 패킷을 보내는 스캔 공격으로, 이 패킷을 받은 피해대상은 Xmas Scan과 동일하게 응답한다. alert 로그파일에 " * * * ... " 같이 어떠한 패킷도 기록되는지 확인한다.

라. TCP FIN Scan

공격명칭(중분류)	공격명칭(소분류)´	공격 구문
스캐닝(sanning)	Port Scanning	nmap -sF 10.0.150.10 nmap -sF 10.0.200.10 namp -sF 10.0.250.10

Snort rule
alert tcp any any -> any any (msg:"TEST:FIN SCAN Possible"; flags:F; sid:1000004; rev:1;)
<p>[설명]</p> <p>tcp 헤더 flag에 FIN만 설정해서 패킷을 보내는 스캔 공격으로, 이 패킷을 받은 공격대상은 Xmas Scan과 동일하게 응답한다. .</p>

4.2 DoS

가. ICMP Flooding Attack

공격명칭 (중분류)	공격명칭(소분류)	공격 구문
DoS	ICMP Flooding Attack	<p>[구문1] hping3 10.0.100.10 -a 10.0.100.10 --icmp --flood -d 65000</p> <p>[설명] Source IP주소와 Destination IP주소 값을 Target IP주소 값으로 같게 만드는 Land Attack + ICMP Flooding Attack을 동시에 적용</p> <hr/> <p>[구문2] hping3 10.0.100.255 -a 10.0.100.10 --icmp --flood -d 65000</p> <p>[설명] ICMP Flooding 공격을 변형한 ICMP 스퍼핑 공격. 공격자가 전송한 ICMP Request 패킷은 10.0.100.0/24 대역에 있는 모든 호스트에게 전달되어지며, ICMP 요청을 받은 호스트는 Source IP주소를 참조해 ICMP 응답 패킷을 전송하기에 피해대상은 어느 순간불처 10.0.100.0/24 대역의 모든 호스트로부터 ICMP 응답 패킷을 받아 과부하가 발생한다.</p> <hr/> <p>[구문3] hping3 10.255.255.255 -a 10.0.100.10 --icmp --flood -d 65000</p> <hr/> <p>[구문4] hping3 --rand-source 10.0.100.13 -S --flood -p 80</p> <p>[설명] 조작된 Source IP로부터 서버가 SYN 패킷을 전송받고 TCP 3-Way-Handshaking을 위해 SYN-ACK flag를 가짜 출발지로</p>

		전송한다. 가짜 출발지이기에 ACK 응답을 받지 못하고 연결할때까지 기다리게 되는데 이는 백로그큐에 syn_recv 대기열이 길어져 할당된 가용 queue 메모리를 전부 소진하게 만드므로 다른 사용자들의 접속 제한이 생긴다.
--	--	--

Snort rule		
alert icmp any any -> any any (msg:"ICMP Flooding Possible"; threshold: type threshold, track by_dst, count 10, seconds1; sid:2000010; rev:1;)		
[설명]) type threshold 옵션을 주면 count 횟수마다 탐지를 하는데, count 10이면 icmp_seq10, icmp_seq=20 이렇게 두번 잡힌다.		
alert icmp any any -> any any (msg:"ICMP Flooding Possible"; threshold: type both, track by_src, count 20, seconds 5; sid:2000012;; rev:1;) alert icmp any any -> any any (msg:"ICMP Flooding Possible"; threshold:type both, track by_src, count 20, secound 5; sid:2000003;)		
[설명] Land Attack rule을 추가한 뒤 공격을 재시행하고, sudo snort -q -A console -b -c /etc/snort/snort.conf -i [인터페이스명 eth0] 디버깅 하여 출력된 드롭율(%) 확인		
drop tcp any any -> any any (msg:"SYN Flood Attack Possible"; flag:S; threshold: type both, track by_dst, count 3, seconds 1; sid: 2000015; rev:1;)		
[설명] SYN Flood Attack이 랜덤한 ip로 n번 포트에 공격을 해오는 경우 flag가 S이면 SYN 패킷이 반복적으로 오므로 막아준다.		

나. Ping of Death

공격명칭(중분류)	공격명칭(소분류)	공격 구문
DoS	Ping of Death	hping3 --icmp 10.0.100.10 -d 65000 --flood hping3 --icmp 10.0.250.0 -d 65000 --flood hping3 --icmp 10.0.150.0 -d 65000 --flood hping3 --icmp 10.0.200.0 -d 65000 --flood

Snort rule		
drop ip any any -> any any (msg:"Ping of Death Attack Possible"; content:"[5858585858]"; threshold:type both, track by_src, count 10, seconds:5; sid: 2000016; rev:1;)		

[설명]

공격명령어를 실행 후 패킷을 캡쳐해보면 데이터에 X문자열(58)을 채워서 보내는 경우를 확인할 수 있다.
이 경우 content에 문자열을 넣어 drop 시킨다.

다. Slowloris(Slow HTTP Header DoS)

공격명칭(중분류)	공격명칭(소분류)	공격 구문
DoS	Slow HTTP Header DoS	slowhttptest -c 4000 -H -g -o slowloris -i 10 -r 200 -t GET -u http://10.0.0.254/ -x 24 -p 3

Snort rule
alert tcp any any → any 80 (msg:"SlowLoris DoS Attack Detected Possible"; flow:to_server,established; pcre:"/^[^x0d\x0a]\x0d\x0a\$/"; threshold: type both, track by_dst, count 30, seconds 3; sid: 2000017; rev:1;)
alert tcp any any -> any 80 (msg:"SlowLoris DoS Attack Detected Possible"; flow:established, to_server; dsize:8; content:"X-a 3a20 b 0d0a "; depth:8; sid:2000018; rev:1;)
alert tcp any any → any any (msg:"SlowLoris DoS Attack Detected Possible"; flow:to_server,established,no_stream; content:"X-a:"; dsize:<15; threshold:track by_dst, count3, seconds 30; classtype:denial-of-service; sid:2000019; rev:1;)
any any -> any 80 (msg:"SlowLoris DoS Attack Possible"; flow:established, to_server; content:"User-Agent 3A Mozilla/4.0 28 compatible 3B MSIE 7.0 3B Windows NT 5.1 3B Trident/4.0 3B .NET CLR 1.1.4322 3B .NET CLR 2.0.50313 3B .NET CLR 3.0.4506.2152 3B .NET CLR 3.5.30729 3B MSOffice 12 29 0D 0A "; content:"Content-Length 3A 42 0d0a "; distance:0; threshold: type both, track by_dst, count 50, seconds 2; sid:2000020)

위의 Slowloris 공격을 예시로 공격 시도하고 적용한 스노트룰이 탐지되는지 확인해본다.

1) Slowloris 공격이란?

HTTP, GET 메소드를 사용해서 헤더의 최종 끝을 알리는 ‘\r\n\r\n’, Hex: \0d\0a\0d\0a’를 전송하지 않고 ‘\r\n’, ‘\0d\0a’ 만 전송하는 공격으로 연결 자원을 모두 소진시키는 서비스 거부 공격(DoS)이다.

2) Slowloris 공격 취약점

HTTP 헤더는 마지막 ‘\r\n\r\n’, ‘\0d\0a\0d\0a’ 로 종결되는 것이 정상적이다.

2번의 개행문자가 없을 경우 웹 서버는 HTTP 헤더 정보가 다 수신하지 않은 것으로 판단하여 연결을 유지한다. 이는 연결상태가 장시간 지속되는 Connection full 상태를 만들어서 웹 서버가 다른 클라이언트에 대한 정상적인 연결을 제공하지 못하게 하는데 해당 공격에

취약하다고 볼 수 있다.

3) 공격 방법

HTTP GET 메소드 형태로 가장해 HTTP GET Flooding Attack을 수행한다.

공격하고자 하는 대상 서버 도메인, URL을 사용한다.

4) 공격 탐지

패킷량을 GUI 형태로 보기 위해서 Etherape 도구를 활용한다.

네트워크를 통해 공격자 IP(10.20.20.30)에서 피해 서버(10.0.0.254)로 이동하는 트래픽을 모니터링할 수 있다.



[그림29] Slowloris 공격 모니터링

Snort Alert Log를 확인해보면 다음과 같이 공격 탐지되는 것을 볼 수 있다.

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2024-02-22 15:17:28	⚠	2	TCP	Detection of a Denial of Service Attack	10.0.0.1	40618	10.0.0.254	80	1:100010	[Slowloris DoS Attack detected]
2024-02-22 15:17:10	⚠	0	TCP		10.0.0.1	35148	10.0.0.254	80	1:100009	[Slowloris DoS Attack detected]
2024-02-22 15:17:07	⚠	0	TCP		10.0.0.1	58766	10.0.0.254	80	1:100009	[Slowloris DoS Attack detected]
2024-02-22 15:17:04	⚠	0	TCP		10.0.0.1	45994	10.0.0.254	80	1:100009	[Slowloris DoS Attack detected]
2024-02-22 15:17:01	⚠	0	TCP		10.0.0.1	41532	10.0.0.254	80	1:100009	[Slowloris DoS Attack detected]
2024-02-22 15:16:59	⚠	0	TCP		10.0.0.1	38530	10.0.0.254	80	1:100009	[Slowloris DoS Attack detected]
2024-02-22 14:40:45	⚠	0	TCP		10.0.0.1	35822	10.0.0.254	80	1:100006	Blind SQL Injection Detector
2024-02-22 14:41:43	⚠	0	TCP		10.0.0.1	58732	10.0.0.254	80	1:100009	[Slowloris DoS Attack detected]
2024-02-22 14:41:33	⚠	0	TCP		10.0.0.1	58768	10.0.0.254	80	1:100009	[Slowloris DoS Attack detected]

[그림30] Slowloris 공격 탐지 Log

4.3 Injection

가. OS Command Injection

공격명칭(중분류)	공격명칭(소분류)	공격 구문
Injection	OS Command Injection	<pre>&cat /etc/passwd& ;cat /etc/passwd; &&cat /etc/passwd head -3 "&cat /etc/passwd&" ";cat /etc/passwd;" '&cat /etc/passwd&' ';cat /etc/passwd;' &sleep 15& "&sleep 15&" &type &SYSTEMROOT%\win.ini& type &SYSTEMROOT%\win.ini& "&type %SYSTEMROOT%\win.ini&" " type %SYSTEMROOT%\win.ini" '&type %SYSTEMROOT%\win.ini&' ' type %SYSTEMROOT%\win.ini'</pre>

Snort rule	
alert tcp EXTERNAL_NET any → \$HOME_NET any (msg:" OS Command Injection URI Possible"; pcre:"(ls dir cat tail head type chmod)\x20.*[\x2f\x5c]/UI", sid:4000001; rev:1;)	
alert tcp any any → \$HOME_NET any (msg:"OS Command Injection URI \ Possible"; flow:to_server,established; content: "%3b"; nocase: http_uri; sid:4000002; rev:1;)	
alert tcp any any → \$HOME_NET any (msg:"OS Command Injection URI \&& Possible"; flow:to_server,established; content: "%26%26 "; nocase: http_client_body; sid:4000006; rev:1;)	
alert tcp any any → \$HOME_NET any (msg:"OS Command Injection URI V Possible"; flow:to_server,established; content: "%2f"; nocase: http_client_body; sid:4000007; rev:1)	

나. SQL Injection

공격명칭(중분류)	공격명칭(소분류)	공격 구문
Injection	SQL Injection	<pre>[구문1] ' or 1=1 limit 0,1# ' or 1=1 limit 1,1#</pre>

		<p>[구문2]</p> <pre>' union select all 1# ' union select null, @@hostname# ' union select null, @@hostname order by 3#</pre> <p>[설명]</p> <p>union은 2개 이상의 select 구문을 실행할 때 사용. 컬럼 개수를 파악해서 해당 컬럼에 SQL 구문을 인젝션하여 정보를 확인</p> <hr/> <p>[구문3]</p> <pre>' union select null,schema_name, 1 from information_schema.schemata# ' union select null,table_name, column_name from information_schema.columns where table_schema='id'#</pre> <p>[설명]</p> <p>첫번째 공격 구문으로 알게된 데이터베이스 이름의 데이터베이스에 포함된 컬럼 이름 모두 노출</p> <hr/> <p>[구문4]</p> <pre>' union select null,concat('name:', name) from user ' union select concat(id,pw) from member# ' union all select group_concat(table_name), 2, 3 from information_schema.TABLES where table_schema=database() // 테이블 추출 ' union select concat(id,pw) from member where NOT id='user'# ' union select all 1,concat(login,password), 3, 4, 5, 6, 7 from users# ' union select 1,concat(id, login),password,email,secret,6,7 from users# 1 union select null,concat(first_name,0x0a,password) from user# '/**/union/**/select/**/pw/**/from/**/member#</pre> <p>[설명]</p> <p>테이블,컬럼,데이터 정보 추출 마우스 우클릭 해서 '페이지 소스' 확인 해서 id, password, email 정보 확인 공백이 막혀있을 경우를 대비하여 우회 공격</p>
--	--	---

Snort rule
alert tcp any any → any any (msg:"[SQL Injection Test] '1'='1 query detect Possible"; content:"GET"; http_method; content:"sqli/"; http_uri; content:" 27 1 27 = 271"; http_uri; flow:to_server,established; sid:4000008; rev:1;)
<p>[설명]</p> <p>정상적인 데이터 뒤에 or '1'='1 형태의 SQL 쿼리를 삽입해 참/거짓 논리 결과를 웹 페이지를 통해 확인하는 패턴은 가장 많이 사용하는 SQL Injection 공격시도기에 탐지조건 설정한다.</p>
alert tcp any any → any any (msg:[SQL Injection Test] union select detect Possible"; flow:to_server,established; content:"GET";http_method; content:"sqli/"; http_uri; pcre:"/union\s+(all\s+)?select\s+UI"; sid:4000009; rev:1;)
<p>[설명]</p> <p>union select 구문 탐지를 위해 정규표현식 사용한다. union all select 구문에 대해서 대소문자 구별없이 탐지한다.</p>
alert tcp any any → any any (msg:"[SQL Injecton Test] information_schema detect Possible"; flow:to_server,established; content:"GET"; http_method; content:"sqli/"; http_uri; content:"information_schema"; nocase; sid:4000010; rev:1;)
<p>[설명]</p> <p>information_schema를 직접 호출하는 행위는 데이터베이스를 확인하기 위해 SQL Injection 공격을 시도하는 행위로 대소문자 구별없이 문자열을 탐지한다.</p>
alert tcp any any → any any (msg:"[SQL Injection Test] or 1=1 query pattern detect Possible"; content:"POST"; http_method; content:"sqli/"; http_uri; pcre:"/or\s+1=1/Pi"; sid:4000011; rev:1;)
<p>[설명]</p> <p>or 1=1 패턴 탐지를 위해 정규표현식 사용한다. or 문자 뒤 공백문자(\s)가 1개 이상 있고 1=1 문자열이 오는 패턴을 탐지한다.</p>
alert tcp any any → any any content:"POST"; http_method; (msg:"[SQL Injection Test] union select concat detect Possible"; flow:to_server,established; content:"sqli/"; http_uri; pcre:"/union\s+(all\s+)?select\s.*concat/Pi"; sid:4000012; rev:1;)
<p>[설명]</p> <p>concat() 함수는 복수 개의 문자열, 칼럼을 합쳐서 보여주는 기능으로 데이터 확인을 하기 위한 용도로 공격구문에 사용되기에 탐지한다.</p>
alert tcp any any → any any (msg:"[SQL Injection Test] union select null query pattern detect Possible"; flow:to_server,established; content:"POST"; http_method; content:"sqli/"; http_uri; pcre:"/union.*select.*(null)?.*i"; sid:4000014 rev:1;)
<p>[설명]</p> <p>컬럼 개수를 알아내기 위해 union select null, null, null 등 null의 개수를 늘려가며 사용한다.</p>

union select 사이 문자열이 0개 이상 올 수 있는 조건을 추가해 탐지한다.

4.4 XSS

가. reflected XSS

공격명칭 (중분류)	공격명칭(소분류)	공격 구문
XSS	reflected XSS	<pre>><script>alert(document.domain)</script> " '\ ",script>alert(1);</script> </div><script>alert(1);</script><div> %3e<script>alert(document.domain)</script></pre>

Snort rule
<pre>alert tcp any any -> any any (msg:"XSS Injection POST Possible"; flow:to_server,established; content:"%3c%2fscript%3e"; nocase; http_client_body; sid:5000002; rev:1;)</pre>
<pre>alert tcp any any -> any any (msg:"XSS Injection URI Possible"; flow:to_server,established; content:"</script>"; nocase; http_uri; sid:5000001; rev:1;)</pre>

나. SVG XSS(Stored XSS using SVG file)

공격명칭 (중분류)	공격명칭(소분류)	공격 구문
XSS	SVG XSS	<p>악성 페이로드가 포함된 xss.svg 파일을 생성한 후 이미지 업로드한다.</p> <pre><?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg version="1.1" baseProfile="full" > <polygon id="triangle" points="0,0 0,50 50,0" fill="#009900" stroke="#004400"/> <script type="text/javascript"> alert("SVG XSS"); </script> </svg></pre>

Snort rule
alert tcp any any -> any any (msg:"SVG XSS Possible"; flow:to_server,established; pcre:"/<svg.+>/i"; sid:5000003; rev:1;)
<p>[설명]</p> <p>svg의 모든 것(svg, svga~z, svgA~Z 등), +의 앞이 1글자 이상오는 경우(svg, svgg, svgggg...), 대소문자 구분 없이 검색(svg, SVG) 하는 경우를 모두 탐지한다.</p>

다. Stored XSS

공격명칭 (중분류)	공격명칭(소분류)	공격 구문
XSS	Stored XSS	<pre>"><script>alert(1);</script> '>'><script[^>]*>[^<]*</script> <div><script>alert(1);</script></div> </p><script>alert(1);</script><p> </h2><script>alert(1);</script><h2> javascript.alert(1)</pre>

Snort rule
alert tcp any any -> any any (msg:"Stored XSS Possible"; content:"/j_security_check?loginbutton"; nocase; pcre:"/j_username\x3d%27%3E%5C%22%3E%3Cscript[^>]*%3E[^\<]*%3C%2Fscript%3E/i"; sid:5000004; rev:1;)
alert tcp any any -> any any (msg:"XSS Possible"; flow:to_server,established; content:"javascript:"; nocase; http_uri; sid:5000005; rev:1;)

5. 통합보안관제시스템 구축

5.1 통합보안관제시스템(SIEM)을 위한 Splunk

스위치 허브를 SOC(Security Operations Center)로 사용해 Splunk를 설치하여 통합보안관제시스템을 운영한다. Agent 프로그램인 Splunk Forwarder를 사용해 실시간으로 Splunk Server에 로그를 보낸다.



[그림31] Splunk - 스위치 허브

5.1.1 목적

- 가. 3계층 Client Tier, Application Tier, Data Tier으로 나누어 3 Tier-Architecture로 구축한 각 서버의 환경(CPU, Memory, Disk 등)을 실시간으로 전송받아 관리
- 나. 서버의 Access.log, Error.log 등을 전송받아 이상 징후 탐지 및 분석
- 다. pfSense로 구축한 Firewall, IDS, IPS 등 로그와 공격 탐지시 발생하는 Snort Rule 알람을 통해 빠른 차단
- 라. 인프라 내 SOC(Security Operations Center)에서 전체 시스템 모니터링 및 통제

5.1.2 환경 구성

IDS/IPS로 사용하는 pfSense는 로그를 자체 전송한다.

장비	상세정보	설치된 서버
Splunk	Splunk-9.1.3-Linux-x86_64	SOC
		WEB

Forwarder	Splunkforwarder-9.2.0.1-Linux-2.6	WAS DB
-----------	-----------------------------------	-----------

5.1.3 Splunk 기본 설정

[그림32] Splunk 기본 설정

가. UDP 514 포트에 데이터 수신 설정을 통해 방화벽,IDS,IPS로 사용하는 pfSense 로부터 데이터 수신을 받는다.

나. pfSense, OpenWRT 자체에 보관할 수 있는 로그 크기는 한정적이기에 Splunk Syslog 서버에 복사해서 로그 크기에 한계없이 장기적인 모니터링 가능하도록 설정한다.

다. WEB/WAS/DB 서버에 설치한 Splunk Forwarder를 통해 해당 경로(/var/log/webLog)에 있는 로그 파일(access.log, error.log)를 Splunk 서버로 Forwarder하도록 /opt/splunkforwarder/etc/system/local에 다음과 같이 작성한다.

```
root@web: /opt/splunkforwarder/etc/system/local

[default]
host = 40.0.0.10

[monitor:///var/log/webLog/]
disabled = false
index = web
sourcetype = web_log

"inputs.conf" 7 lines, 107 bytes
```

[그림33] Splunk Forwarder

라. 연결 확인 및 Output.conf 파일 확인

```
[tcpout]
defaultGroup = default-autolb-group

[tcpout:default-autolb-group]
server = 40.0.0.10:9997

[tcpout-server:///40.0.0.10:9997]
```

[그림34] Splunk Forwarder

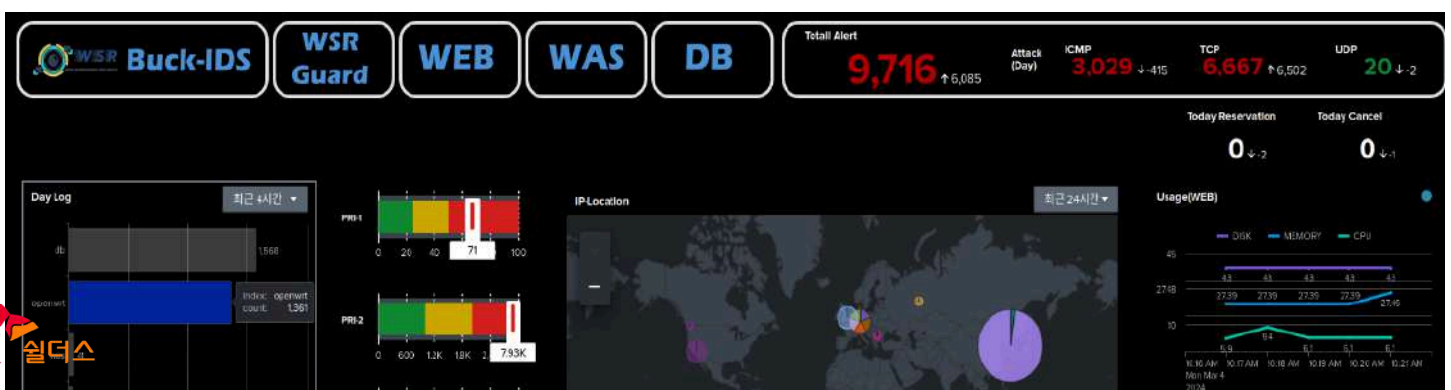
```
web@web:~$
web@web:~$ netstat -ano | grep 9997
tcp        0      0 10.0.0.10:48448    40.0.0.10:9997    ESTABLISHED off (0.00/0/0)
```

[그림35] Splunk Forwarder

5.2 보안관제 고도화를 위한 Splunk Dashboard

5.2.1 전체 Dashboard

Splunk Dashboard Studio를 사용해서 시각화를 정렬할 수 있다. SOC 서버에서 모니터링 할 대상들을 Splunk Processing Language(SPL)을 사용해 다음과 같이 대시보드를 구성한다.



[그림36] Splunk DashBoard

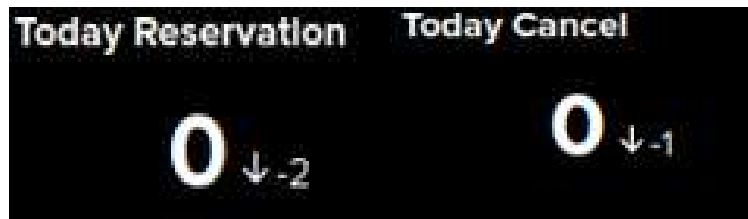
가.



[그림37] Main Alert

SPL
index=openwrt source= "udp:514" Snort=snort timechart span=1d count by Snort
index=openwrt source= "udp:514" snort stats count by Protocol, _time timechart span=1d sum(count) as TotalCount by Protocol

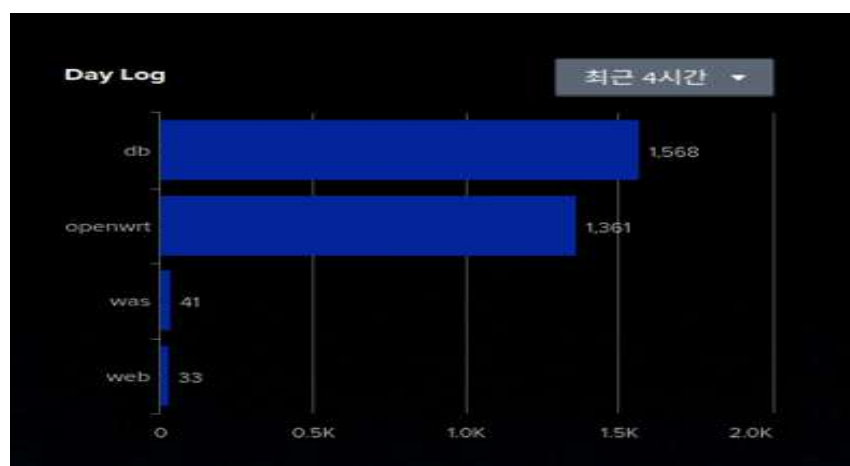
나.



[그림38] Today Reservation

SPL
index=web source= "/var/log/webLog/access.log" Request= "POST /reservation /reservationCom.re HTTP/1.1" Response= " http://211.58.82.226/reservation.re " timechart span=1d count eval count=ifnull(count,0)

다.



[그림39] Day Log

SPL
index=web source= “/var/log/webLog/per.txt” OR (index=was source!=/var/log/wasLog/per.txt) OR (index=db source!=/var/log/dbLog/per.txt) OR (index=openwrt source= “udp:514”) stats count by index

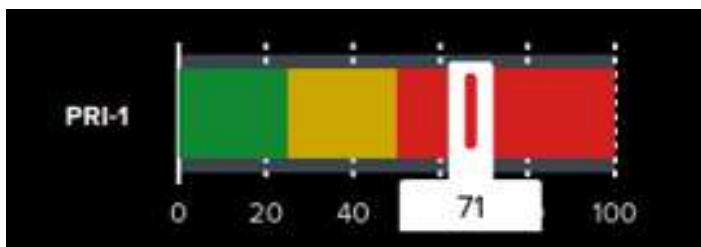
라.



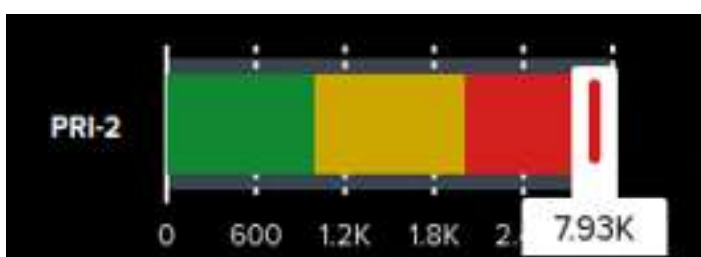
[그림40] Week Alert

SPL
index=openwrt Snort= “snort” NOT (Start_ip=211.58.82.226 OR Start_ip=8.8.8.8) earliest=-1w latest=now timechart span=1d count by index

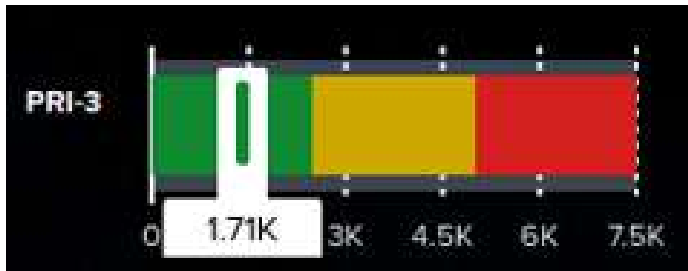
마.



[그림40] PRI-1 위험도



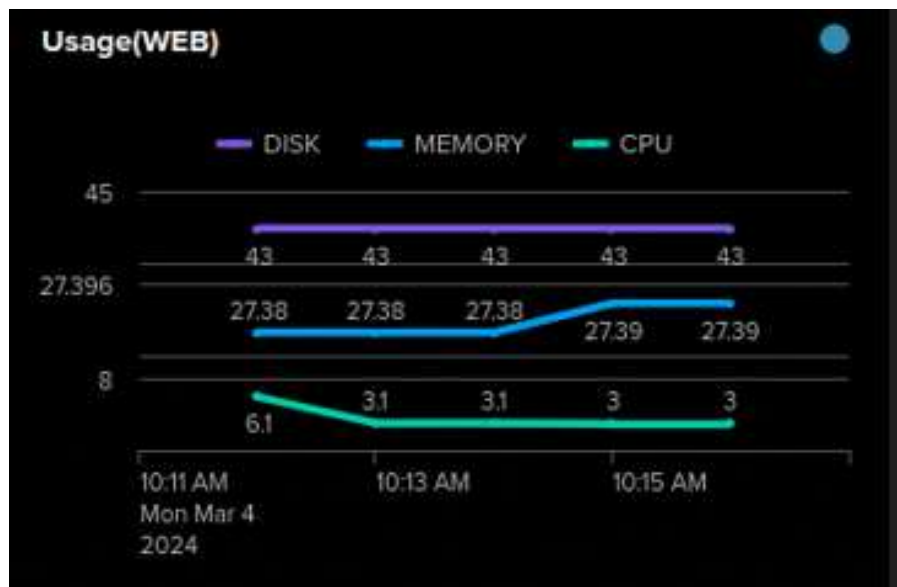
[그림41] PRI-2 위험도



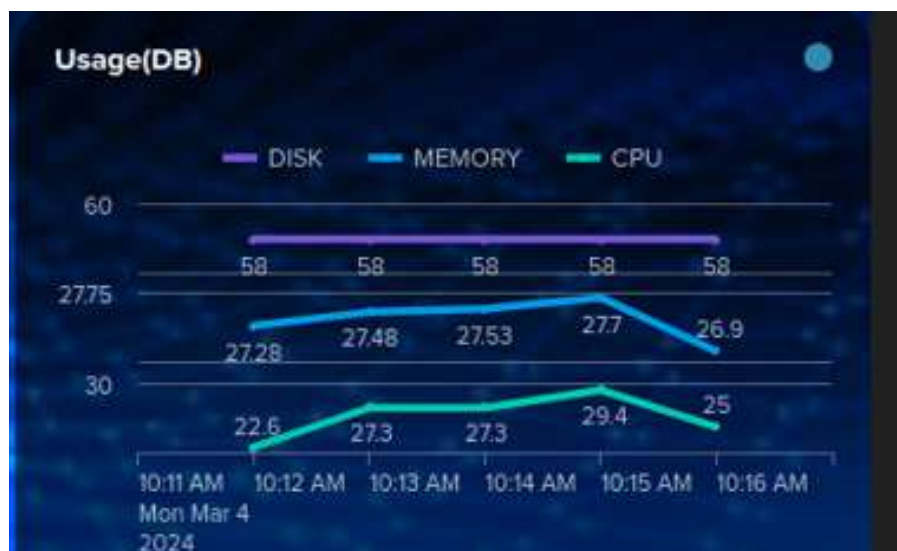
[그림42] PRI-3 위험도

SPL
index=openwrt source= "udp:514" snort Pri=(Pri 위험도) stats count

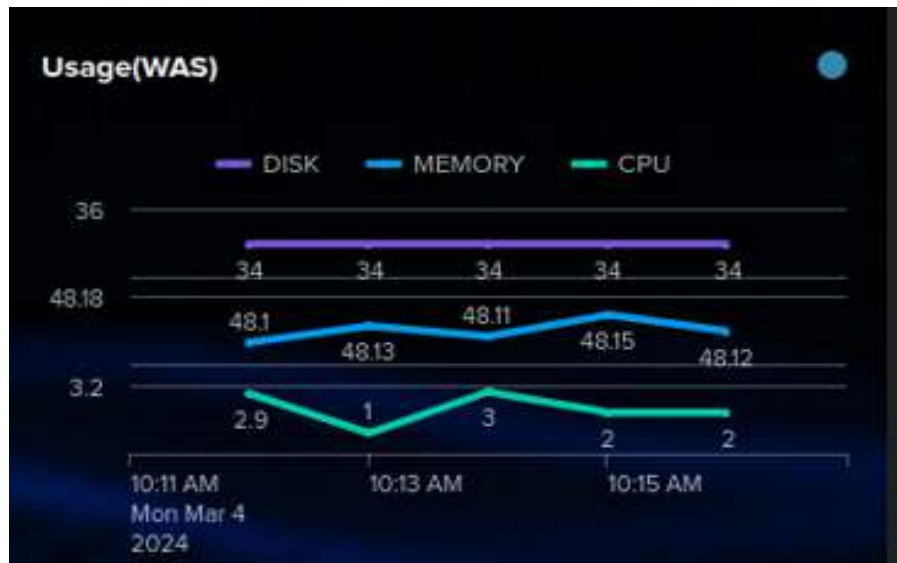
바.



[그림43] Usage(WEB)



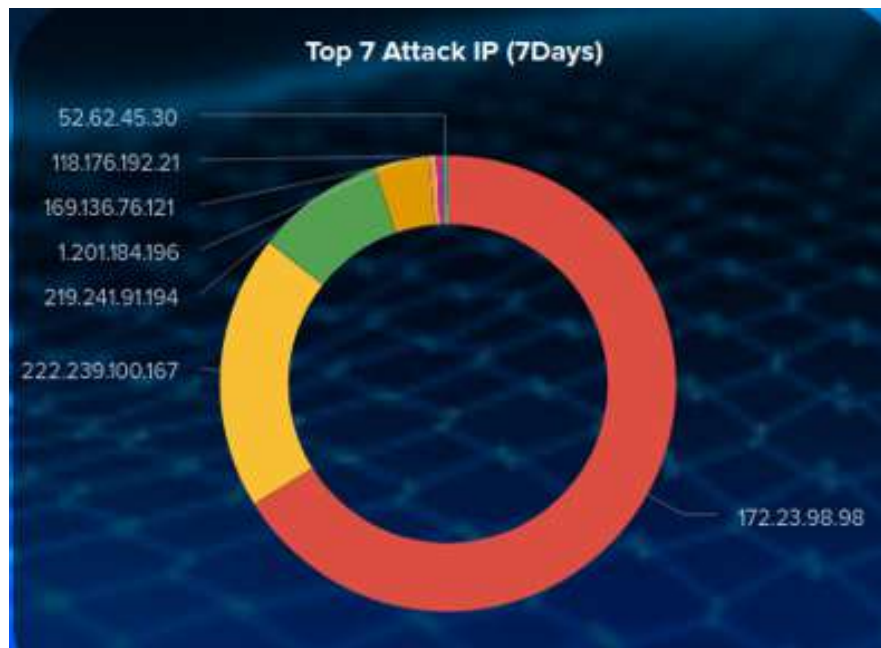
[그림44] Usage(DB)



[그림45] Usage(WAS)

SPL
<pre>index=web source= "/var/log/[구간로그]/pert.txt" timechart spna=1m lateset(DISK_Per) as DISK latest(MEMORY_Per) as MEMORY latest(CPU_Per) as CPU</pre>

사.



[그림46]Top Attack IP(7Days)

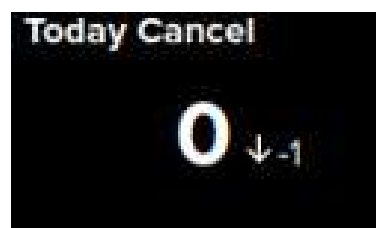
SPL
index=openwrt Snort= “snort” NOT(Start_ip=211.58.82.226 OR Start_ip=8.8.8.8) stat count by Start_ip sort - count head 7



[그림47]Top 5 Attack Event

SPL
index=openwrt Snort= “snort” NOT(Start_ip=211.58.82.226 OR Start_ip=8.8.8.8) stats count by MSG sort -count head 5

아.



[그림48]Today Cancel

SPL
index=web source=” /var/log/webLog/access.log” Request= “POST /reservation/reservationCom.re HTTP/1.1” Response= “ http://211.58.82.226/reservation/reservation.re ” timechart span=1d count eval count=ifnull(count,0)

자.



[그림49]Find link

SPL

```
index=openwrt Snort= "snort| table Pri, MSG, Start_ip, Dest_ip| dedup Start_ip, Dest_ip| sort -Pri
```

차.

Snort Alert				
Time ↕	MSG ↕	Pr ↕	Start_ip ↕	Dest_ip ↕
Mar 4 10:20:29	"PROTOCOL-ICMP PING"	3	172.23.98.98	211.58.82.226
Mar 4 10:17:40	"PROTOCOL-ICMP PING"	3	172.23.98.98	211.58.82.226
Mar 4 10:17:16	"PROTOCOL-ICMP PING"	3	172.23.98.98	211.58.82.226
Mar 4 10:16:49	"PROTOCOL-ICMP PING"	3	172.23.98.98	211.58.82.226
Mar 4 10:16:12	"PROTOCOL-ICMP PING"	3	172.23.98.98	211.58.82.226
Mar 4 10:15:30	"PROTOCOL-ICMP PING"	3	172.23.98.98	211.58.82.226
Mar 4 10:12:37	"PROTOCOL-ICMP PING"	3	172.23.98.98	211.58.82.226
Mar 4 10:12:12	"PROTOCOL-ICMP PING"	3	172.23.98.98	211.58.82.226

[그림50] Snort Alert

SPL

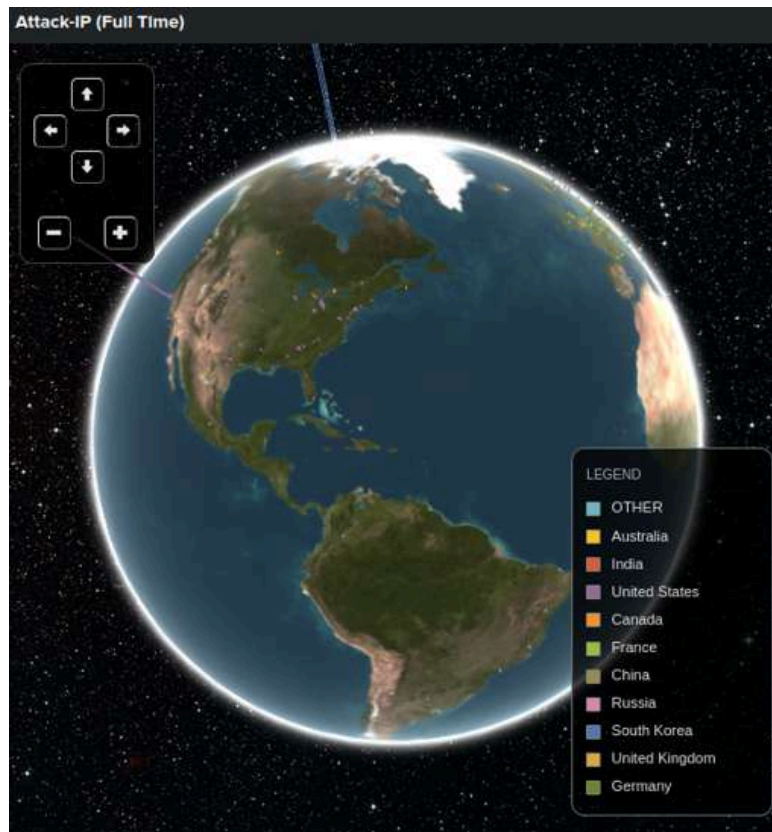
```
index=openwrt Snort= "snort| table time, MSG, Pri, Start_ip, Dest_ip| sort -Time
```


5.2.2 WSR Guard DashBoard

WSR 보안관제팀의 WSR_GUARD Field 지정은 다음과 같다.

Mar 4 02:25:30 40.0.0.1 Mar 3 17:25:30 OpenWrt snort: [1:500031:1] "XSS Injection URI Possible" [Classification: Web Application Attack] [Priority: 1] {TCP} 128.134.38.141:61147 -> 211.58.82.226:80								
Time	Snort	GID_SID	MSG	ClassType	Pri	Protoco l	Start_ip	Dest_ip
Mar 3 17:25:30	snort	1:500031: 1	XSS Injection URI Possible	Classification : Web Application Attack	1	TCP	128.134.38.14 1	211.58.82.22 6

가.



[그림51] Attack-IP Full time

SPL
index=openwrt snort where Start_ip=" 211.58.82.226" iplocation Start_ip geo stats count by Country latfield=lat logfile=lon

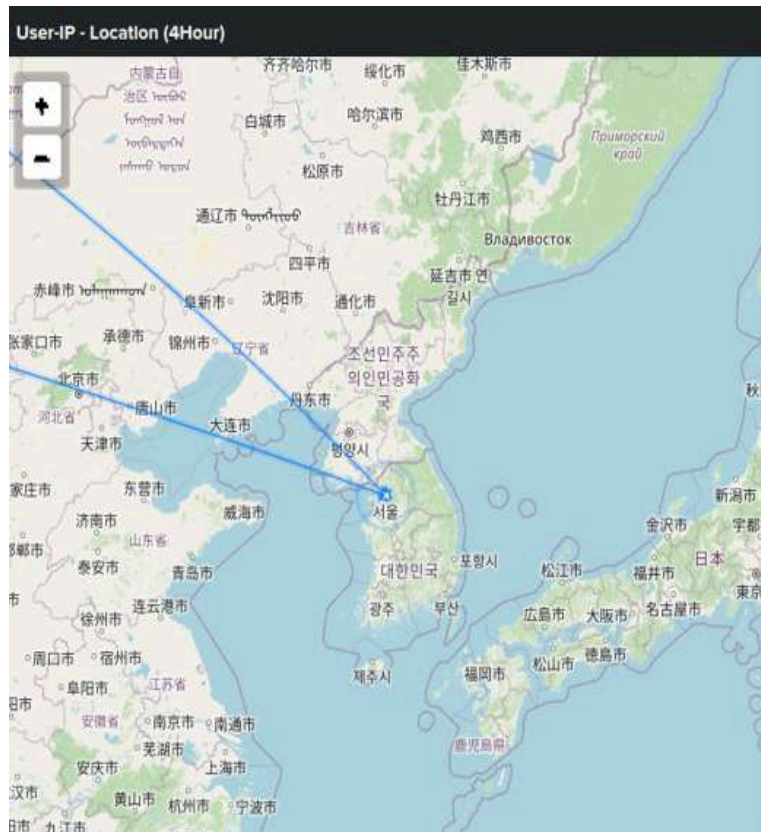
나.



[그림52] Attack-IP Location

SPL
<pre> index=openwrt snort where Start_ip!="211.58.82.226" iplocation Start_ip rename lat as start_lat, lon as start_lon appendpipe [iplocation Dest_ip rename lat as end_lat, lon as end_lon] stats count by Country, start_lat, start_lon, end_lat, end_lon eval start_lat, start_lon, color="#FF2121", animate="true", Weight=0.1, pulse_a </pre>

다.

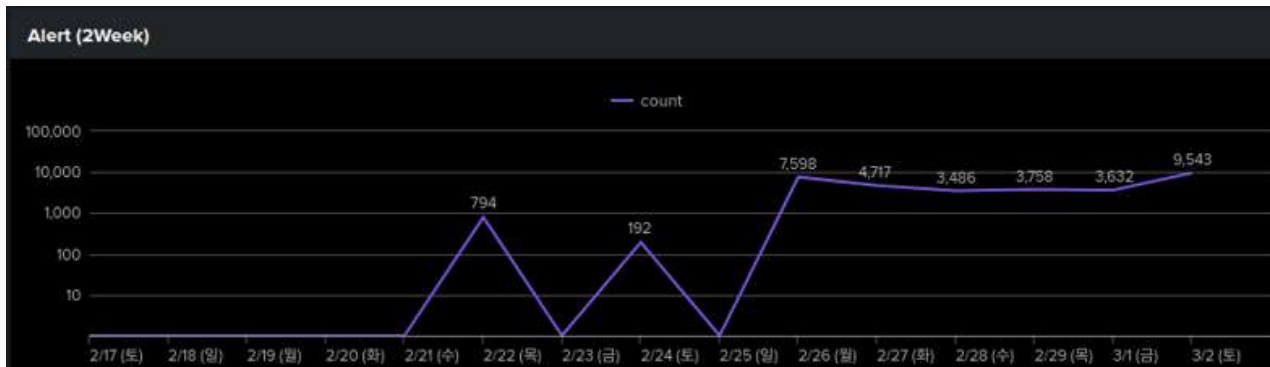


[그림53] User-IP Location

SPL

```
index=web Request="GET / HTTP/1.1" Response="-"
| iplocation Client_ip
| rename lat as start_lat, lon as start_lon
| appendpipe [
| makeresults
| eval destination_ip = "211.58.82.226"
| iplocation destination_ip
| rename lat as end_lat, lon as end_lon
]
```

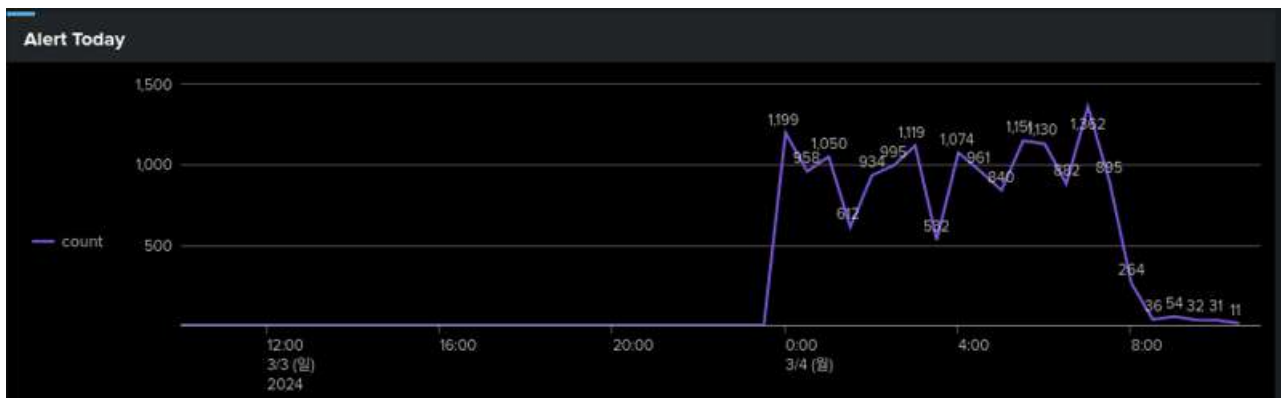
라.



[그림54] Alert 2Week

SPL

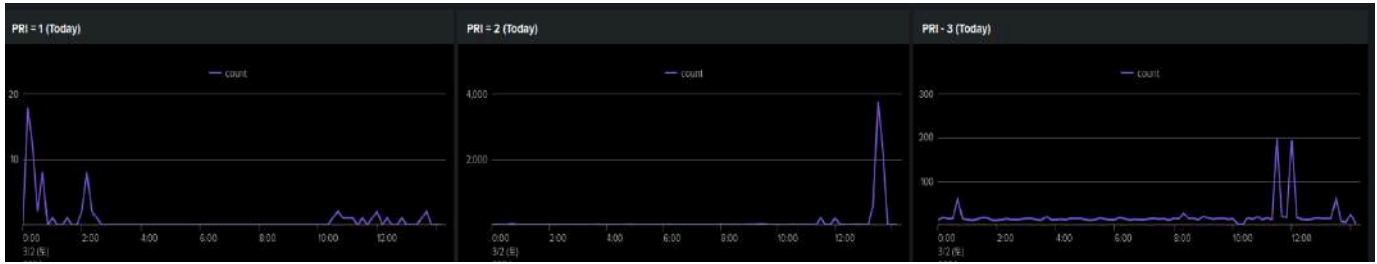
```
index=openwrt snort earliest=-2w latest=now | timechart span=1d count
```



[그림55] Alert Today

SPL

```
index=openwrt snort Start_ip!="211.58.82.226"
| where _time >= relative_time(now(), "@d")
| timechart count
```



[그림56] Pri Today

SPL
index=openwrt Snort="snort" Pri={위험도} timechart count

마.

_time	Start_ip	Dest_ip	Protocol	Pri	MSG	GID_SID
2024/03/02 14:11:53	172.23.98.98	211.58.82.226	ICMP	3	"PROTOCOL-ICMP PING"	1:384:8
2024/03/02 14:11:14	172.23.98.98	211.58.82.226	ICMP	3	"PROTOCOL-ICMP PING"	1:384:8
2024/03/02 14:10:30	172.23.98.98	211.58.82.226	ICMP	3	"PROTOCOL-ICMP PING"	1:384:8
2024/03/02 14:09:35	128.9.29.131	211.58.82.226	ICMP	3	"PROTOCOL-ICMP PING"	1:384:8
2024/03/02 14:07:46	172.23.98.98	211.58.82.226	ICMP	3	"PROTOCOL-ICMP PING"	1:384:8
2024/03/02 14:07:20	172.23.98.98	211.58.82.226	ICMP	3	"PROTOCOL-ICMP PING"	1:384:8
2024/03/02 14:06:55	172.23.98.98	211.58.82.226	ICMP	3	"PROTOCOL-ICMP PING"	1:384:8
2024/03/02 14:06:46	83.211.89.85	211.58.82.226	ICMP	3	"PROTOCOL-ICMP PING"	1:384:8

[그림56] Snort Table

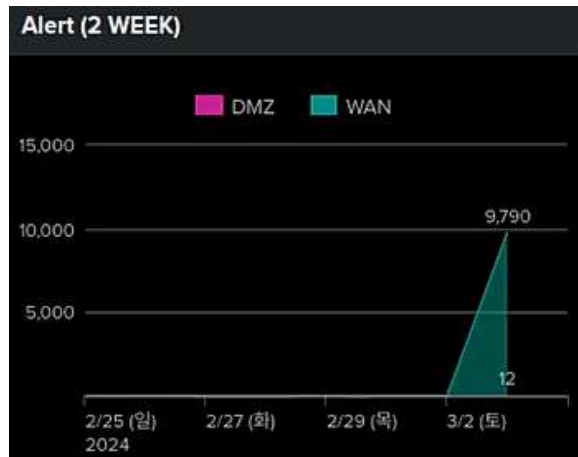
SPL
index=openwrt snort table _time, Start_ip, Dest_ip, Protocol, Pri, MSG, GID_SID sort -_time

5.2.3 WSR Guard 1

방화벽,IDS,IPS로 사용하는 pfSense Snort Rule Msg에 구간명을 추가한다.

Mar 3 20:41:35 10.0.10.1 1 2024-03-03T20:41:35.813637+09:00 pfSense.home.arpa snort 13797 -- [1:100002:0] SYN Snort Attack Possible WAN [Classification: Attempted Denial of Service] [Priority: 2] {TCP} 185.63.42.74:3198 -> 10.0.0.254:0								
Time	Snort	SID	Msg	ClassType	Pri	Protocol	Start_ip	Dest_ip
Mar 3 20:41:35	pfSense.home.arpa snort	1:100002:0	SYN Snort Attack Possible WAN	Classification: Web Application Attack	2	TCP	185.63.42.74	10.0.0.254

가.

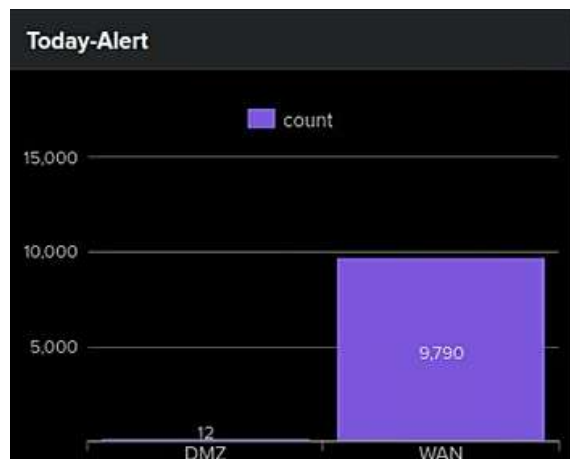


[그림57]Alert 2Week

SPL

```
index=pfsense1 (Msg="*WAN" OR Msg="*DMZ")
| eval source_a=case(match(Msg, ".*DMZ.*"), "DMZ", match(Msg, ".*WAN.*"), "WAN",
true(), Msg)
| stats count by source_a, _time
| timechart span=1d sum(count) as TotalCount by source_a
| fillnull value=0
```

나.

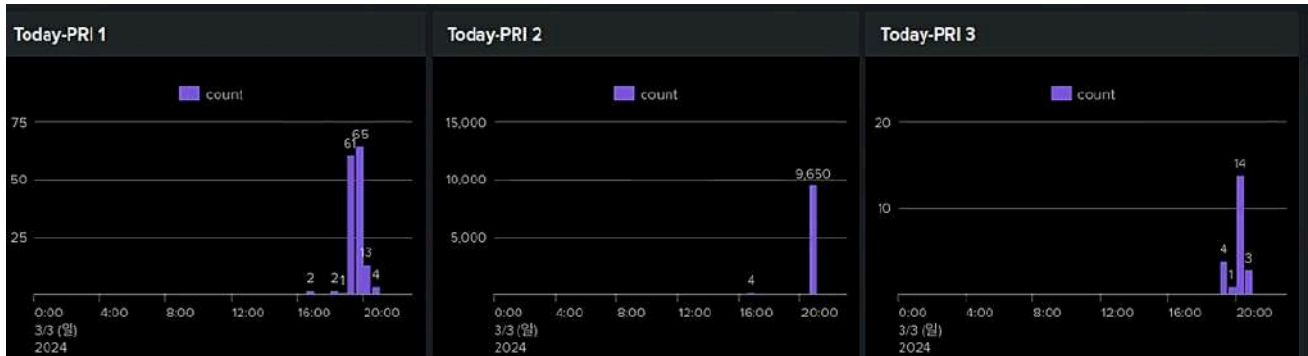


[그림58]Alert 2Week

SPL

```
index=pfsense1 Msg="*WAN" OR Msg="*DMZ"
| eval source_a=case(match(Msg, ".*DMZ.*"), "DMZ", match(Msg, ".*WAN.*"), "WAN",
true(), Msg)
| stats count by source_a
| fillnull value=0
```

다.



[그림59]PRI1, PRI2, PRI3

SPL

```
index=pfsense1 Msg="*WAN" OR Msg="*DMZ" Pri="{위험도}" | timechart count
```

WAN - LOG								
Time #	Start_ip #	Start_port #	Des_ip #	Des_port #	Protocol #	Sid #	Msg #	Pri #
Mar 3 20:41:35	185.63.42.34	3168	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2
Mar 3 20:41:35	175.190.191.76	3200	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2
Mar 3 20:41:35	72.138.141.95	3122	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2
Mar 3 20:41:35	135.22.120.22	3118	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2
Mar 3 20:41:35	97.145.47.181	3078	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2
Mar 3 20:41:35	251.189.190.34	3092	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2
Mar 3 20:41:35	110.81.120.182	3058	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2
Mar 3 20:41:35	102.55.64.136	3038	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2
Mar 3 20:41:35	245.38.239.226	3049	10.0.0.254	0	TCP	1100002:0	SYN Short Attack Possible WAN	2

[그림60]WAN Log

SPL

```
index=pfsense1 Msg="*WAN"
| sort - Time
| table Time, Start_ip, Start_port, Des_ip, Des_port, Protocol, Sid, Msg, Pri
```

<div>Row-Data</div>					
i	시간	이벤트			
>	24/03/03 20:54:49.000	<div>\x040\xA1\x00\x00\x00\x00\x00\x00\x00\x00\x00\xE4eJ\u29\x8BJ\u8B\u80\u80\u80'ja\u8D)\xD5\u8E\u80<\xC8\u80=\xAF\uDF</div> <div>\x0d</div> <div>\x00\xeFA</div> <div>\xE3\uA0\u99\u80\uC2\uCA\u80\u80\u80\u80\uFA\uF0\u80\u80\b4</div> <div>r\uB1\uXf\u80\u80\u80\u80e\uE4e\u97\u99\u80J\u80\u80\u80\u80\u80\uDD\uD5'\u8j\u8E\u80\u80<\u80\u80ei\u80?]</div> <div>99개 행 모두 표시</div> <div>host = 10.0.150.10 source = /var/log/wasLog/test3.pcap sourcetype = was_log</div>			
>	24/03/03 20:54:47.000	<div></div></div> <div><table></div> <div><tr></div> <div><th><번호</th></div> <div>69개 행 모두 표시</div> <div>host = 10.0.150.10 source = /var/log/wasLog/test3.pcap sourcetype = was_log</div>			
>	24/03/03 20:54:47.000	<div>Date: Sun, 03 Mar 2024 11:54:47 GMT</div> <div>Connection: close</div> <div><!DOCTYPE html></div> <div><html></div> <div><head></div> <div>257개 행 모두 표시</div> <div>host = 10.0.150.10 source = /var/log/wasLog/test3.pcap sourcetype = was_log</div>			

SPL
index=was source="/var/log/wasLog/Rawdata*.pcap"

Alert (2 Week)

The chart displays the number of alerts over a two-week period. The x-axis represents dates from 2/25 (일) to 3/2 (토). The y-axis represents the number of alerts, with markers at 20 and 40. Two data series are shown: DB (blue) and WAS (orange). Both series show zero alerts from 2/25 to 2/29. On 3/2 (토), DB alerts increase to 18 and WAS alerts increase to 36.

날짜	DB	WAS
2/25 (일)	0	0
2/26 (월)	0	0
2/27 (화)	0	0
2/28 (수)	0	0
2/29 (목)	0	0
3/2 (토)	18	36

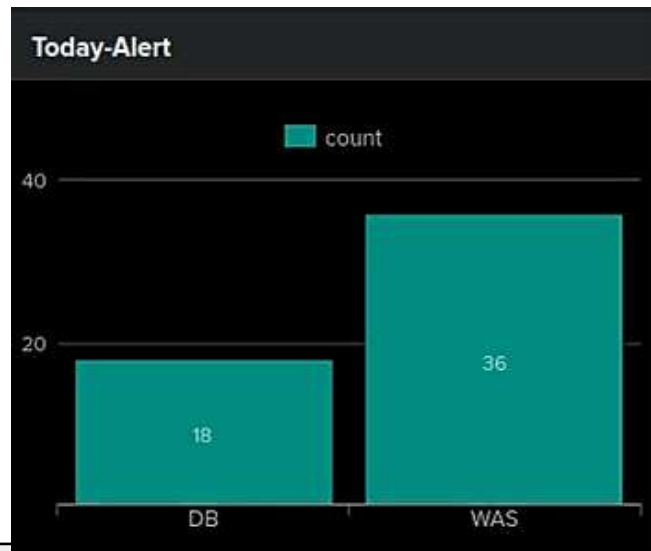
```

SPL

index=pfsense2 Msg="*WAS" OR Msg="*DB"
| eval source_a=case(match(Msg, ".*WAS.*"), "WAS", match(Msg, ".*DB.*"), "DB", true(),
Msg)
| stats count by source_a, _time
| timechart span=1d sum(count) as TotalCount by source_a
| fillnull value=0

```

바.



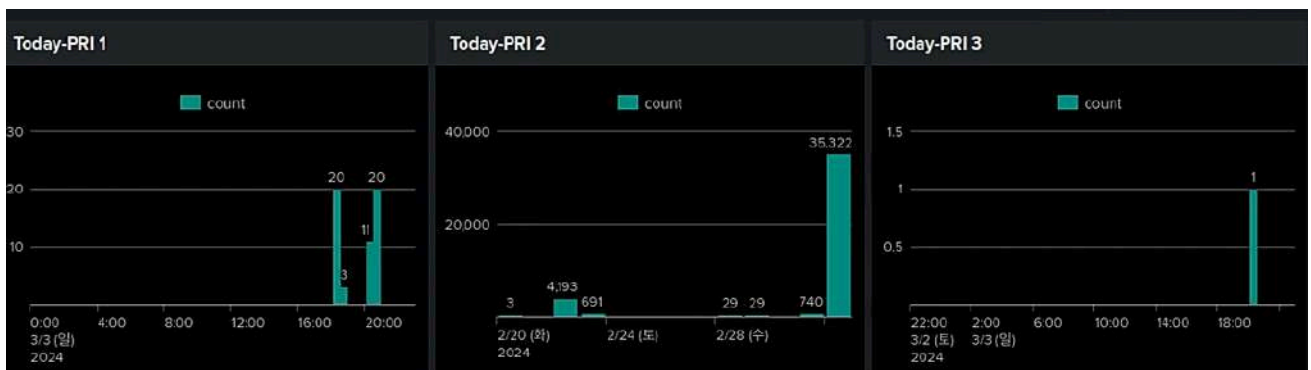
[그림62]WAS,DB

Alert(2Week)

SPL

```
index=pfsense2 Msg="*WAS" OR Msg="*DB"
| eval source_a=case(match(Msg, ".*WAS.*"), "WAS", match(Msg, ".*DB.*"), "DB", true(),
Msg)
| stats count by source_a
| fillnull value=0
```

사.



[그림63]PRI 1,2,3

SPL

```
index=pfsense2 Msg="*WAS" OR Msg="*DB" Pri=" {위험도}" | timechart count
```


아.

WAS - LOG								
Time #	Start_ip #	Start_port #	Des_ip #	Des_port #	Protocol #	Sid #	Msg #	Pri #
Mar 3 20:34:49	10.0.100.10	58272	10.0.250.10	8080	TCP	1:400031:0	SQL Injection Union select Possible WAS	1
Mar 3 20:35:50	10.0.100.10	39402	10.0.250.10	8080	TCP	1:500009:0	File Upload pdf Possible WAS	1
Mar 3 20:35:25	10.0.250.10	46212	10.0.200.10	1521	TCP	1:500016:0	File Upload Possible WAS	1
Mar 3 20:34:06	10.0.100.10	57710	10.0.250.10	8080	TCP	1:400031:0	SQL Injection Union select Possible WAS	1
Mar 3 20:34:01	10.0.100.10	49374	10.0.250.10	8080	TCP	1:400031:0	SQL Injection Union select Possible WAS	1
Mar 3 20:34:01	10.0.100.10	49374	10.0.250.10	8080	TCP	1:400032:0	SQL Injection Union all select Possible WAS	1
Mar 3 20:33:45	10.0.100.10	40668	10.0.250.10	8080	TCP	1:400031:0	SQL Injection Union select Possible WAS	1
Mar 3 20:33:43	10.0.100.10	40668	10.0.250.10	8080	TCP	1:400032:0	SQL Injection Union all select Possible WAS	1

[그림64]WAS Log

SPL
<pre> index=pfsense2 Msg="*WAS" sort - Time table Time, Start_ip, Start_port, Des_ip, Des_port, Protocol, Sid, Msg, Pri </pre>

자.

DB - LOG								
Time #	Start_ip #	Start_port #	Des_ip #	Des_port #	Protocol #	Sid #	Msg #	Pri #
Mar 3 20:40:23	10.0.250.10	46212	10.0.200.10	1521	TCP	1:500002:0	Stored XSS Attempt Possible DB	1
Mar 3 20:40:06	10.0.250.10	46212	10.0.200.10	1521	TCP	1:500002:0	Stored XSS Attempt Possible DB	1
Mar 3 20:39:58	10.0.250.10	46212	10.0.200.10	1521	TCP	1:500002:0	Stored XSS Attempt Possible DB	1
Mar 3 20:38:07	10.0.250.10	46212	10.0.200.10	1521	TCP	1:500002:0	Stored XSS Attempt Possible DB	1
Mar 3 10:33:27	10.0.250.10	47412	10.0.200.10	1521	TCP	1:500002:0	Stored XSS Attempt Possible DB	1
Mar 3 10:29:35	10.0.250.10	47412	10.0.200.10	1521	TCP	1:500002:0	Stored XSS Attempt Possible DB	1
Mar 3 10:28:41	10.0.250.10	47412	10.0.200.10	1521	TCP	1:500002:0	Stored XSS Attempt Possible DB	1
Mar 3 10:28:25	10.0.250.10	47412	10.0.200.10	1521	TCP	1:500002:0	Stored XSS Attempt Possible DB	1

[그림65]DB Log

SPL
<pre> index=pfsense2 Msg="*DB" sort - Time table Time, Start_ip, Start_port, Des_ip, Des_port, Protocol, Sid, Msg, Pri -count </pre>

차.

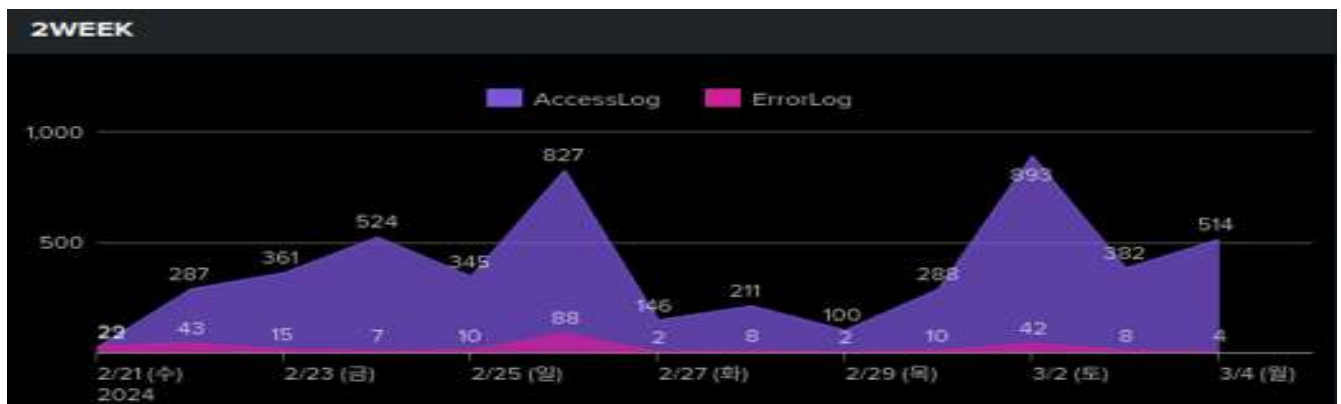
Row-Data		
i	시간	이벤트
>	24/03/03 20:54:49.000	<pre>\xD4\xA1\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xE4e7J\x99\x00J\x00\x00J\x00\x00\x00'ja\x00'\xCD\xD5\x0E\x00<\xC8\x00-\xAF\xDF \x0d \x00\xFA \xE3\xA0\x90\x0C2\xCA\x00\x00\x00\xA0\xFA\xF0\x00\x69\xB4 r\xB1\x91\FD\x00\x00\x00\xE4e7J\x99\x00J\x00\x00\x00\x00\x00\x00\xCD\xD5\x00'ja\x00E\x00<\x00\x00\x00fj 99개 행 모두 표시 host = 10.0.150.10 source = /var/log/wasLog/test3.pcap sourcetype = was_log</pre>
>	24/03/03 20:54:47.000	<pre></div> <table> <tr> <th>번호</th> 69개 행 모두 표시 host = 10.0.150.10 source = /var/log/wasLog/test3.pcap sourcetype = was_log</pre>
>	24/03/03 20:54:47.000	<pre>Date: Sun, 03 Mar 2024 11:54:47 GMT Connection: close <!DOCTYPE html> <html> <head> 257개 행 모두 표시 host = 10.0.150.10 source = /var/log/wasLog/test3.pcap sourcetype = was_log</pre>

[그림 66] Raw Data

SPL
index=was source="/var/log/wasLog/RawData*.pcap"

5.2.4 WEB Dashboard

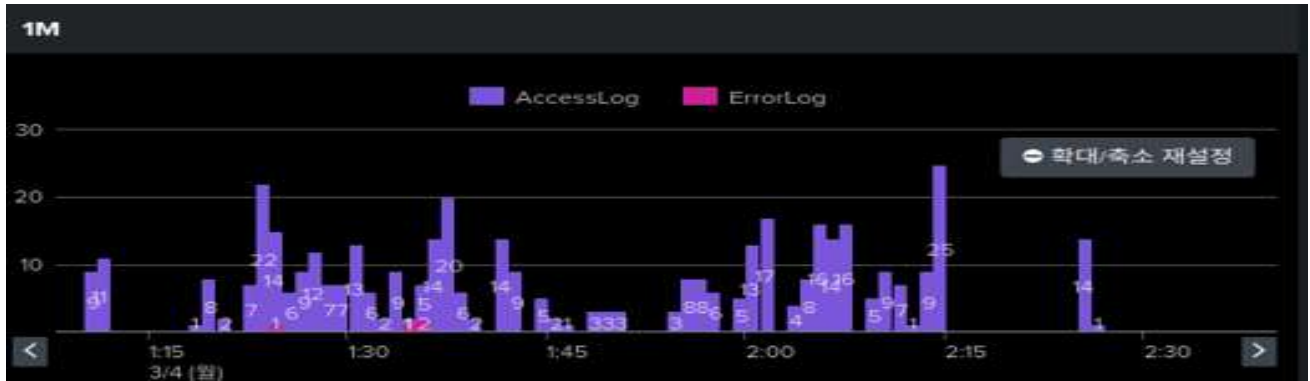
가.



[그림67]WEB Alert(2Week)

SPL
<pre>index=web (source!=" /var/log/webLog/per.txt" AND source!=" /var/log/webLog/cpu_use.txt") latest=now eval source = case(source == " /var/log/webLog/access.log", "AccessLog", source == "/var/log/webLog/error.log", "ErrorLog", true(), source) timechart span=1d count by source</pre>

나.



[그림68]WEB Daylog(1m)



[그림69]WEB Error Status

SPL

```
index=web source=/var/log/webLog/error.log" | stats count by System
```

다.

Access Log				
Time ↕	Client_Ip ↕	Request ↕	Response ↕	Status ↕
[04/Mar/2024:09:52:24]	42.96.15.107	POST / HTTP/1.1	-	200
[04/Mar/2024:09:52:23]	42.96.15.107	GET /.env HTTP/1.1	-	404
[04/Mar/2024:09:07:30]	40.77.167.1	GET /faq/faq.bo HTTP/1.1	-	200
[04/Mar/2024:08:46:11]	188.166.71.161	GET /Pages/log/ HTTP/1.1	-	404
[04/Mar/2024:08:46:11]	188.166.71.161	GET /Temporary_Listen_Addresses HTTP/1.1	-	404

[그림70]WEB Access Log

SPL

```
index=web source="/var/log/webLog/access.log" | sort - Time
```

Error_Log			
Day	Time	System	Content
2024/03/04	01:35:30	[warn]	23#23: *1041 a client request body is buffered to a temporary file /var/cache/nginx/client_temp/0000000015, client: 128.134.38.141, server: localhost, request: "POST /qna/qnaUpdate.bo HTTP/1.1", host: "211.58.82.226", referer: "http://211.58.82.226/qna/qnaEdit.bo?qnaNo=65"
2024/03/04	01:35:11	[error]	23#23: *1036 client intended to send too large body: 1992192 bytes, client: 128.134.38.141, server: localhost, request: "POST /qna/qnaUpdate.bo HTTP/1.1", host: "211.58.82.226", referer: "http://211.58.82.226/qna/qnaEdit.bo?qnaNo=65"
2024/03/04	01:34:53	[error]	23#23: *1035 client intended to send too large body: 1992192 bytes, client: 128.134.38.141, server: localhost, request: "POST /qna/qnaUpdate.bo HTTP/1.1", host: "211.58.82.226", referer: "http://211.58.82.226/qna/qnaEdit.bo?qnaNo=65"

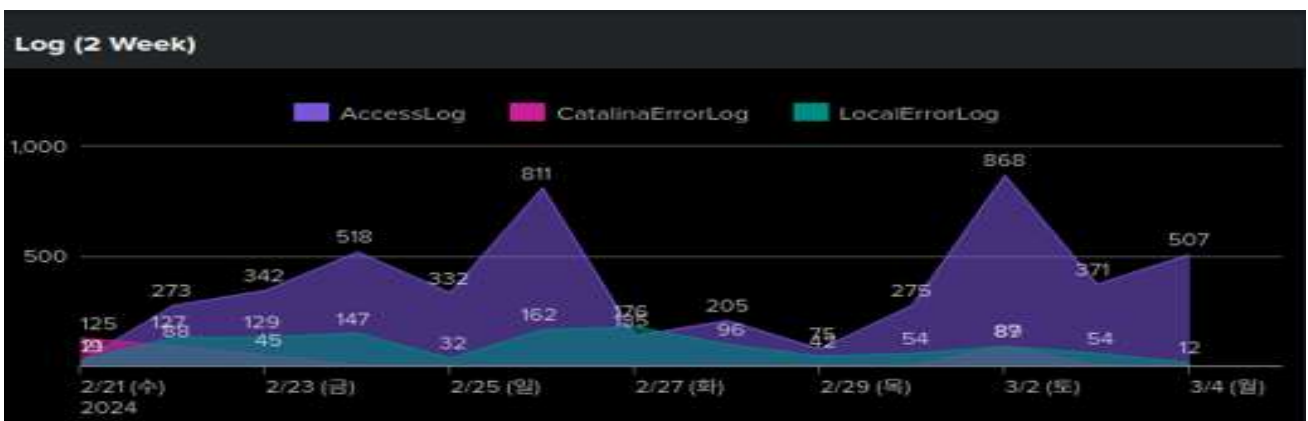
[그림71]WEB Error Log

SPL

```
index=web source="/var/log/webLog/error.log"
| sort - Day, Time
| table Day,Time, System, Content
```

5.2.5 WAS Dashboard

가.



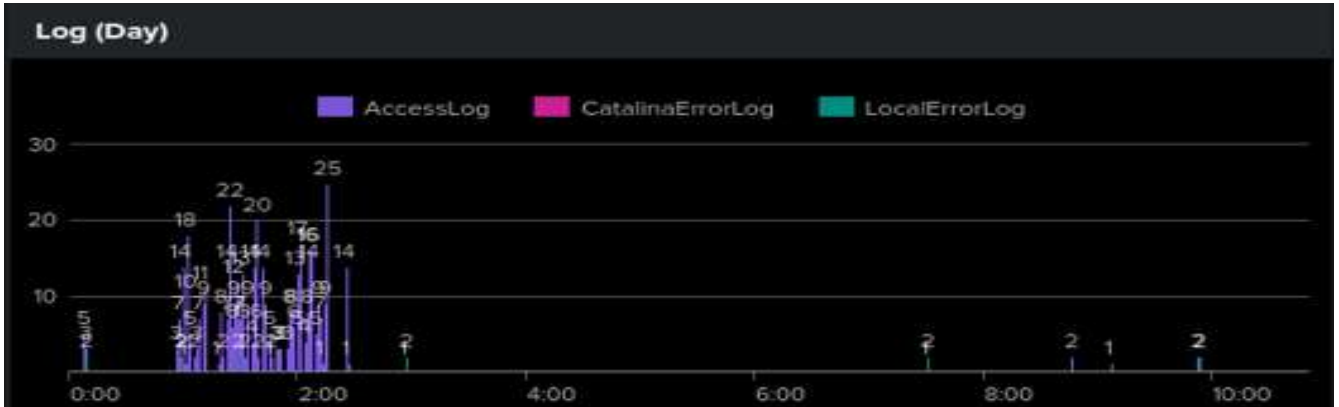
[그림72] WAS Log(2Week)

SPL

```
index=was (source!="var/log/wasLog/per.txt" AND source!="var/log/wasLog/*.pcap" AND
source !="/var/log/syslog") latest=now
| eval source_alias = case(
    match(source, "/var/log/wasLog/localhost_access_log*.txt"), "AccessLog",
```

```
match(source, "/var/log/wasLog/catalina.*.log"), "CatalinaErrorLog",
match(source, "/var/log/wasLog/localhost.*.log"), "LocalErrorLog",
true(), source)
| timechart span=1d count by source_alias
```

나.

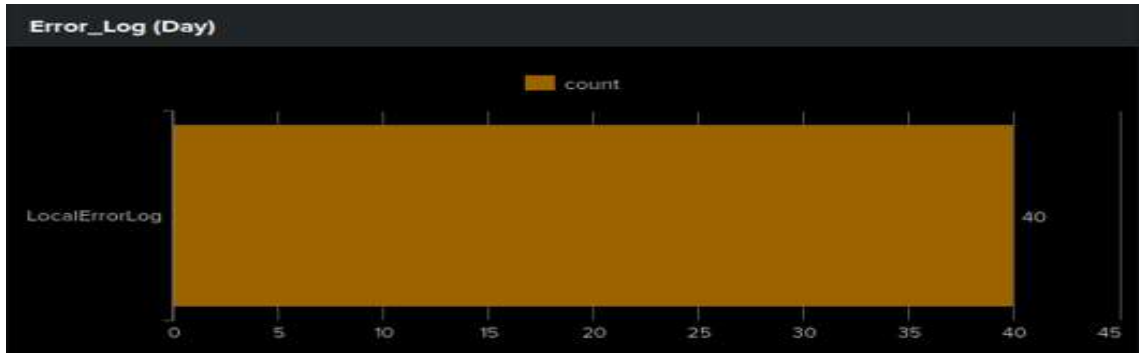


[그림73] WAS Daylog

SPL

```
index=was (source!="var/log/wasLog/per.txt" AND source!="var/log/wasLog/*.pcap" AND
source !="/var/log/syslog") earliest=@d latest=now
| sort Time
| eval source = case(
    match(source, "/var/log/wasLog/localhost_access_log.*.txt"), "AccessLog",
    match(source, "/var/log/wasLog/catalina.*.log"), "CatalinaErrorLog",
    match(source, "/var/log/wasLog/localhost.*.log"), "LocalErrorLog",
    true(), source)
| stats count by source, _time
| timechart span=1m sum(eval(if(source=="AccessLog", count, 0))) as AccessLog,
sum(eval(if(source=="CatalinaErrorLog", count, 0))) as CatalinaErrorLog,
sum(eval(if(source=="LocalErrorLog", count, 0))) as LocalErrorLog
```

다.



[그림73] WAS Daylog

SPL
<pre>index=was (source="/var/log/wasLog/catalina.*.log" OR source="/var/log/wasLog/localhost.*.log") eval source_alias = case(match(source, "/var/log/wasLog/catalina.*.log"), "CatalinaErrorLog",match(source, "/var/log/wasLog/localhost.*.log"), "LocalErrorLog",true()), source) stats count by source_alias</pre>

라.



[그림74]Log level

SPL
<pre>index=was (source="/var/log/wasLog/localhost.*.log" OR source="/var/log/wasLog/catalina.*.log") Level={로그 레벨} stats count by Level, _time timechart span=1d sum(count) as TotalCount by Level</pre>

마.

ACCESS_LOG			
Time ↕	Client_ip ↕	Request ↕	Status ↕
[04/Mar/2024:09:52:24	10.0.0.10	POST / HTTP/1.0	500
[04/Mar/2024:09:52:23	10.0.0.10	GET /.env HTTP/1.0	404
[04/Mar/2024:09:07:30	10.0.0.10	GET /faq/faq.bo HTTP/1.0	200
[04/Mar/2024:08:46:11	10.0.0.10	GET /Pages/log/ HTTP/1.0	404
[04/Mar/2024:08:46:11	10.0.0.10	GET /Temporary_Listen_Addresses HTTP/1.0	404
[04/Mar/2024:07:30:05	10.0.0.10	GET / HTTP/1.0	500

[그림75] Access log

SPL
<pre>index=was source="/var/log/wasLog/localhost_access_log.*.txt" sort - Time table Time, Client_ip, Request, Status</pre>

ERROR_LOG		
Lo_Time ↕	Level ↕	Tread ↕
04-Mar-2024 09:52:24	SEVERE	[http-nio-8080-exec-4]
04-Mar-2024 09:52:24	SEVERE	[http-nio-8080-exec-4]
04-Mar-2024 07:30:05	SEVERE	[http-nio-8080-exec-1]
04-Mar-2024 07:30:05	SEVERE	[http-nio-8080-exec-1]
04-Mar-2024 02:56:30	SEVERE	[http-nio-8080-exec-10]

[그림76] Error log

SPL
<pre>index=was source="/var/log/wasLog/localhost.*.log" OR source="/var/log/wasLog/catalina.*.log" sort - Lo_Time table Lo_Time, Level, Tread</pre>

6. 프로젝트 결론 및 산출물

가상 인프라 구축 및 보안관제 탐지/분석 프로젝트를 진행하며 공격 유형에 따른 즉각적인 공격유무 판단, 공격 대응을 위한 스노트를 작성, 대시보드를 통해 들어오는 공격의 알람과 로그 분석을 통해 정탐/오탐 분석을 할 수 있었던 가치있는 프로젝트였다. 다음과 같이 종합 관제 보고까지 팀원들과 고민해가며 작성하는 경험을 마지막으로 프로젝트 산출물 등을 결과적으로 얻게 되었다.

6.1 종합 관제 보고

3. 보안관제 업무

사이버위협 탐지 및 대응 현황														
구분	계		침해링		비인가접근		악성코드		서비스거부		스캐닝		기타	
	탐지	대응	탐지	대응	탐지	대응	탐지	대응	탐지	대응	탐지	대응	탐지	대응
자체	5,544	5	423	0	1,515	5	0	0	3,525	0	81	0	0	0
000	25,407	0	8	0	46	0	7,257	0	1	0	0	0	36,240	0
000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
총계	30,951	5	431	0	1,561	5	7,257	0	3,526	0	81	0	36,240	0
업무 내용		건수		내역										
보안 이벤트 탐지		5		[IPS/비인가접근] 네트워크 스캔, 악성 프로그램 활동 1건 [IPS/비인가접근] ICMP traceroute 시도 1건 [IPS/비인가접근] RDP 접근시도 2건 [IPS/비인가접근] 디렉터리 비정상 접근 탐지 1건										

5 [IPS/비인가접근] 디렉터리 비정상 접근 탐지_240226_05

- 이벤트 : IPS.Center_Directory Traversal_211010

- 트래픽 : 39.107.77.197(중국) → 211.58.82.254(Bucklist)

- 내용 : 상대경로 문자열을 이용한 Directory Traversal 공격이 탐지되었으며 IPS 정책에 의해 모두 자동 차단되어 공격 영향도 없음. 예방차원에서 공격지 IP 외부망 방화벽에 차단조치 완료

* RawData

: GET /theme/default/img/../../../../../../../../admin/admin.ad HTTP/1.1

Host: www.

User-Agent: Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/34.0.1866.237 Safari/537.36

Connection: close

Accept: */*

Accept-Language: en

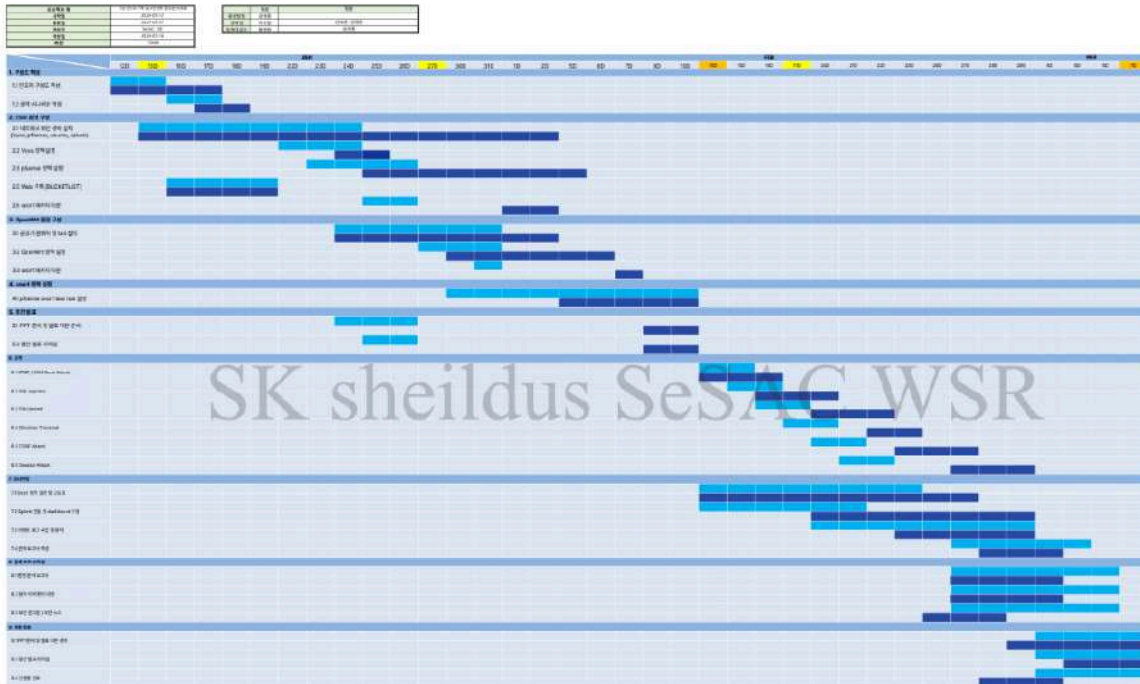
□ 유해 IP 차단

순번	트래픽	공격유형
1	8.212.130.140(마닐라) → 211.58.82.254(Bucklist_Openwrt)	Misc activity
2	47.242.176.194(홍콩) → 211.58.82.254(Bucklist_Openwrt)	Attempted Information Leak
3	76.8.60.135(미국) → 10.0.100.10(Bucklist_ESXI)	Misc activity
4	85.209.11.254(러시아) → 10.0.100.10(Bucklist_ESXI)	Executable code was detected
5	85.209.11.254:39428(러시아) → 211.58.82.226:22	Executable code was detected
6	128.134.38.141:61147(한국) → 211.58.82.226:80	Web Application Attack
7	128.134.38.141: 51427(한국) → 211.58.82.226:80	Web Application Attack
8	176.97.210.4:9093(독일) → 211.58.82.226:7001	Misc activity
9	71.6.134.232:45272(미국) → 211.58.82.254:161	Attempted Information Leak

[그림77] 종합 관제 보고

6.2 프로젝트 산출물

6.2.1 WBS/OPR



[그림78] 산출물1

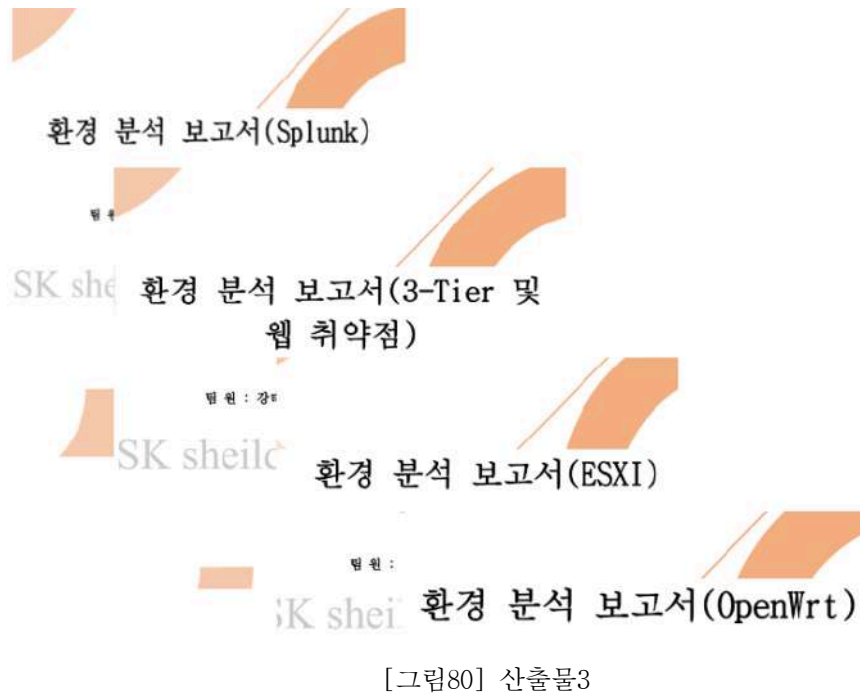
6.2.2 회의록

회의록

회의일시	2024/01/30(화) 오후17:00	장소	보안관제	주최자	김희중
참석자	강태훈, 김영준, 김희중, 이승철, 황정현				
회의장인	환경구축 완료 후 진행상황 숙지				
회의내용	내용		이슈 / 비고		
	1. 1/30 이내 보안관제 기본 환경구축 및 3-tier 연동 구축 완료 2. 1/31 or 2/1 팀원 전체 환경구축 숙지를 위한 A-to-Z 시연 연습				
	1. ptSense 내 Snort 구축 및 Snort 관련 공부 2. Snort Rule 공격유형&공격시나리오 구체화할 것 3. 침입탐지 공격유형별 스노트를 작성 후 공격시행, 탐지결과 확인할 것		- Snort Rule 연구		
	1. 노트북5대 환경구축 및 3-tier 연동 1/30~2/1 내 완료할 것				

[그림79] 산출물2

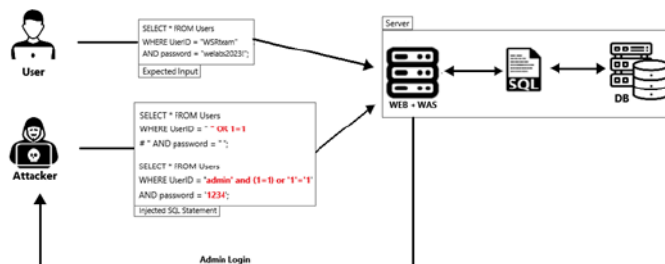
6.2.3 환경 분석 보고서



[그림80] 산출물3

6.2.4 공격 시나리오

공격 시나리오3			
사용 기술	SQL Injection	공격대상	Bucketlist 웹서버
		WEB 주소	http://10.0.100.10/bucketlist/
요약	<p>공격자가 데이터 기반 애플리케이션이나 데이터베이스에 악성 SQL 을 주입하는 사이버 공격으로 데이터베이스와 통신하는데 사용하는 구조적 쿼리인 SQL 을 사용하여 애플리케이션의 보안 조치를 우회하고 데이터베이스의 레코드를 수정, 추가, 업데이트 또는 삭제할 수 있다. SQL Injection을 통해 로그인 우회, 데이터 추출 및 변조, 파일 업로드를 통해 웹에 삽입하여 관리자 권한 획득, 데이터 시스템 완전 파괴 등이 가능하다.</p>		



1. SQL Injection 공격을 위해 구축한 취약 웹 시스템(http://10.0.100.10/bucketlist/)에 SQL Injection 공격을 시행...

SQL Injection 유형
Union SQL Injection
Error Based SQL Injection
Blind SQL Injection

2. 웹 서버에서 반환 정보를 dump하여 csv 파일로 만든 뒤 외부 유출을 시도...

3. 다운로드한 csv 파일을 열어 취약점 정보가 들어있는지 확인...

4. 달취한 취약점정보로 로그인 시도 및 성공.

[그림81] 산출물4

5.2.5 보안 관제 보고서

종합 보안관제 보고서

2024년 3월 4일(일)

결 계	기 안	1차검토	2차검토	결 계

1. 취약점점검

업무 내용								
1. 정보시스템								
1) 정보시스템 정기점검								
일정	대상	자산			진수(%)			비고
		OS	WAS	DBMS	OS	WAS	DBMS	
02. 25 ~ 03.04	Bucklist_Openwrt	10	2	1	51 (6.0)	52 (34.2)	26 (44.1)	
	Bucklist_ESXI	5	2	2	27 (10.7)	11 (27.5)	확인중	
계		15	4	3	78 (41)	63 (48)	26	
2. 모의해킹								
1) 내부망 모의해킹								
일정	대상	취약점진수		취약점		비고		
02. 25 ~ 03.04	Bucklist_Openwrt	진행중		CSRF				
02. 25 ~ 03.04	Bucklist_ESXI	진행중		XSS Injection URI				

[그림82] 산출물5

6.2.6 보안 뉴스 및 보안 권고문

보안 권고문

스크랩 일자 2024-02-23

출처

KISA 보안공지

URL

<https://www.bobo.or.kr/kr/bbs/view.do?searchCnd=1&bbsId=8000013&searchWrd=&menuNo=205000&pageIndex=1&categoryCode=&ntId=71334>

제목

BIND DNS 취약점 보안 업데이트 권고

□ 개요

- o ISC(Internet Systems Consortium) 업데이트 발표 [1] [2]
- o 공격자는 해당 취약점을 악용하여 파. 권고

□ 설명

- o DNSSEC 서명결증 중 무한루프 진입: 50387) [2]

□ 영향을 받는 버전 및 해결 버전

취약점	해결버전
CVE-2023-50387	BIND 9

o 참고사이트에 명시되어 있는 'BIND 9'

□ 기타 문헌사항

- o 한국인터넷진흥원 사이버변환센터: 국

[참고사이트]

- [1] <https://kb.isc.org/vi/docs/cve-2023-5>
- [2] <https://vuln.nist.gov/vuln/detail/CVE-2023-50387>
- [3] <https://www.isc.org/download>

□ 작성 : WSR 보안관제팀 침해사고분석

기타사항

보안뉴스

일자	24.02.28	작성일자	24.03.28
URL	https://www.baenews.com/media/view.do?article=127925		
제목	미국식 시공법 3개도 7차 1차년도 2차년도 2차년도		

1. 2차년도 2

[그림83] 산출물6

6.2.7 발표자료



[그림84] 산출물7

6.2.8 스노트 Community rule 고도화 보고서(해석 보고서)

공격 제목	설명	검출된 룰
Dagger_1.4.0 백도어 공격 시도	이 규칙은 Dagger_1.4.0 백도어 악성 소프트웨어의 네트워크 트래픽을 탐지하는 규칙이다. 해당 악성 소프트웨어는 TCP 포트 2589를 사용하여 외부 네트워크로부터 내부 네트워크 드라이브 정보를 전송하고 있다. 이 규칙은 해당 트래픽의 패킷을 분석하여 드라이브 정보를 포함하는 패킷을 탐지하고, 이를 통해 해당 악성 소프트웨어의 존재를 알리게 된다. 이를 통해 네트워크 보안 담당자는 해당 악성 소프트웨어의 존재를 파악하고 대응할 수 있다.	tcp \$HOME_NET 2589 -> \$EXTERNAL_NET any (msg:"MALWARE-BACKDOOR - Dagger_1.4.0"; flowto:client,established; content:"200 00 00 06 00 00 00[Drives]24 00"; depth:16; metadata:ruleset community, classtype:misc-activity; sid:105; rev:14); #
QAZ Worm Client Login access	이 snort rule은 QAZ worm client의 로그인 접근을 탐지하는 역할을 한다. 외부 네트워크로부터 7597 포트로 전송되는 모든 TCP 패킷을 검사하며, 패킷이 "qazwsx.hsq"라는 내용이 포함되어 있을 경우 해당 패킷을 탐지한다. 이 패턴은 QAZ worm의 클라이언트가 서버에 접근하는 시도를 탐지하는 데 사용된다.	tcp \$EXTERNAL_NET any -> \$HOME_NET 7597 (msg:"MALWARE-BACKDOOR QAZ Worm Client Login access"; flowto:server,established; content:"qazwsx.hsq"; metadata:ruleset community, classtype:misc-activity; sid:108; rev:12); #
# msg:"MALWARE-BACKDOOR netbus getinfo"; flowto:server,established; content:"GetInfo[00]"; metadata:ruleset community, classtype:trojan-activity; sid:110; rev:10)	이 규칙은 외부 네트워크에서 내부 네트워크로 TCP 프로토콜을 통해 전송되는 포트 12345 또는 12346으로 들어오는 패킷을 감시합니다. 해당 패킷에는 "GetInfo"라는 문자열이 포함되어 있으며, 해당 패킷은 서버와 연결되어 있어야 합니다. 이 패킷은 백도어 프로그램인 넷버스의 GetInfo 명령을 실행하는 것으로 추정되며, 악성행위로 분류됩니다.	tcp \$EXTERNAL_NET any -> \$HOME_NET 12345:12346 (msg:"MALWARE-BACKDOOR netbus getinfo"; flowto:server,established; content:"GetInfo[00]"; metadata:ruleset community, classtype:trojan-activity; sid:110; rev:10); #
MALWARE-BACKDOOR NetBus Pro 2.0 connection established	해당 규칙은 NetBus Pro 2.0 백도어가 연결이 성립되었을 때 발생하는 패킷을 탐지하는 규칙이다. NetBus Pro 2.0은 백도어 프로그램으로, 외부 공격자가 내부 시스템에 접근하여 제어를 할 수 있게 해주는 악성 소프트웨어이다. 이 규칙은 내부 시스템의 IP 주소와 포트 번호가 20034인 경우, 외부 시스템으로부터 연결이 성립되었을 때 해당 패킷을 탐지한다. 패킷의 내용 중 "BN[10 00 02 00]"와 "05 00"이라는 문자열이 포함되어야 하며, 이는 NetBus Pro 2.0의 특징적인 프로토콜을 나타내는 값이다. 이 규칙을 통해 NetBus Pro 2.0 백도어의 연결을 탐지하여 시스템의 보안을 강화할 수 있다.	tcp \$HOME_NET 20034 -> \$EXTERNAL_NET any (msg:"MALWARE-BACKDOOR NetBus Pro 2.0 connection established"; flowto:client,established; flowbits:isset,backdoor.netbus.2.connect; content:"BN[10 00 02 00]"; depth:6; content:"05 00"; depth:2; offset:8; metadata:ruleset community, classtype:trojan-activity; sid:115; rev:15); #
Infector.1.x	해당 공격은 외부 네트워크에서 내부 네트워크로 접근하는 공격을 탐지하는 규칙이다. 메시지는 WHATISIT이라는 문자열을 포함한 패킷을 탐지한다. flow는 클라이언트에서 서버로 전송하는 패킷을 대상으로 한다. 이 공격은 백도어를 이용해 외부에서 내부 네트워크에 접근하려는 시도를 탐지할 수 있다.	tcp \$HOME_NET any -> \$EXTERNAL_NET any (msg:"MALWARE-BACKDOOR Infector 1.x"; flow:established,to_client; content:"WHATISIT"; depth:9; metadata:impact_flag red, ruleset community, references:nessus,11157; classtype:misc-activity; sid:117; rev:17); #
MALWARE-BACKDOOR Satansbackdoor.2.0 Beta	이 서명은 SatansBackdoor 2.0 Beta 백도어를 탐지하기 위한 것이다. 이 백도어는 외부 네트워크에서 내부 네트워크의 포트 666으로 연결하고, 연결이 성립되면 "Remote: "라는 단어를 포함한 패킷을 보낸다. 또한 "You are connected to me. Remote: Ready for commands"라는 메시지를 포함한 패킷을 보내는 것이 탐지 대상이다. 이 백도어는 원격 명령을 수신하고 실행할 수 있으며, 이를 통해 내부 시스템에 대한 외부 공격이 가능하다. 따라서 이 서명을 통해 내부 시스템에 대한 외부 공격을 탐지할 수 있다.	tcp \$HOME_NET 666 -> \$EXTERNAL_NET any (msg:"MALWARE-BACKDOOR SatansBackdoor 2.0 Beta"; flowto:client,established; content:"Remote[3A] "; depth:11; nocase; content:"You are connected to me[0A][Remote]3A[Ready for commands"; distance:0; nocase; metadata:ruleset community, references:www.megasecurity.org/trojans/satanbackdoor/SB02.lib.html; reference:www3.ca.com/securityadvisor/pest/pest.aspx?id=5260; classtype:trojan-activity; sid:118; rev:12); #

[그림85] 산출물8