2023.12.28

DB보안 솔루션 개발자 양성

데이터베이스 공격 탐지 및 차단 서비스

팀 명 : Sentinel

CONTENTS

01

프로젝트 소개

프로젝트 소개

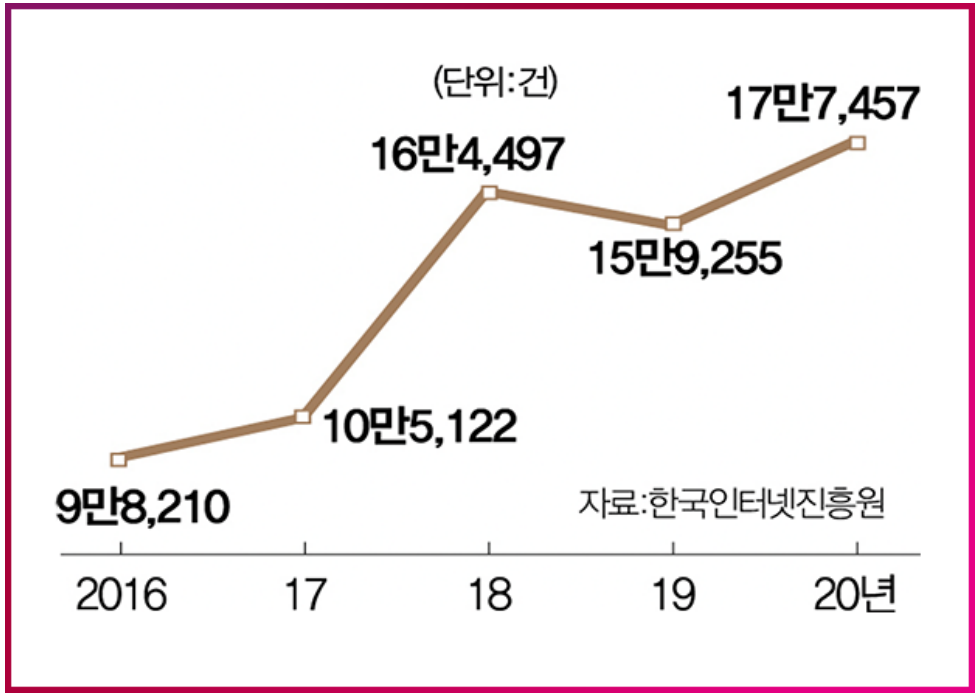
DB보안 솔루션

사용자 DB를 보호하기 위한 프로그램

DB는 비즈니스 의사결정을 지원하고, 고객정보, 재고관리, 금융데이터 등을 저장합니다. 또한 DB는 빅데이터 분석에도 필수적으로 활용되고 이를 활용하여 기업들은 경쟁 우위를 확보할 수 있습니다. 따라서 DB에 대한 보안은 중요하고 유출이나 외부 공격으로부터 안전하게 관리되어야 합니다.

프로젝트 소개

개인정보 유출 사례



금융사	내용
DB손해보험	직원이 외부인에 170여명의 고객 정보 전달
삼성증권	모니모 앱에서 다른 사람의 이름, 보유 주식 등 조회
KB국민카드	모바일 앱에서 다른 고객 카드 이용정보 조회
하나금융	마이데이터 서비스에서 본인 아닌 타인 정보 조회
네이버파이낸셜	마이데이터 서비스에서 본인 아닌 타인 정보 조회

프로젝트 멤버

박희승(팀장)
프로젝트 관리
탐지 엔진 개발

양주목
탐지 엔진 개발
탐지 UI 디자인

정찬진
탐지 엔진 개발

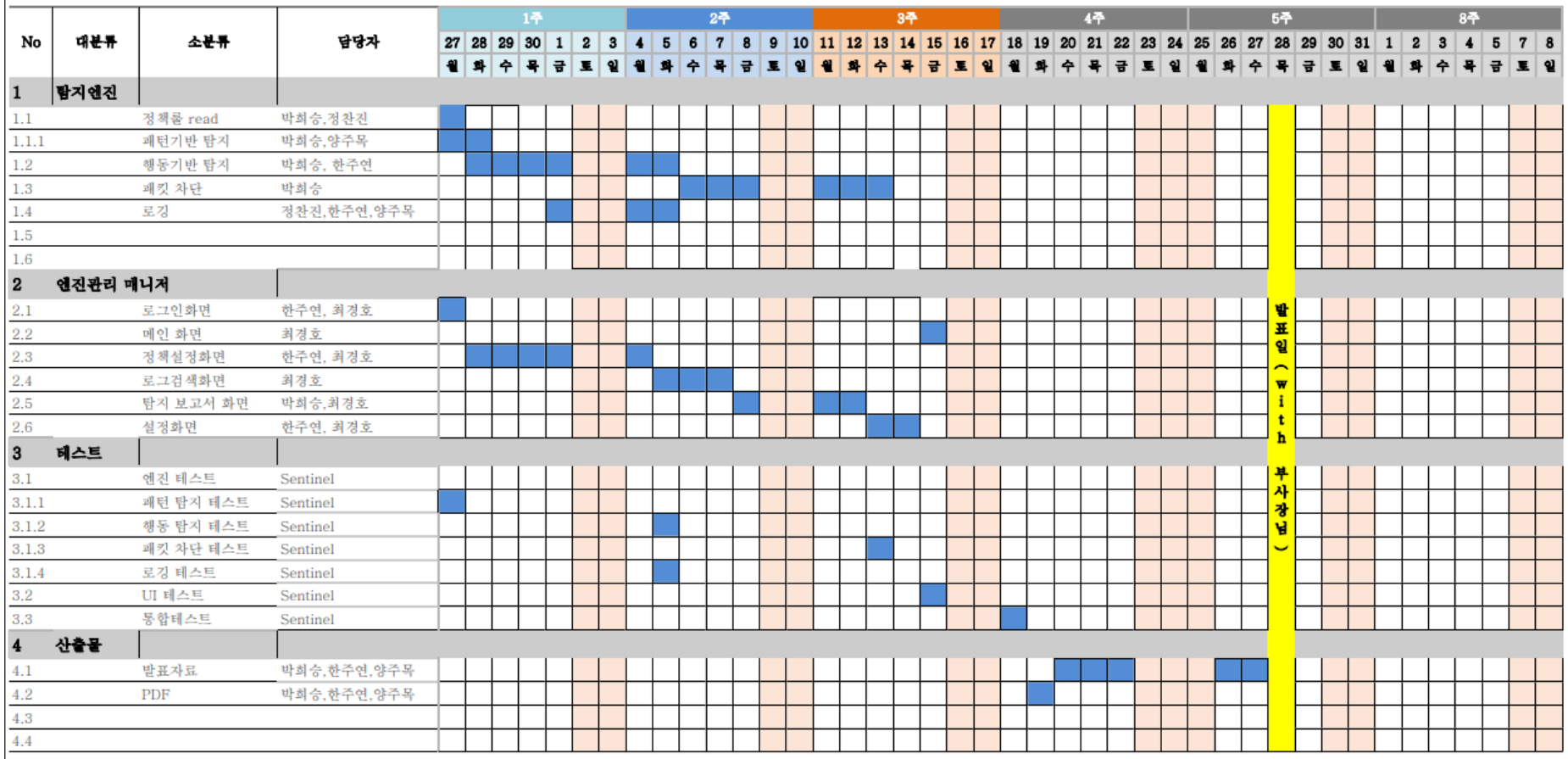
최경호
탐지 UI 개발
탐지 UI 디자인

한주연
탐지 엔진 개발
탐지 UI 개발 디자인

프로젝트 일정

프로젝트 일정

프로젝트 이름	Gauadian	프로젝트 조이름	Sentinel
프로젝트 관리자	박희승, 양주목, 최경호, 정찬진, 한주연	날짜	2023-11-07 ~ 2023-12-28



활용 방안 및 기대 효과

1. 암호화 기술 적용
  - 저장된 비밀번호 등의 중요한 데이터를 암호화하여 보호

2. 감시 및 로깅
  - 데이터 베이스 활동을 감시하고 로그를 기록하여 이상 징후를 신속하게 감지하고 대응
  - 실시간으로 데이터 베이스 활동을 모니터링하고 로그 정보를 분석함으로써 보안 이슈에 대한 대응이 가능

02

프로그램 구현

프로그램 주요기능

탐지 엔진

공격 탐지

1. IP, PORT 탐지
2. 패턴 탐지
3. 행동 탐지

로그 기록

1. 탐지 로그 기록

정책 관리

1. 정책 업데이트

엔진 UI

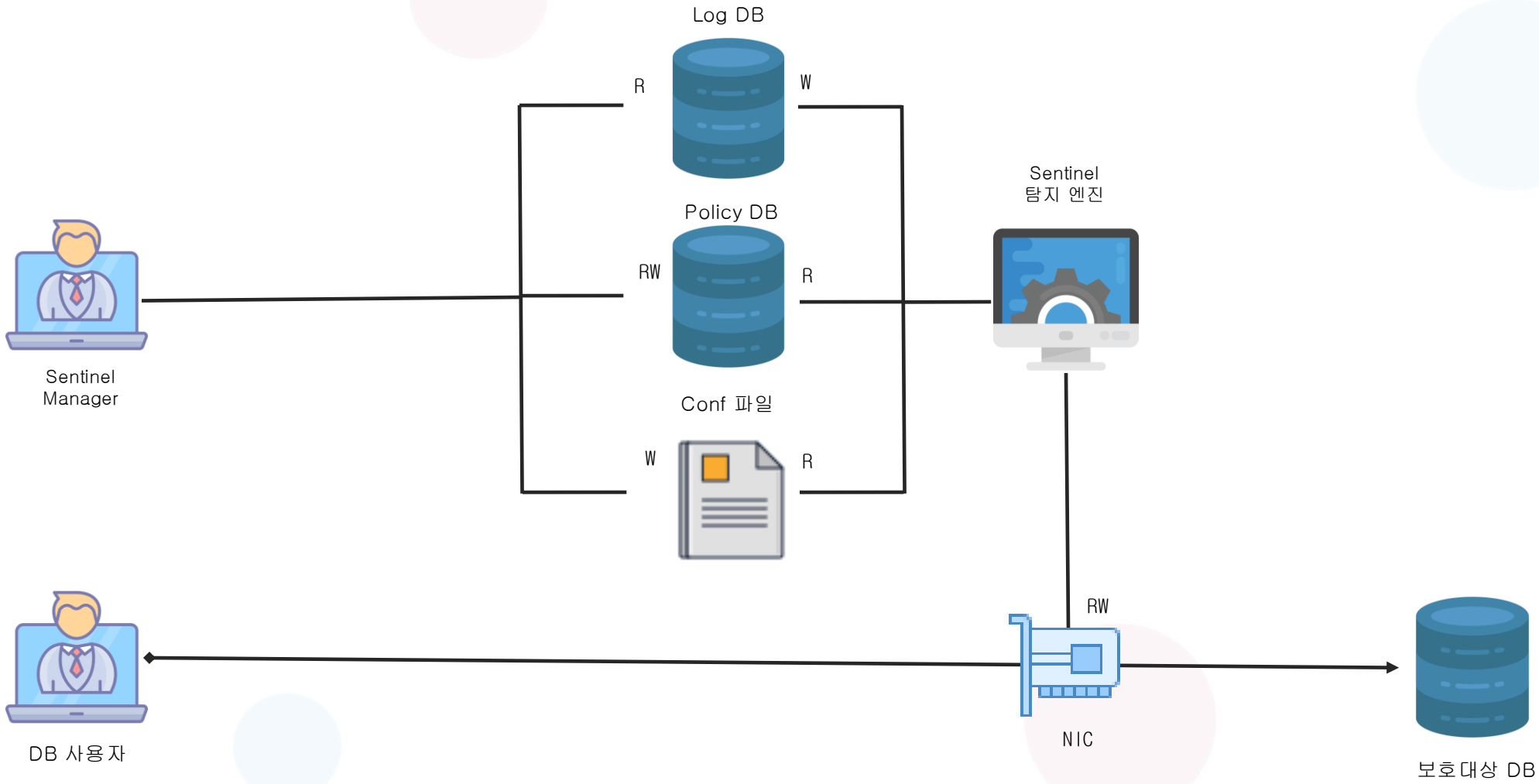
정책 관리

1. 정책 추가
2. 정책 수정
3. 정책 삭제

로그 조회

1. 로그 필터링 검색
2. 로그 상세 보기

시스템 아키텍쳐


Log DB
R
W
Policy DB
RW
R
Conf 파일
W
R
Sentinel
탐지 엔진
Sentinel
Manager
RW
NIC
DB 사용자
보호대상 DB

데이터베이스 설계

Name	S_manage_user.user_info	Table 기술서	작성일	20231130	Page
System	Manager_User		작성자	Sentinel	
Description	IPS_MANAGER 유저정보 테이블				

No	Attribute	Data Type	Length	Null	KY	Default	Description
1	personal_no	int	11	N	PK		고유번호
2	id	varchar	50	N			아이디
3	password	char	64	N			비밀번호
4	salt	varchar	20	N			임의 해시 데이터
5							
6							
7							
8							
9							
10							
11							
12							

비고

데이터베이스 설계

Name	S_ips_policy_db.ips_policy	Table 기술서	작성일	20231121	Page
System	dbms_ips_manager		작성자	Sentinel	
Description	탐지룰 설정 테이블				

No	Attribute	Data Type	Length	Null	KY	Default	Description
1	detected_no	int	6	N	PK		룰번호
2	detected_name	varchar	50	N			룰 이름
3	content1	varchar	255	Y		null	탐지 문구 1
4	content2	varchar	255	Y		null	탐지 문구 2
5	content3	varchar	255	Y		null	탐지 문구 3
6	enable	int	1	N		1	0 : 룰 실행 X 1 : 룰 실행 O
7	src_ip	varchar	15	Y		0	
8	src_port	int	5	Y		0	
9	action	int	1	N		0	0 : 로그 1 : 로그+차단
10	level	int	1	N		0	위험순위
11	base_time	int	11	Y		0	행동 탐지 기준 초
12	base_limit	int	11	Y		0	행동 탐지 기준 쿼리 수
13	end_time	date		N		1	유효기간
14	detail	varchar	255	Y			탐지 내용
15	to_sip	varchar	15	Y		0	범위입력시 sip
16	to_sp	int	5	Y		0	범위입력시 sp
17	dst_ip	varchar	15	Y		0	탐지할 IP
18	base_size	int	11	Y		0	행동 탐지 기준 데이터 사이즈
비고							

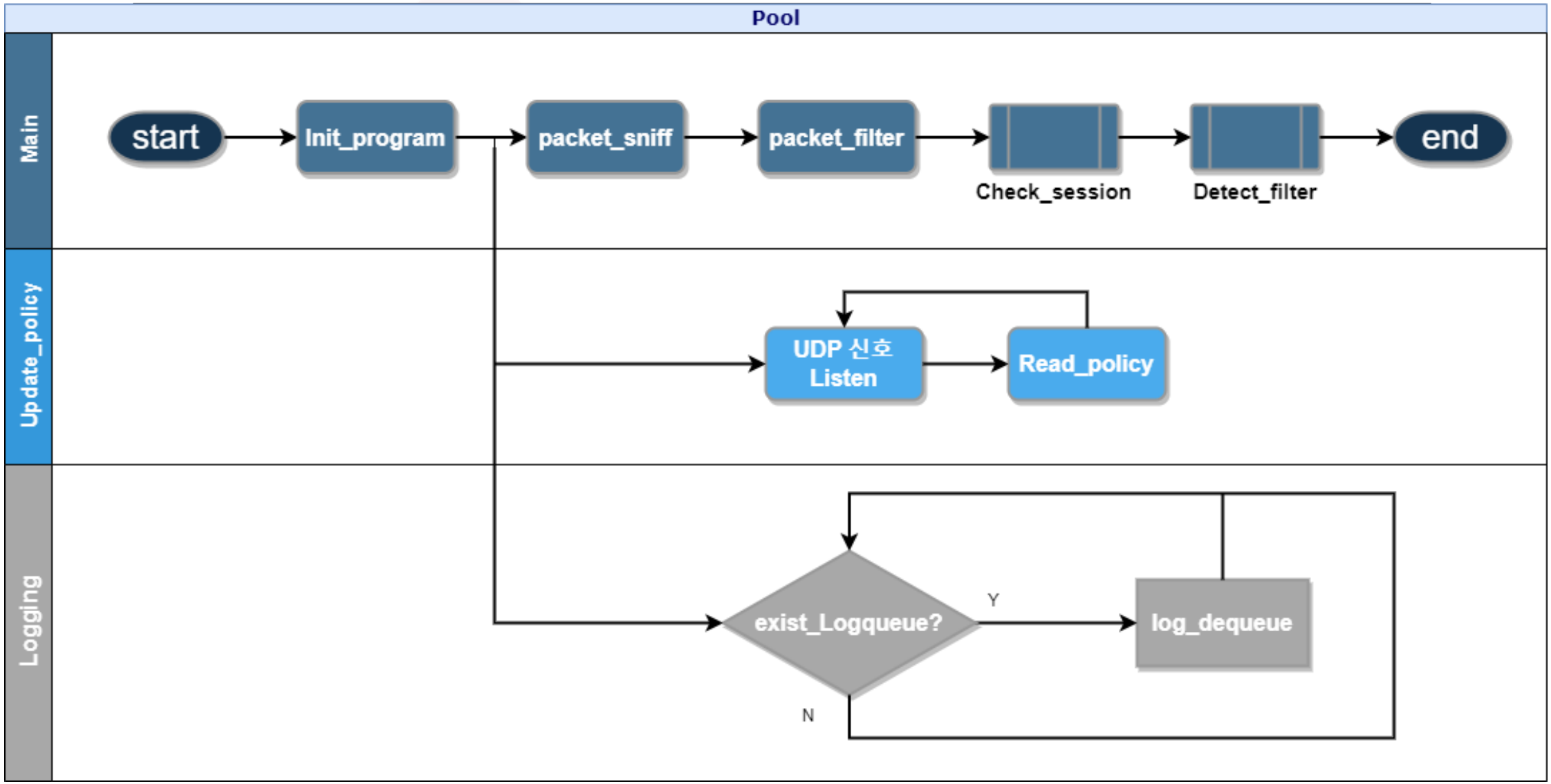
데이터베이스 설계

Name	S_ips_log_db.log_YYYYMMDD	Table 기술서	작성일	20231121	Page
System	dbms_ips_manager		작성자	Sentinel	
Description	탐지결과 로그 테이블				

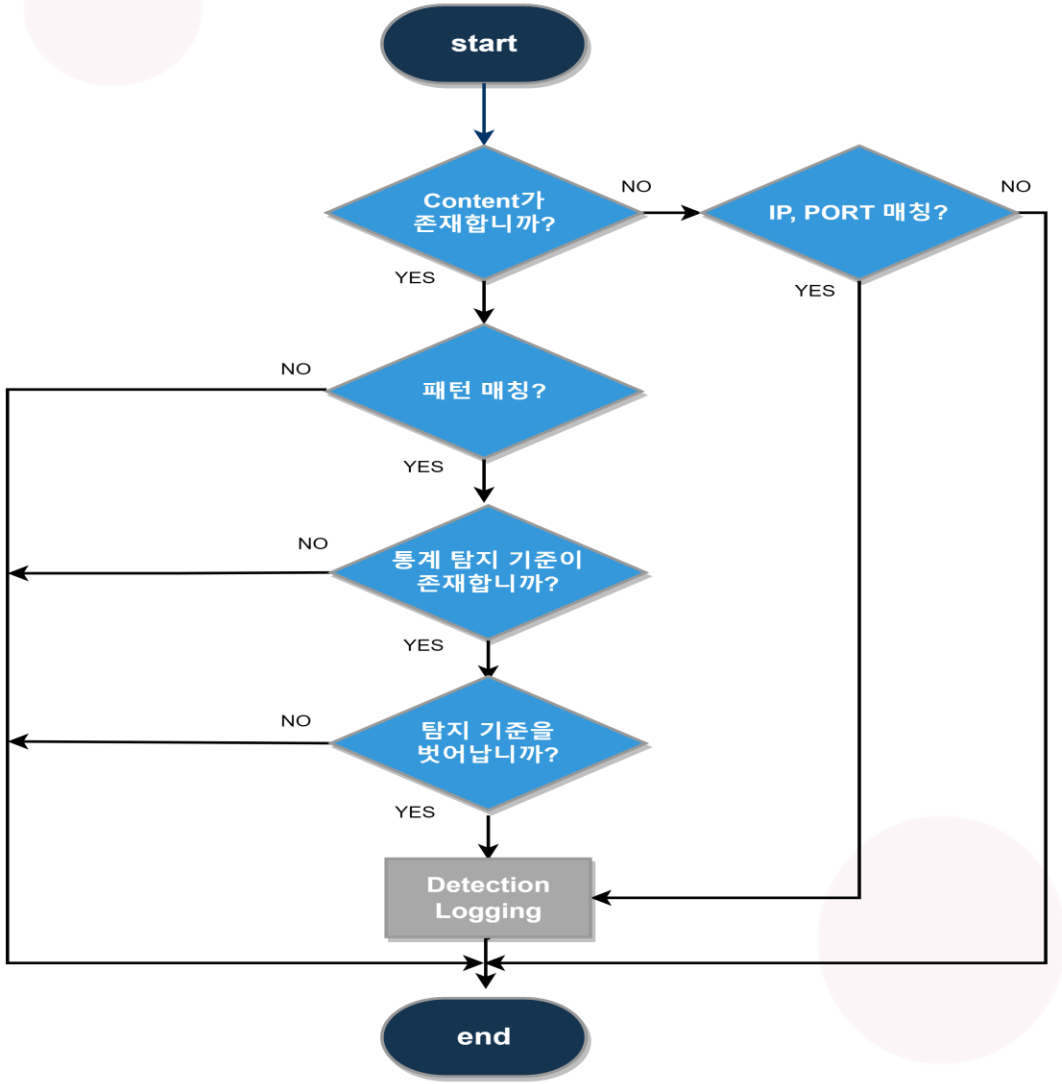
No	Attribute	Data Type	Length	Null	KY	Default	Description
1	log_index	int	11	N	PK		로그 번호
2	detected_no	int	6	N			탐지 룰 번호
3	detected_name	varchar	50	N			탐지 룰 이름
4	time	timestamp		N			기록 시간
5	action	int	1	N			탐지 행동 유형
6	src_ip	varchar	15	N			Source IP
7	packet_bin	varbinary	3000	N			패킷 내용
8	level	int	5	N			위험순위
9	src_port	int	5	N			Source Port
10	dst_ip	varchar	15	N			Destiny IP
11	dst_port	int	5	N			Destiny Port

비고
packet에서 데이터를 뽑을 수 있지만 검색할 때 용이하게 하기위해 src_ip를 따로 저장

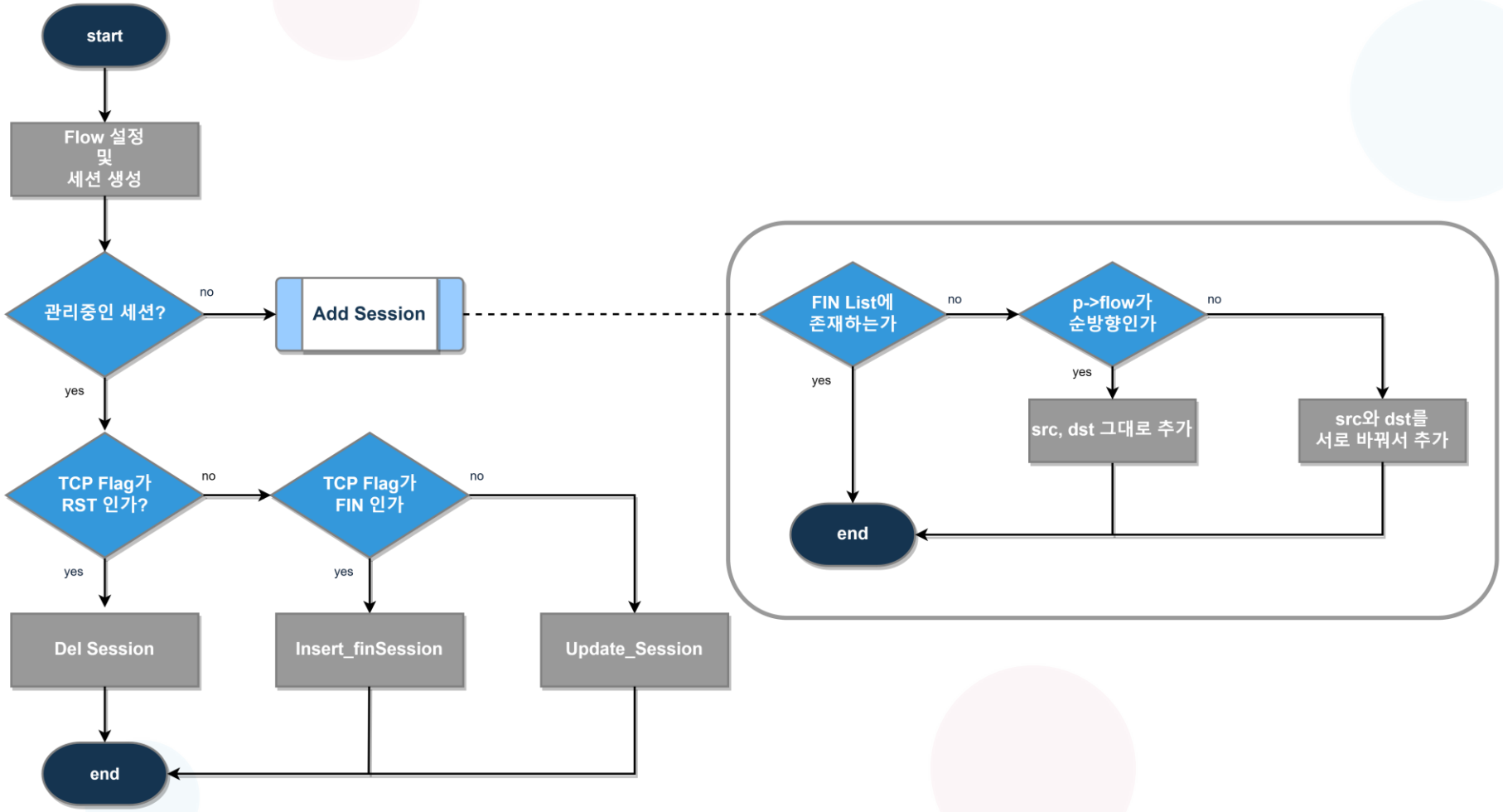
Flow Chart


Pool
Main
start
Init_program
packet_sniff
packet_filter
Check_session
Detect_filter
end
Update_policy
UDP 신호
Listen
Read_policy
Logging
exist_Logqueue?
Y
N
log_dequeue

Flow Chart(Detect_Filter)

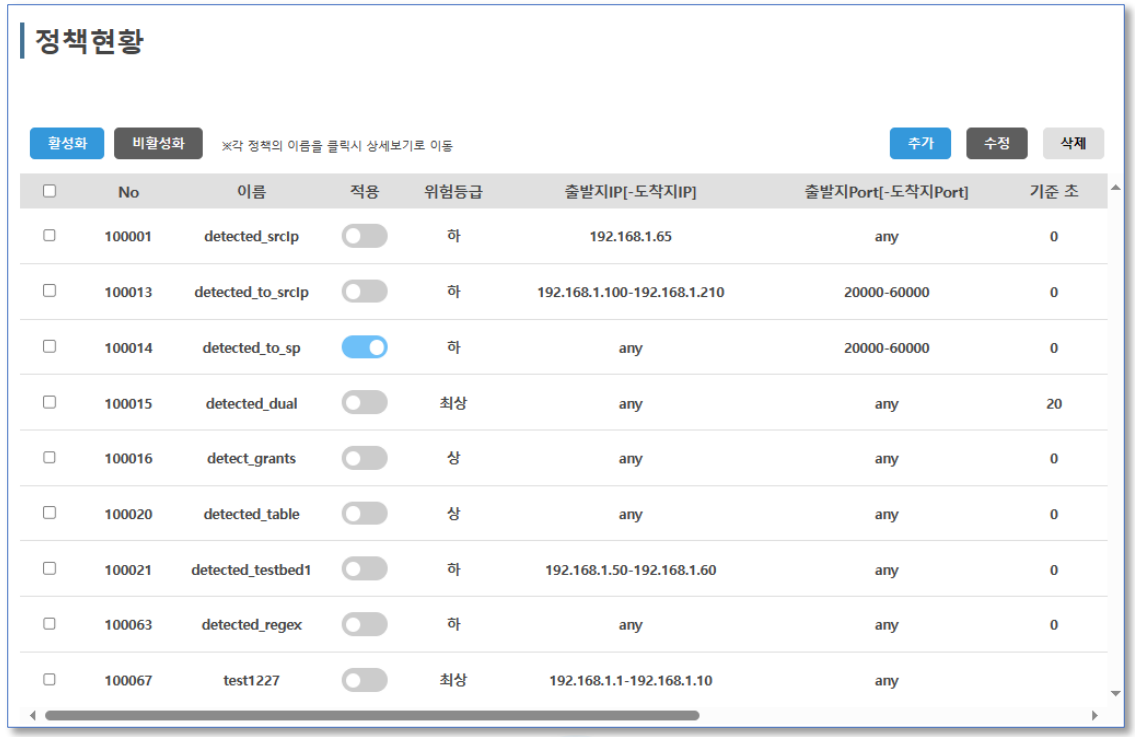


Flow Chart(Check_session)



주요 기능

Port 범위 탐지



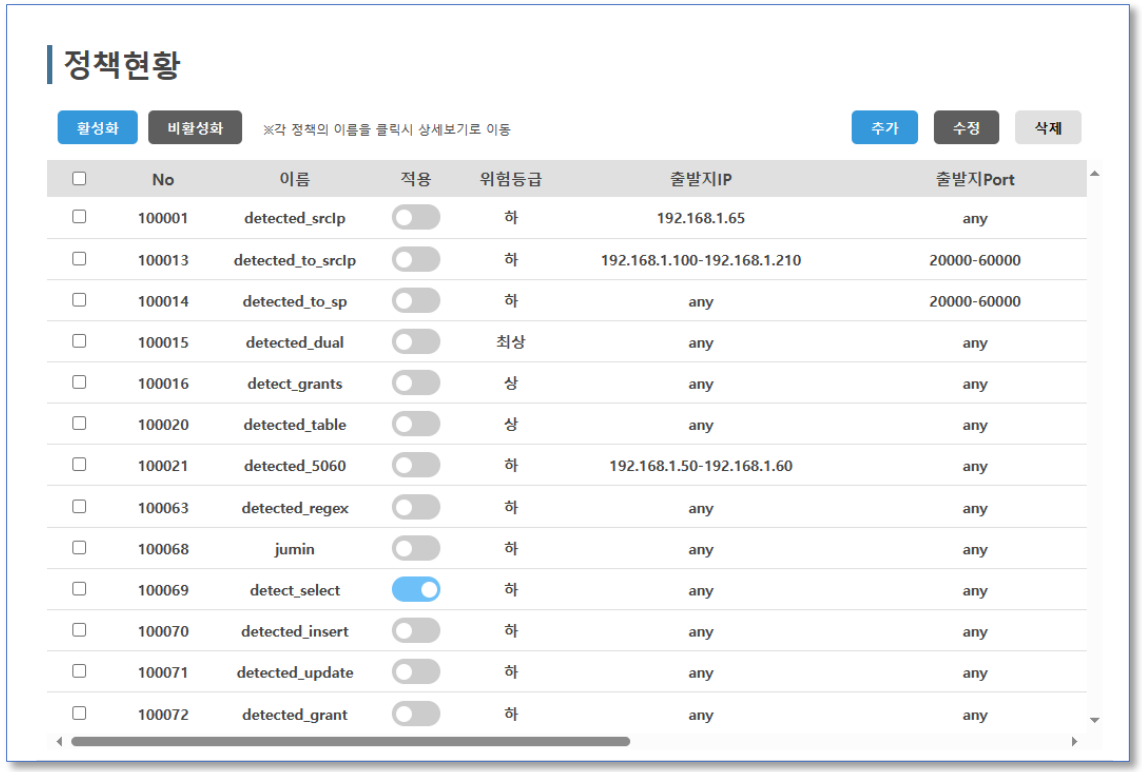
주요 기능

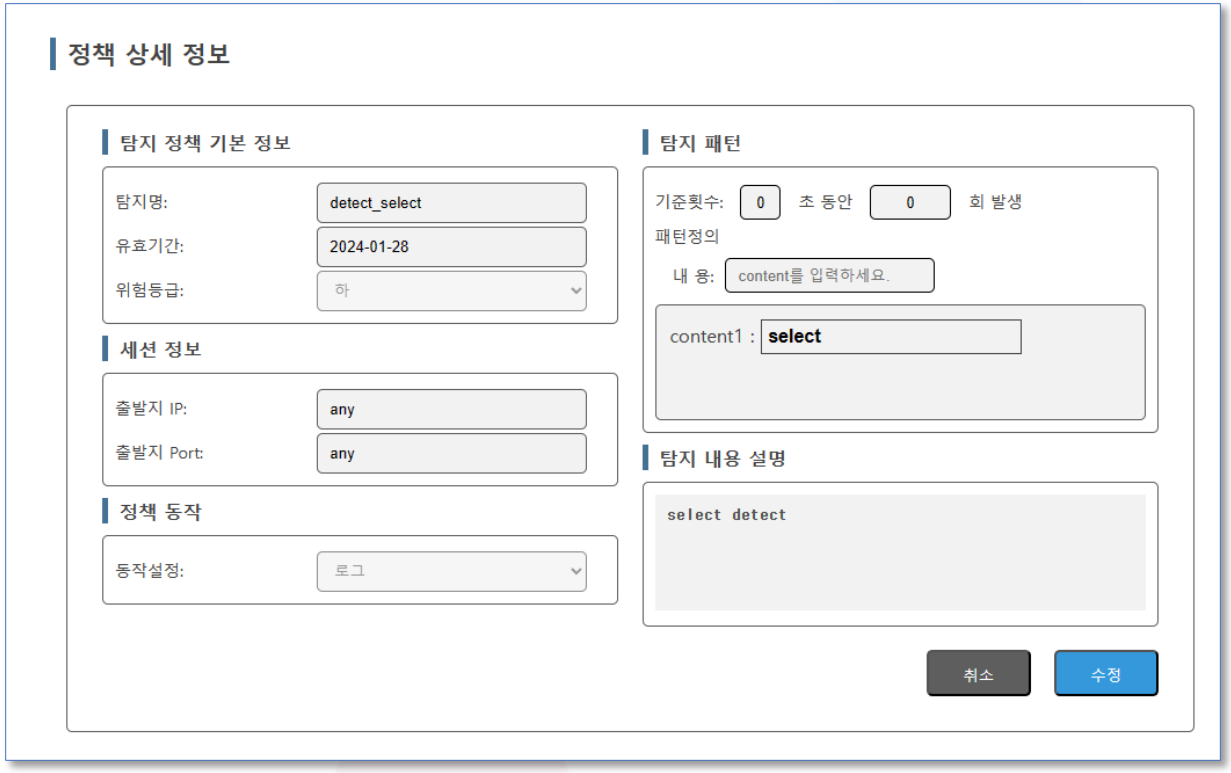
Port 범위 탐지

```
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.41(F4:C8:8A:25:D3:34) SP:30728 -> DIP:192.168.1.107(8C:71:F8:F6:03:56) DP:3306 len:
66 flag:.S.... ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.119(00:E0:4C:5B:FD:F1) SP:58335 -> DIP:192.168.1.9(00:0E:0C:00:CD:7B) DP:3306 len:6
0 flag:...PA. ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.57(00:E0:4C:5B:FD:F1) SP:29808 -> DIP:192.168.1.19(00:0E:0C:00:CD:7B) DP:3306 len:6
0 flag:...PA. ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.73(00:E0:4C:5B:FD:F1) SP:28569 -> DIP:192.168.1.33(00:0E:0C:00:CD:7B) DP:3306 len:6
6 flag:.S.... ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.99(00:E0:4C:5B:FD:F1) SP:50594 -> DIP:192.168.1.35(00:0E:0C:00:CD:7B) DP:3306 len:6
0 flag:...PA. ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.33(00:E0:4C:5B:FD:F1) SP:25958 -> DIP:192.168.1.35(00:0E:0C:00:CD:7B) DP:3306 len:6
0 flag:...PA. ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.71(00:E0:4C:5B:FD:F1) SP:51785 -> DIP:192.168.1.91(00:0E:0C:00:CD:7B) DP:3306 len:6
0 flag:...PA. ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.35(00:E0:4C:5B:FD:F1) SP:37209 -> DIP:192.168.1.5(00:0E:0C:00:CD:7B) DP:3306 len:60
 flag:...PA. ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.21(00:E0:4C:5B:FD:F1) SP:33568 -> DIP:192.168.1.79(00:0E:0C:00:CD:7B) DP:3306 len:6
0 flag:...PA. ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.15(F8:C2:88:FF:95:44) SP:36424 -> DIP:192.168.1.21(00:0E:0C:00:CD:7B) DP:3306 len:6
0 flag:...PA. ttl:125
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.27(A0:48:1C:D5:54:9E) SP:49509 -> DIP:192.168.1.37(78:2B:CB:51:B0:F3) DP:3306 len:6
6 flag:.S.... ttl:128
(Detected_Name : detected_to_sp)
[00:00:00] ETH:0800 PROT:tcp SIP:192.168.1.14(00:00:00:00:00:00) SP:59180 -> DIP:192.168.1.14(00:00:00:00:00:00) DP:3306 len:8
7 flag:...PA. ttl:64
```

주요 기능

패턴 탐지


정책현황
활성화
비활성화
※각 정책의 이름을 클릭시 상세보기로 이동
추가
수정
삭제
No
이름
적용
위험등급
출발지IP
출발지Port
100001 detected_srclp 하 192.168.1.65 any
100013 detected_to_srclp 하 192.168.1.100-192.168.1.210 20000-60000
100014 detected_to_sp 하 any 20000-60000
100015 detected_dual 최상 any any
100016 detect_grants 상 any any
100020 detected_table 상 any any
100021 detected_5060 하 192.168.1.50-192.168.1.60 any
100063 detected_regex 하 any any
100068 jumin 하 any any
100069 detect_select 하 any any
100070 detected_insert 하 any any
100071 detected_update 하 any any
100072 detected_grant 하 any any


정책 상세 정보
탐지 정책 기본 정보
탐지명: detect_select
유효기간: 2024-01-28
위험등급: 하
세션 정보
출발지 IP: any
출발지 Port: any
정책 동작
동작설정: 로그
탐지 패턴
기준횟수: 0 초 동안 0 회 발생
패턴정의
내 용: content를 입력하세요.
content1 : select
탐지 내용 설명
select detect
취소
수정

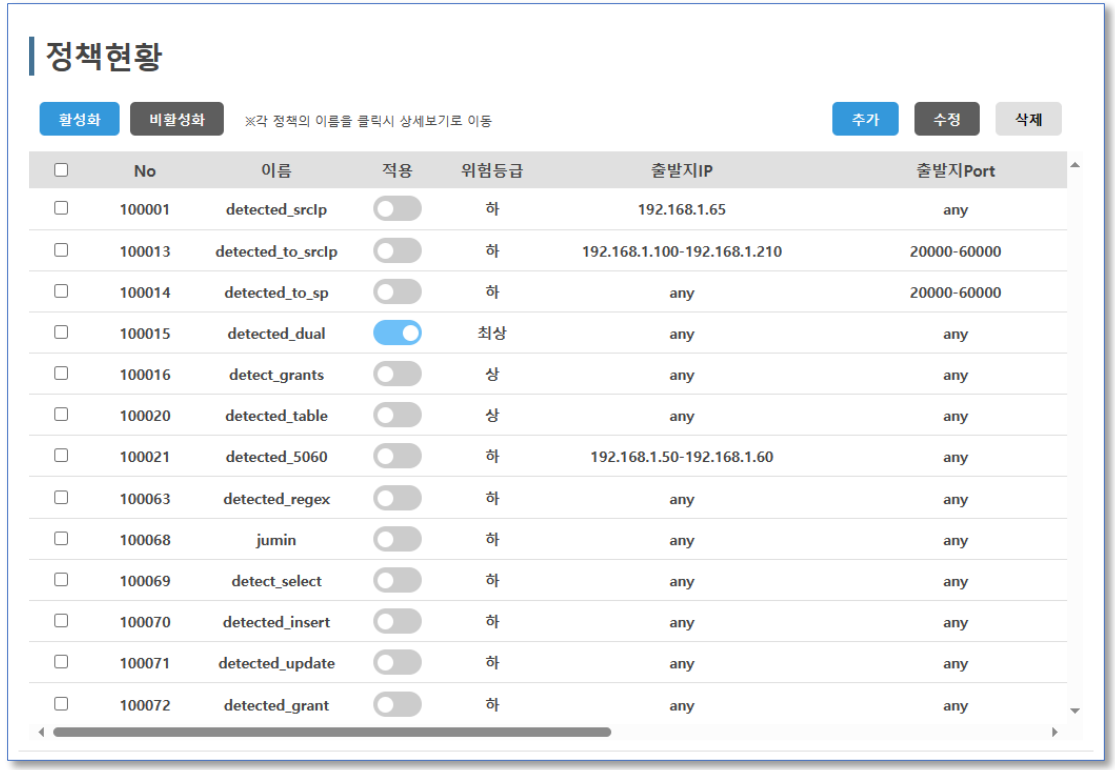
주요 기능

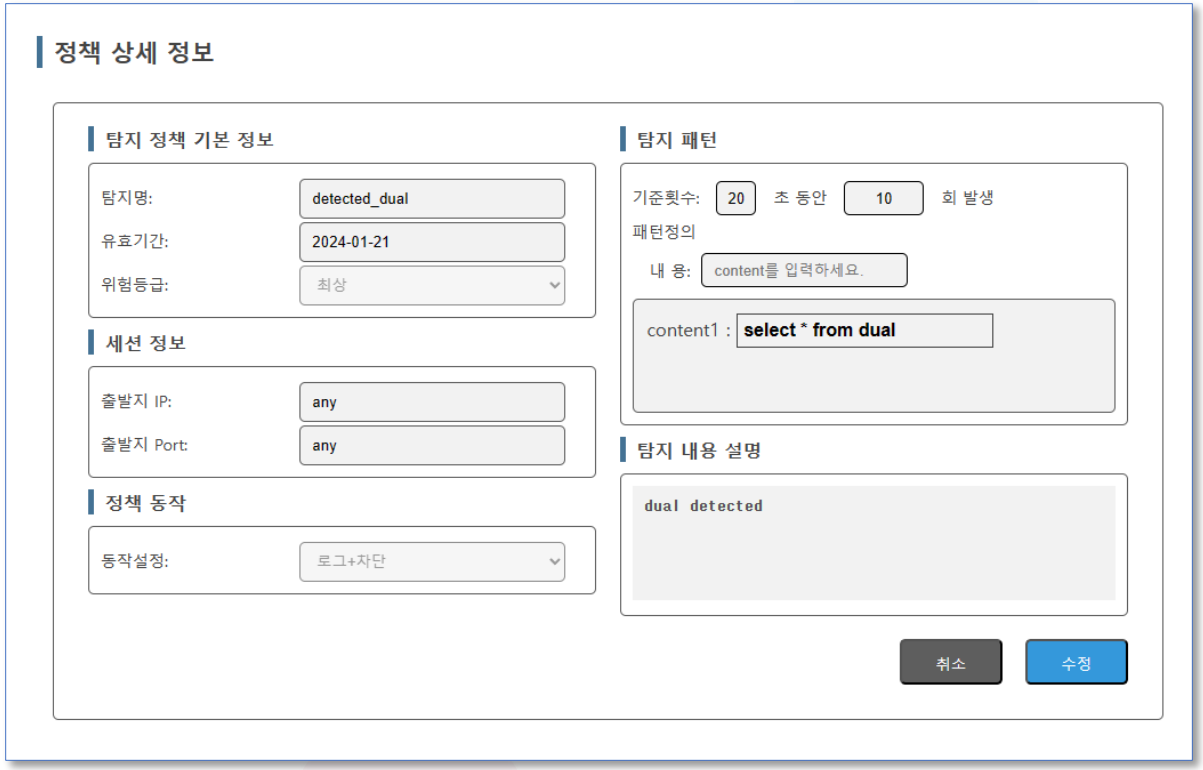
패턴 탐지

```
[12:12:54] ETH:0800 PROT:tcp SIP:192.168.43.67(00:0E:0C:00:CD:7B) SP:1631 -> DIP:192.168.103.117(00:E0:81:4C:D4:62) DP:1521 le
n:129 flag:...PA. ttl:128
[000] 00 E0 81 4C D4 62 00 0E  0C 00 CD 7B 08 00 45 00  00 73 70 0E 40 00 80 06  ...L.b.....{..E..sp.@...
[018] 76 6D C0 A8 2B 43 C0 A8  67 75 06 5F 05 F1 55 CE  E8 99 13 6E 7D AF 50 18  vm..+C..gu._..U....n}.P.
[030] FF FF 5C 4A 00 00 00 4B  00 00 06 00 00 00 00 00  03 5E 00 02 80 21 00 01  ..\J...K.........^...!..
[048] 01 12 01 01 0D 00 00 00  00 04 7F FF FF FF 00 00  00 00 00 00 00 00 00 12  ........................
[060] 53 45 4C 45 43 54 20 2A  20 46 52 4F 4D 20 44 55  41 4C 01 01 00 00 00 00  SELECT * FROM DUAL......
[078] 00 00 01 01 00 00 00 00  00                                                .........

(Detected_Name : detect_select)
[12:12:54] ETH:0800 PROT:tcp SIP:192.168.119.113(00:0E:0C:00:CD:7B) SP:1632 -> DIP:192.168.237.7(00:E0:81:4C:D4:62) DP:1521 le
n:129 flag:...PA. ttl:128
[000] 00 E0 81 4C D4 62 00 0E  0C 00 CD 7B 08 00 45 00  00 73 74 2F 40 00 80 06  ...L.b.....{..E..st/@...
[018] A0 8B C0 A8 77 71 C0 A8  ED 07 06 60 05 F1 22 60  9D 17 16 DB BD E0 50 18  ....wq.....`.."`......P.
[030] FF FF C5 DA 00 00 00 4B  00 00 06 00 00 00 00 00  03 5E 00 02 80 21 00 01  .......K.........^...!..
[048] 01 12 01 01 0D 00 00 00  00 04 7F FF FF FF 00 00  00 00 00 00 00 00 00 12  ........................
[060] 53 45 4C 45 43 54 20 2A  20 46 52 4F 4D 20 44 55  41 4C 01 01 00 00 00 00  SELECT * FROM DUAL......
[078] 00 00 01 01 00 00 00 00  00                                                .........

(Detected_Name : detect_select)
[12:12:54] ETH:0800 PROT:tcp SIP:192.168.133.23(00:0E:0C:00:CD:7B) SP:1633 -> DIP:192.168.141.57(00:E0:81:4C:D4:62) DP:1521 le
n:129 flag:...PA. ttl:128
[000] 00 E0 81 4C D4 62 00 0E  0C 00 CD 7B 08 00 45 00  00 73 79 4A 40 00 80 06  ...L.b.....{..E..syJ@...
[018] ED 98 C0 A8 85 17 C0 A8  8D 39 06 61 05 F1 FD F3  C2 33 30 86 60 E1 50 18  .........9.a.....30.`.P.
[030] FF FF 5A A6 00 00 00 4B  00 00 06 00 00 00 00 00  03 5E 00 02 80 21 00 01  ..Z....K.........^...!..
[048] 01 12 01 01 0D 00 00 00  00 04 7F FF FF FF 00 00  00 00 00 00 00 00 00 12  ........................
[060] 53 45 4C 45 43 54 20 2A  20 46 52 4F 4D 20 44 55  41 4C 01 01 00 00 00 00  SELECT * FROM DUAL......
[078] 00 00 01 01 00 00 00 00  00                                                .........

(Detected_Name : detect_select)
[12:12:54] ETH:0800 PROT:tcp SIP:192.168.155.115(00:0E:0C:00:CD:7B) SP:1627 -> DIP:192.168.161.33(00:E0:81:4C:D4:62) DP:1521 l
en:129 flag:...PA. ttl:128
[000] 00 E0 81 4C D4 62 00 0E  0C 00 CD 7B 08 00 45 00  00 73 7F 6F 40 00 80 06  ...L.b.....{..E..o@...
[018] BD 2F C0 A8 9B 73 C0 A8  A1 21 06 5B 05 F1 BF 5A  F3 CC 61 94 FC DA 50 18  ./...s...!.[...Z..a...P.
[030] FF FF 70 60 00 00 00 4B  00 00 06 00 00 00 00 00  03 5E 00 02 80 21 00 01  ..p`...K.........^...!..
[048] 01 12 01 01 0D 00 00 00  00 04 7F FF FF FF 00 00  00 00 00 00 00 00 00 12  ........................
[060] 53 45 4C 45 43 54 20 2A  20 46 52 4F 4D 20 44 55  41 4C 01 01 00 00 00 00  SELECT * FROM DUAL......
[078] 00 00 01 01 00 00 00 00  00                                                .........
```

주요 기능

행동 탐지


정책현황
활성화
비활성화
※각 정책의 이름을 클릭시 상세보기로 이동
추가
수정
삭제
No
이름
적용
위험등급
출발지IP
출발지Port
100001 detected_srcIp 하 192.168.1.65 any
100013 detected_to_srcIp 하 192.168.1.100-192.168.1.210 20000-60000
100014 detected_to_sp 하 any 20000-60000
100015 detected_dual 최상 any any
100016 detect_grants 상 any any
100020 detected_table 상 any any
100021 detected_5060 하 192.168.1.50-192.168.1.60 any
100063 detected_regex 하 any any
100068 jumin 하 any any
100069 detect_select 하 any any
100070 detected_insert 하 any any
100071 detected_update 하 any any
100072 detected_grant 하 any any

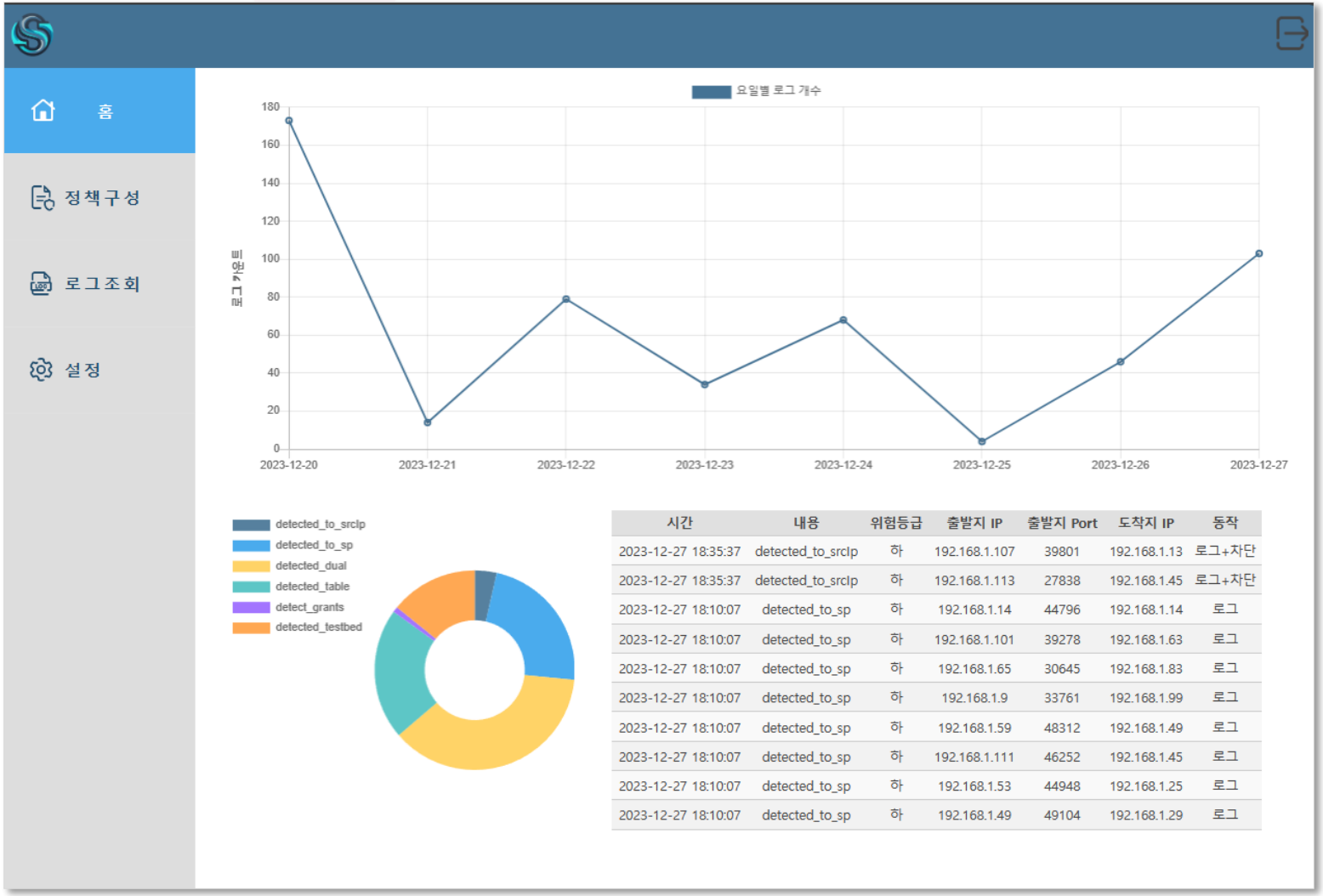

정책 상세 정보
탐지 정책 기본 정보
탐지명: detected_dual
유효기간: 2024-01-21
위험등급: 최상
세션 정보
출발지 IP: any
출발지 Port: any
정책 동작
동작설정: 로그+차단
탐지 패턴
기준횟수: 20 초 동안 10 회 발생
패턴정의
내 용: content를 입력하세요.
content1 : select * from dual
탐지 내용 설명
dual detected
취소
수정

주요 기능

행동 탐지

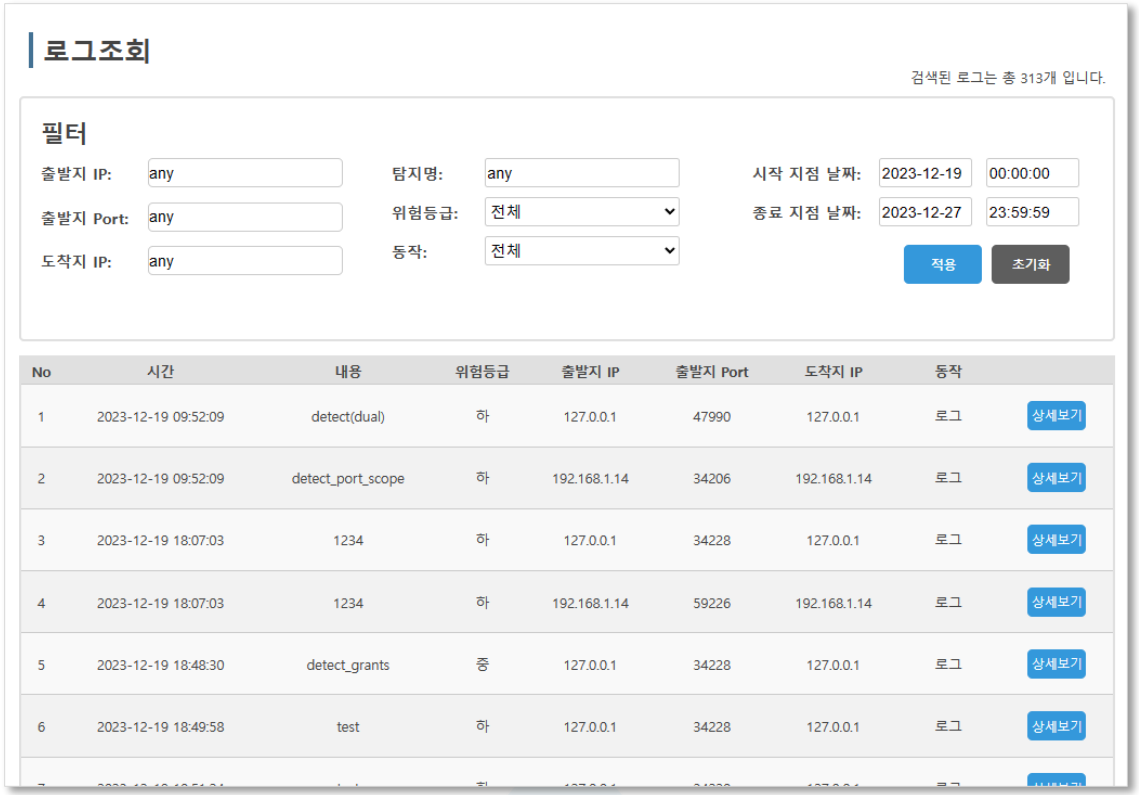
```
(Behavior_count : 10) (Detected_Name : detected_dual)
[12:21:07] ETH:0800 PROT:tcp SIP:192.168.1.65(50:B7:C3:A2:F6:88) SP:55889 -> DIP:192.168.1.14(08:00:27:43:0F:1A) DP:3306 len:9
0 flag:...PA. ttl:128
[000] 08 00 27 43 0F 1A 50 B7  C3 A2 F6 88 08 00 45 00  00 4C AF 1A 40 00 80 06  ..'C..P.......E..L..@...
[018] C7 F1 C0 A8 01 41 C0 A8  01 0E DA 51 0C EA BB 68  95 CA 86 5D 7E 6B 50 18  .....A.....Q...h...]~kP.
[030] 04 00 18 5C 00 00 20 00  00 00 03 73 65 6C 65 63  74 20 2A 20 66 72 6F 6D  ...\.. ....select * from
[048] 20 64 75 61 6C 0A 4C 49  4D 49 54 20 30 2C 20 32  30 30                    dual.LIMIT 0, 200
```

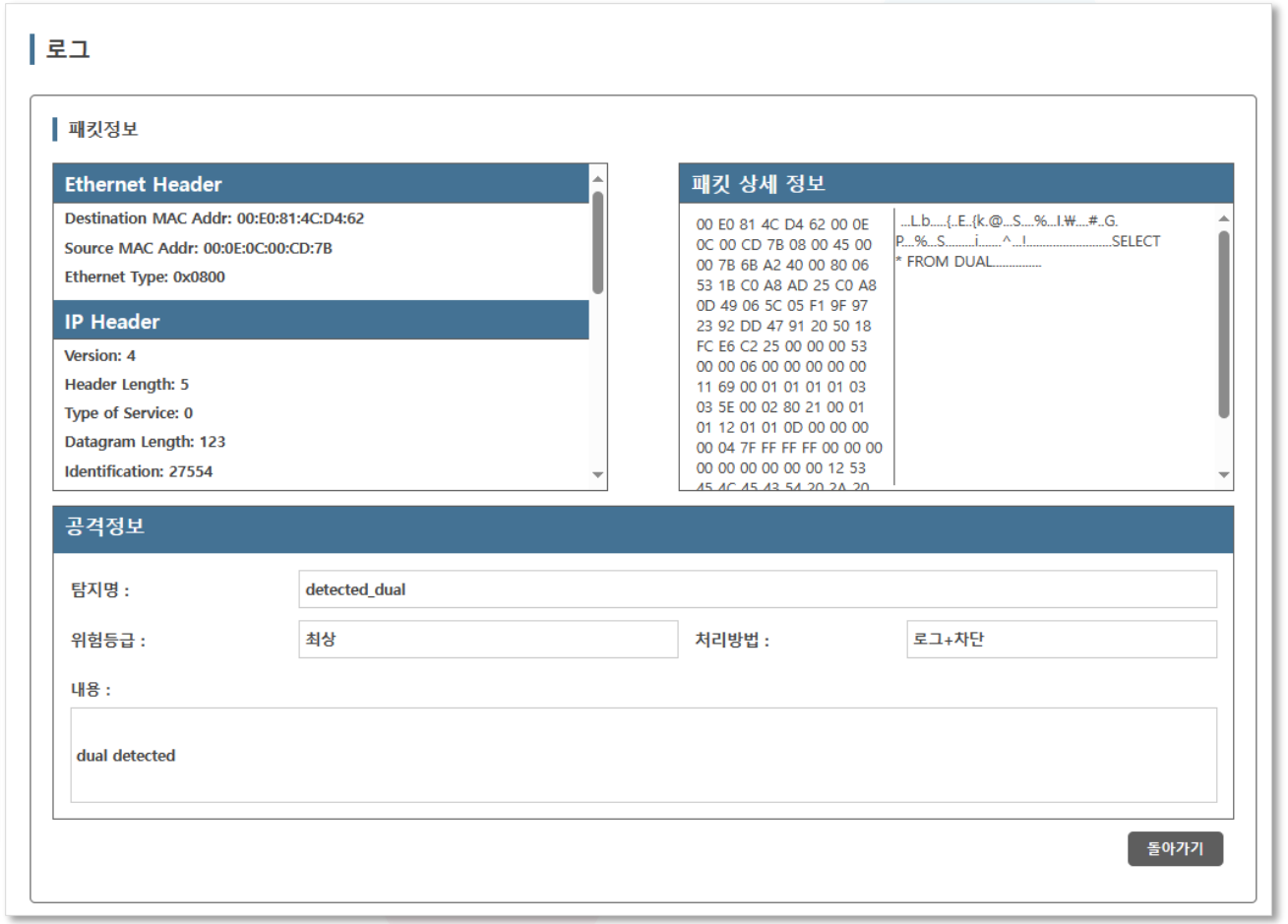
화면 구현



시간	내용	위험등급	출발지 IP	출발지 Port	도착지 IP	동작
2023-12-27 18:35:37	detected_to_srcIp	하	192.168.1.107	39801	192.168.1.13	로그+차단
2023-12-27 18:35:37	detected_to_srcIp	하	192.168.1.113	27838	192.168.1.45	로그+차단
2023-12-27 18:10:07	detected_to_sp	하	192.168.1.14	44796	192.168.1.14	로그
2023-12-27 18:10:07	detected_to_sp	하	192.168.1.101	39278	192.168.1.63	로그
2023-12-27 18:10:07	detected_to_sp	하	192.168.1.65	30645	192.168.1.83	로그
2023-12-27 18:10:07	detected_to_sp	하	192.168.1.9	33761	192.168.1.99	로그
2023-12-27 18:10:07	detected_to_sp	하	192.168.1.59	48312	192.168.1.49	로그
2023-12-27 18:10:07	detected_to_sp	하	192.168.1.111	46252	192.168.1.45	로그
2023-12-27 18:10:07	detected_to_sp	하	192.168.1.53	44948	192.168.1.25	로그
2023-12-27 18:10:07	detected_to_sp	하	192.168.1.49	49104	192.168.1.29	로그

화면 구현


로그조회
검색된 로그는 총 313개 입니다.
필터
출발지 IP: any
출발지 Port: any
도착지 IP: any
탐지명: any
위험등급: 전체
동작: 전체
시작 지점 날짜: 2023-12-19 00:00:00
종료 지점 날짜: 2023-12-27 23:59:59
적용
초기화
No 시간 내용 위험등급 출발지 IP 출발지 Port 도착지 IP 동작
1 2023-12-19 09:52:09 detect(dual) 하 127.0.0.1 47990 127.0.0.1 로그 상세보기
2 2023-12-19 09:52:09 detect_port_scope 하 192.168.1.14 34206 192.168.1.14 로그 상세보기
3 2023-12-19 18:07:03 1234 하 127.0.0.1 34228 127.0.0.1 로그 상세보기
4 2023-12-19 18:07:03 1234 하 192.168.1.14 59226 192.168.1.14 로그 상세보기
5 2023-12-19 18:48:30 detect_grants 중 127.0.0.1 34228 127.0.0.1 로그 상세보기
6 2023-12-19 18:49:58 test 하 127.0.0.1 34228 127.0.0.1 로그 상세보기


로그
패킷정보
Ethernet Header
Destination MAC Addr: 00:E0:81:4C:D4:62
Source MAC Addr: 00:0E:0C:00:CD:7B
Ethernet Type: 0x0800
IP Header
Version: 4
Header Length: 5
Type of Service: 0
Datagram Length: 123
Identification: 27554
패킷 상세 정보
00 E0 81 4C D4 62 00 0E
0C 00 CD 7B 08 00 45 00
00 7B 6B A2 40 00 80 06
53 1B C0 A8 AD 25 C0 A8
0D 49 06 5C 05 F1 9F 97
23 92 DD 47 91 20 50 18
FC E6 C2 25 00 00 00 53
00 00 06 00 00 00 00 00
11 69 00 01 01 01 01 03
03 5E 00 02 80 21 00 01
01 12 01 01 0D 00 00 00
00 04 7F FF FF FF 00 00 00
00 00 00 00 00 00 12 53
* FROM DUAL
공격정보
탐지명 : detected_dual
위험등급 : 최상
처리방법 : 로그+차단
내용 :
dual detected
돌아가기

프로젝트 이슈

1. 패턴 탐지 문제

패턴을 탐지함에 있어 패킷의 들어오는 데이터를 그대로 비교할 경우 대문자,소문자의 차이나 공백의 차이 등으로 인해 룰에 있는 Content를 탐지하지 못하는 경우가 발생하였습니다.

따라서 payload의 시작점을 잡지 못하여 payload의 시작점이 같지 않으므로 OFFSET을 따로 설정하여 payload를 가져와서 기호와 공백을 가공하여 정책에서 설정한 content와 비교하여 패턴을 탐지하였습니다.

```
int preBuildData(packet_t *p, u_char *pPacket, int nDataSize, int nOffset)
{
    int i;
    u_char *nocase;
    nocase = p->nocase;

    if( !p || !pPacket || nDataSize == 0 )
        return -1;

    for (i = 0 ; i < nDataSize; i++)
    {
        if( pPacket[i+nOffset] >= 127 || pPacket[i+nOffset] < ' ' ){
            *(nocase+i) = ' ';
            continue;
        }
        *(nocase+i) = pPacket[i+nOffset];
        // 알파벳 전부 소문자로
        if( *(nocase+i) >= 65 && *(nocase+i) <= 90 ){
            *(nocase+i) = *(nocase+i) + 32;
        }
    }
    *(nocase+nDataSize) = '\0';

    // 연속적인 공백을 하나의 공백으로 바꿉니다.
    for( i = 0 ; i < nDataSize ; i++ ){
        if( *(nocase+i) == ' ' && *(nocase+(i+1)) == ' '){
            for(int j = 0 ; j < nDataSize ; j++){
                *(nocase+(i+j)) = *(nocase+(i+1+j));
            }
            i--;
        }
    }

    return 0;
}
```

프로젝트 이슈

2. Logging 방식에 대한 문제

로깅 방식에 있어서 처음에는 탐지가 되었을 때 그 즉시 로그DB에 insert 쿼리를 전송하였습니다. 하지만 이러한 방식은 DB에 쓰여지는 동안 또 다른 패킷이 탐지 되지 못하고 지나가는 문제가 발생하였고, 로깅 부분을 Multi Thread로 구성하여 이 문제를 해결하였습니다.

```
void* log_insert(void* element){

    pthread_mutex_t log_mutex = PTHREAD_MUTEX_INITIALIZER;

    while(1){
        //  큐에 저장된 것이 없으면 대기
        if( logs.is_empty_logQueue() ){
            sleep(1);
            continue;
        }
        //  큐에 저장된 것이 있으면 log 쿼리 날리기
        pthread_mutex_lock(&log_mutex);
        logs.logDequeue();
        pthread_mutex_unlock(&log_mutex);
    }

}
```

```
//  룰 중에 content가 없는 IP와 PORT를 먼저 비교하여 차단
ruleIndex = rules.sessionFilter(p);
if ( ruleIndex != -1 && p->flow == 1 ){
    if(rules.is_check_matchSession(p, ruleIndex) ){
        return ACTION_PASS;
    }
    printf("(Detected_Name : %s) ", (rules.getRule(ruleIndex))->deName);
    logs.logEnqueue(packet, p, ruleIndex);
    return ACTION_LOG;
}

preBuildData(p, packet, p->dsize, p->caplen - p->dsize);

//  룰 매칭 확인
ruleIndex = rules.ruleFilter(p, p->nocase);
if ( ruleIndex != -1 && p->flow == 1){
    if( rules.is_check_matchSession(p, ruleIndex) ){
        return ACTION_PASS;
    }
    rule_t* match = rules.getRule(ruleIndex);
    printf("(Detected_Name : %s) ", match->deName);
    logs.logEnqueue(packet, p, ruleIndex);
    return ACTION_LOG;
}
```

프로젝트 이슈(관리자 UI)

3. 로그인 시 보안 문제

초기에는 관리자의 로그인 정보를 평문으로 저장하고 있었으나, 보안 및 사용자 정보 안정성이 부족하다고 판단하여 비밀번호와 무작위 솔트값을 해시화하여 저장하고 비교하도록 구현했습니다.

```java
/**
 * 회원 로그인 처리 메서드
 *
 * @param memberEntity 로그인 시도한 회원 엔티티
 * @return 로그인 성공 여부 (true: 성공, false: 실패)
 * @throws NoSuchAlgorithmException 암호화 알고리즘이 지원되지 않는 경우 발생
 */
public boolean login(MemberEntity memberEntity) throws NoSuchAlgorithmException {
    MemberEntity member = memberMapper.signin(memberEntity.getId());
    if (member != null) {
        String sha256hex = hashPassword(memberEntity.getPassword(), member.getSalt());
        if (member.getPassword().equals(sha256hex)) {
            return true;
        }
    }
    return false;
}
```

```java
/**
 * 비밀번호를 해시하는 메서드
 *
 * @param password 사용자가 입력한 비밀번호
 * @param salt     무작위값 (비밀번호에 추가되는 무작위 문자열)
 * @return 해시된 비밀번호
 * @throws NoSuchAlgorithmException 암호화 알고리즘이 지원되지 않는 경우 발생
 */
private String hashPassword(String password, String salt) throws NoSuchAlgorithmException {
    MessageDigest digest = MessageDigest.getInstance("SHA-256");
    byte[] hash = digest.digest((password + salt).getBytes(StandardCharsets.UTF_8));
    return String.format("%064x", new BigInteger(1, hash));
}
```

프로젝트 이슈(관리자 UI)

4. 유효기간에 관한 이슈

사용자가 설정한 정책의 유효기간이 끝났을 때, 이를 시스템이 자동 감지하고 반응하도록 만들기 위한 고민이 있었습니다. 이 문제를 해결하기 위해서, 유효기간이 만료되는 시점을 스스로 인지하고 해당 정책을 비활성화하는 스케줄링 기능을 도입하였습니다. 이를 통해 시스템은 사용자의 정책 유효기간 종료를 자동 인지하고 적절히 대응할 수 있게 되었습니다.

```java
/**
 * 주기적으로 실행되는 메서드로, 정책 업데이트를 수행하여 활성화 상태를 업데이트합니다.
 */
@Scheduled(cron = "0 0 0 * * *")
public void updateEnableStatus() {
    updatePolicyMapper.updateEnableStatus();
    triggerPolicyUpdate();
}

/**
 * 정책 업데이트를 트리거하여 정책이 업데이트되었음을 표시하는 메서드
 */
public void triggerPolicyUpdate() {
    this.isPolicyUpdated = true;
}

/**
 * 주기적으로 실행되며 정책의 유효기간이 지나 비활성화되었을 때 UDP 신호를 전송하는 메서드
 * 업데이트 후 플래그를 다시 false로 설정하여 중복 전송을 방지합니다.
 *
 * @throws IOException UDP 신호 전송 중 발생할 수 있는 입출력 예외입니다.
 */
@Scheduled(fixedDelay = 10000) // 10초마다 실행
public void processPolicyUpdate() throws IOException {
    if (isPolicyUpdated) {
        sendUDP();
        isPolicyUpdated = false; // 플래그를 다시 false로 설정
    }
}
```

프로젝트 이슈(관리자 UI)

5. 대용량 데이터 로딩 속도 문제

대량의 데이터 출력할 때 페이지 로딩 시간이 오래 걸리는 현상이 나타나 대용량 데이터를 고속으로 처리 가능한 방법들을 알아보았을 때 페이징, 캐싱, 백그라운드 작업 등이 있었고, 이중에서 페이징 기술을 적용한 결과 전에는 약 10초 가량 소요되던 로딩 시간이 페이징 처리 후에는 3초 이내로 단축되었습니다.

```
function leadLogs(currentPage) {
    $.ajax({
        url: '/admin/menu/readLogs/api/getLogList',
        method: 'GET',
        data: $('form').serialize() + '&page=' + currentPage + '&numPerPage=' + numPerPage,
        success: function (data) {
            $('#logTable').empty();

            var totalLogCount = data.totalLogCount;
            var logs = data.readLogs;

            $('.log-count').text('검색된 로그는 총  ' + totalLogCount + '개 입니다.');
```

```
List<ReadLogsEntity> readLogs = allLogs.stream()
        .sorted(Comparator.comparing(ReadLogsEntity::getTime).thenComparing(ReadLogsEntity::getLog_index))
        .skip((long) (page - 1) * numPerPage)
        .limit(numPerPage)
        .collect(Collectors.toList());
readLogs.forEach(ReadLogsEntity::setLogdateFromTime);

int totalLogCount = allLogs.size();
```

앞으로 구현할 기능

행동탐지(포트 스캔)

행동탐지(DDOS)

행동탐지(Flooding)

03 프로그램 사용기술

프로젝트 개발 도구

MariaDB

GDB
The GNU Project Debugger

jQuery

Gradle

MyBatis

Thymeleaf

Lombok

프로젝트 개발 환경

프로젝트 사용 언어

Q&A

"어떤 바보라도 컴퓨터를 사용할 수 있다.
그래서 많은 사람들이 컴퓨터를 사용한다."
Ted Nelson (HTML을 만든 과학자)

Sentinel