닉스테크
서울시 성동구 왕십리로 58 포휴빌딩 10 층
전화 02-3497-8900 **팩스** 02-578-6621

NICS TECH
Network Integrated Computing Service Provider

EDR AGENT 상세 설계서

VERSION	**0.0.3**
DATE	**2018-03-16**
WRITER	**장욱**

DOCUMENT HISTORY (문서정보/수정내역)

수정날짜	수정자	버전	추가/수정항목	내용
2018-02-05	장욱	0.0.1	초안 작성	초안 작성
2018-03-15	송민섭/장욱	0.0.2	전체 아키텍쳐 변경	아키텍쳐 변경에 따른 문서 수정
2018-03-16	송민섭/장욱	0.0.3	항목 추가	정적 정보 전송 추가, Controller 추가

1. CONTENTS (목차)

1. CONTENTS (목차) 3
2. INTRODUCTION (개요) 5

- Purpose (목표) 5
- Product Scope (범위) 5
- Terms and Abbreviations (정의 및 약어) 5

3. OVERALL DESCRIPTION (전체 설명) 6

- Product Perspect (제품 조망) 6
- Overall System Configuration (전체 시스템 구성) 6
  - SafePC (SMSS, MSUpddateService, AutoClientUpdate) 6
  - NicsWS_Interface 7
  - Push_Listener 7
  - Log_Shipper 7
  - Policy_Manager 7
  - Controller (제어기) 7
- Overall Operation (전체 동작 방식) 7
  - SafePC (SMSS) 7
    1) Basic Requirements (기본 요구 사항) 7
    2) Sequence Diagram (시퀀스 다이어그램) 8
    3) Data Struct & Function 8

    표. 3-1 정책 적용 결과 응답 NICSWS 인터페이스 8
    표. 3-2 스타트 타입 (등록된 서비스 항목) 8
    표. 3-3 자산 정보 (메타 정보) 8
    표. 3-4 로그 포맷 9
  - Push_Manager 11
    1) Basic Requirements (기본 요구 사항) 11
    2) Sequence Diagram (시퀀스 다이어그램) 11
    3) Data Struct & Function 11

    표. 3-5 Push_Manager 대응 정책 코드 11
    표. 3-6 Push_Manager 콜백 함수 원형 11
    표. 3-7 Push_Manager 외부 함수 12
    표. 3-8 Push_Manager 내부 함수 12
  - Policy_Manager 12
    1) Basic Requirements (기본 요구 사항) 12
    2) Sequence Diagram (시퀀스 다이어그램) 12
    3) Data Struct & Function 14

표. 3-9 Policy_Manager 정책 코드 14
표. 3-10 정책 수신 요청 NICSWS 인터페이스 14
표. 3-11 수신 정책 포맷 14
표. 3-12 정책 적용 결과 응답 NICSWS 인터페이스 15
표. 3-13 정책 적용 결과 응답 포맷 15
표. 3-14 내부 자료 구조 (설명을 돕기 위해 C/C++ 스타일로 서술) 16
표. 3-15 사용자 정보 (설명을 돕기 위해 C/C++ 스타일로 서술) 17
표. 3-16 Policy_Manager 외부 함수 17
표. 3-17 Policy_Manager 내부 함수 17

- Log_Shipper 21

1) Basic Requirements (기본 요구 사항) 21
2) Sequence Diagram (시퀀스 다이어그램) 21
3) Data Struct & Function 22

표. 3-18 Log_Shipper 이벤트 아이디 (종류) 22
표. 3-19 이벤트 아이디별 로그 포맷 23
표. 3-20 Log_Shipper 내부 함수 25

- Controller 26

1) Basic Requirements (기본 요구 사항) 26
2) Sequence Diagram (시퀀스 다이어그램) 26
3) Data Struct & Function 26

2. INTRODUCTION (개요)

■ Purpose (목표)

- 본 문서는 EDR Agent 의 상세 설계를 목표로 한다.
- 본 문서는 EDR 의 타 Block (SafePC Server, SOC-plus, ADS-plus) 과의 연동 방식을 제공한다.
- 본 문서는 타 Block 과의 연동에 필요한 Data Format 을 제공한다.
- 본 문서는 EDR Agent 내부에서 사용하는 상세 기능을 명세 한다.

■ Product Scope (범위)

- EDR Agent 는 SafePC 의 Sub Module 또는 plug-in 형태로 동작 한다.
- EDR Agent 가 제공하는 기능은 다음과 같다

  ① EDR 정책 수신/적용 기능 (정기/실시간/Known Signature)
  ② EDR Agent 를 통한 시스템 모니터링
  ③ EDR Agent 로그 취합 (필터링 포함) 기능
  ④ EDR Agent 를 통한 대응 정책 수신/적용 (실시간/Unknown Signature)
  ⑤ EDR Agent 자체 보호
  ⑥ 운영 및 보안 감사 로그

■ Terms and Abbreviations (정의 및 약어)

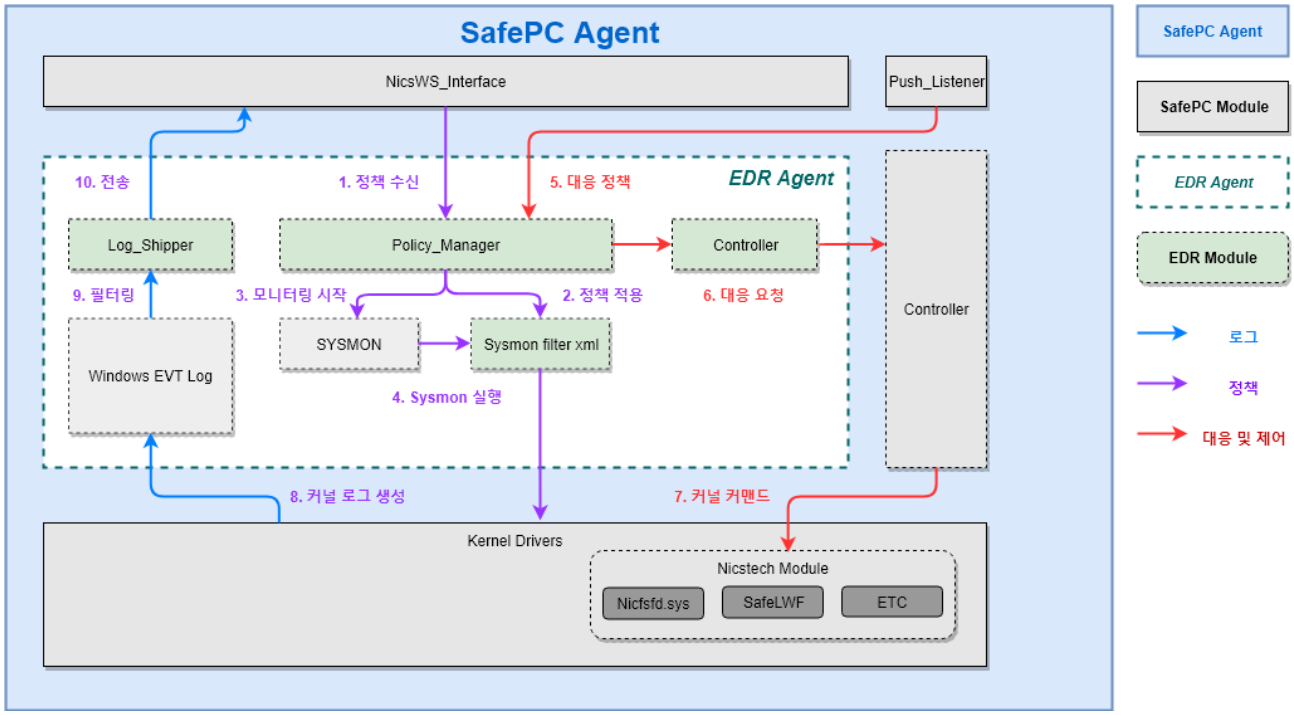
용어	설명
EDR	Endpoint Detection & Response

3. OVERALL DESCRIPTION (전체 설명)

■ Product Perspect (제품 조망)

- 기존의 SafePC 는 Main Platform 으로 동작한다.
- 시스템 모니터링 및 로그 생성은 Microsoft sysinternal Sysmon 을 통해 이루어 진다.
- EDR 의 주요 기능은 다음과 같다

① 시스템 모니터링 : 프로세스, 파일, 레지스트리, 네트워크, Admin API 사용 내역
② 모니터링 로그 전송 : Log shipper 모듈을 통해 시스템 모니터링 로그를 ADS/SOC ES 로 전송
③ 대응 로그 전송 : 대응 정책에 따라 차단된 리소스의 정책을 SafePC 서버로 전송
④ 대응/통제 (매체) : 기존 SafePC 의 매체 제어 기능을 통해 Isolate/Block 수행.
⑤ 대응/통제 (네트워크) : 기존 SafeNAC 제어 기능을 통해 Isolate/Block 수행
⑥ 정책 및 로그 송/수신 : NicsWS 인터페이스를 통해 송/수신



[그림 3-1 EDR Agent 구성]

■ Overall System Configuration (전체 시스템 구성)

● SafePC (SMSS, MSUpddateService, AutoClientUpdate)

- 자산 및 사용자 정보를 NicsWS 인터페이스를 통해 서버로 전송한다.
- SMSS 의 패치 모듈을 통해 EDR Agent 의 설치, 업그레이드를 진행한다.

- **NicsWS_Interface**

- 기존 SafePC 의 NicsWS 인터페이스 모듈을 사용한다.
- RESTful API 를 통해 일반 정책 및 대응 정책을 수신한다.
- RESTful API 를 통해 시스템 모니터링 로그를 송신 한다.

- **Push_Listener**

- 서버로부터 실시간 대응 정책 Push 를 수신한다.
- Push 수신 후 NicWS 인터페이스를 통해 대응 정책을 수신 후 적용한다.
- 대응 정책 수신 후 Controller 에게 대응 요청을 보낸다.

- **Log_Shipper**

- Windows EVT Log 로부터 로그를 취합한다. (실시간 or 주기)
- 1 차 가공 (북마크, 압축) 을 통해 전송 데이터를 준비 한다.
- NicsWS_Interface 를 통해 로그를 전송한다.

- **Policy_Manager**

- NicsWS 인터페이스를 통해 IOC 정책을 수신한다.
- NicsWS 인터페이스를 통해 대응 정책을 수신한다.
- NicsWS 인터페이스를 통해 Sysmon Filter 정책을 수신 한다.
- 수신한 각 정책을 내부 포맷에 맞게 변환 한다.
- IOC 및 대응 정책을 각 Controller 를 통해 Kernel Driver 에 적용한다.
- Sysmon 설치 및 구동을 실행한다.

- **Controller (제어기)**

- Policy_Manger 로부터 대응 정책을 전달 받아 커널 커맨드로 변환한다.
- 변환한 커널 커맨드를 각 커널 드라이브에 전달 한다.

■ Overall Operation (전체 동작 방식)

- **SafePC (SMSS)**

1) Basic Requirements (기본 요구 사항)

- 사용자별로 PC 기반으로 정적 정보 (자산 정보)를 서버로 전송한다.
- NicsWS 인터페이스를 통해 정보를 전송 한다.
- 정적 정보 전송 주기는 SafePC 정책 전송 주기에 따른다.
- 정적 정보는 다음과 같다.

① 운영체제 정보
② 설치된 소프트웨어 정보
③ 네트워크 카드 정보
④ 윈도우 계정 정보
⑤ 시작 프로그램 정보
⑥ 컴퓨터 on/off 이력 정보

2) Sequence Diagram (시퀀스 다이어그램)

- 별도 기재 하지 않음

3) Data Struct & Function

① 정책 전송 NICSWS 인터페이스

표. 3-1 정책 적용 결과 응답 NICSWS 인터페이스		
운영체제 정보 /설치된 소프트웨어 정보	**POST**	**https://{DOMAIN}/nicsws/p4/spe/3000**
등록된 서비스 정보	**POST**	**https://{DOMAIN}/nicsws/p4/spe/3000**
네트워크 카드 정보	**POST**	**https://{DOMAIN}/nicsws/p4/spe/3000**
윈도우 계정 정보	**POST**	**https://{DOMAIN}/nicsws/p4/spe/3000**
시작 프로그램 정보	**POST**	**https://{DOMAIN}/nicsws/p4/spe/3000**
컴퓨터 on/off 이력 정보	**POST**	**https://{DOMAIN}/nicsws/p4/spe/3000**

② 스타트 타입 (등록된 서비스 정보 항목)

표. 3-2 스타트 타입 (등록된 서비스 항목)	
0	**SERVICE_BOOT_START**
1	**SERVICE_SYSTEM_START**
2	**자동 (SERVICE_AUTO_START)**
3	**수동 (SERVICE_DEMAND_START)**
4	**사용 안 함 (SERVICE_DISABLED)**

③ 메타 정보

표. 3-3 자산 정보 (메타 정보)		
persion_id	String	사용자 ID
mac	String	MAC 주소
user_name	String	사용자명
host_name	String	호스트명
asset_type	Int	자산타입
assets	Array<Object>	자산정보

④ 로그 포맷

표. 3-4 로그 포맷

운영체체 정보/설치된 소프트웨어 정보

```
{
  "person_id":"dummy_id",
  "mac":"00:00:00:00:00:00",
  "user_name":"더미유저",
  "host_name":"dummy_host",
  "asset_type":2,
  "assets":
  [{
    "software":"Windows XP Professional",
    "dir":"C:\Windows",
    "version":"5.1"
  },
  {
    "software":"IPlus”
    "dir":"None",
    "version":"None"
  }]
}
```

등록된 서비스 정보

```
{
  "person_id":"dummy_id",
  "mac":"00:00:00:00:00:00",
  "user_name":"더미유저",
  "host_name":"dummy_host",
  "asset_type":268435456,
  "assets":
  [{
    "name":"AxInstSV",
    "display_name":"ActiveX Installer (AxInstSV)",
    "path":"C:\Windows\system32\svchost.exe -k AxInstSVGroup",
    "start_type":"3"
  },
  {
    "name":"Browser",
    "display_name":"Computer Browser",
    "path":"C:\Windows\System32\svchost.exe -k netsvcs",
    "start_type":"4"
  }]
}
```

네크워크 카드 정보

```
{
  "person_id":"dummy_id",
  "mac":"00:00:00:00:00:00",
  "user_name":"더미유저",
  "host_name":"dummy_host",
  "asset_type":536870912,
  "assets":
  [{
    "name":"",
    "Description":"",
    "mac":"",
    "iplist":
    [{
      "ip":"192.168.198.26",
      "subnet":"255.255.255.0"
    },
    {
      "ip":"192.168.198.26",
      "subnet":"255.255.255.0"
    }],
    "gateway":["192.168.99.254", "22.23.11.23"],
    "dns":["8.8.8.8", "123.123.123.123"],
    "dhcp":0
  },
  {
    "name":"",
    "Description":"",
    "mac":"",
    "iplist":
```

```
    [{
      "ip":"192.168.198.26",
      "subnet":"255.255.255.0"
    },
    {
      "ip":"192.168.198.26",
      "subnet":"255.255.255.0"
    }],
    "gateway":["192.168.99.254", "22.23.11.23"],
    "dns":["8.8.8.8", "123.123.123.123"],
    "dhcp":0
  }]
}
```

윈도우 계정 정보

```
{
  "person_id":"dummy_id",
  "mac":"00:00:00:00:00:00",
  "user_name":"더미유저",
  "host_name":"dummy_host",
  "asset_type":1073741824,
  "assets":
  [{
    "name":"Administrator",
    "group":"Administrators",
    "enabled":0
  },
  {
    "name":"user",
    "group":"Administrators",
    "enabled":1
  }]
}
```

시작 프로그램 정보

```
{
  "person_id":"dummy_id",
  "mac":"00:00:00:00:00:00",
  "user_name":"더미유저",
  "host_name":"dummy_host",
  "asset_type":1073741824,
  "assets":
  [{
    "name":"NicdTest 응용 프로그램",
    "command":"C:\SafePCTool\NicdTest.exe",
    "location":"C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup"
  },
  {
    "name":"VMware Tools",
    "command":"C:\Program Files\VMware\VMware Tools\VMwareTray.exe",
    "location":"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
  }]
}
```

컴퓨터 On/Off 이력 정보

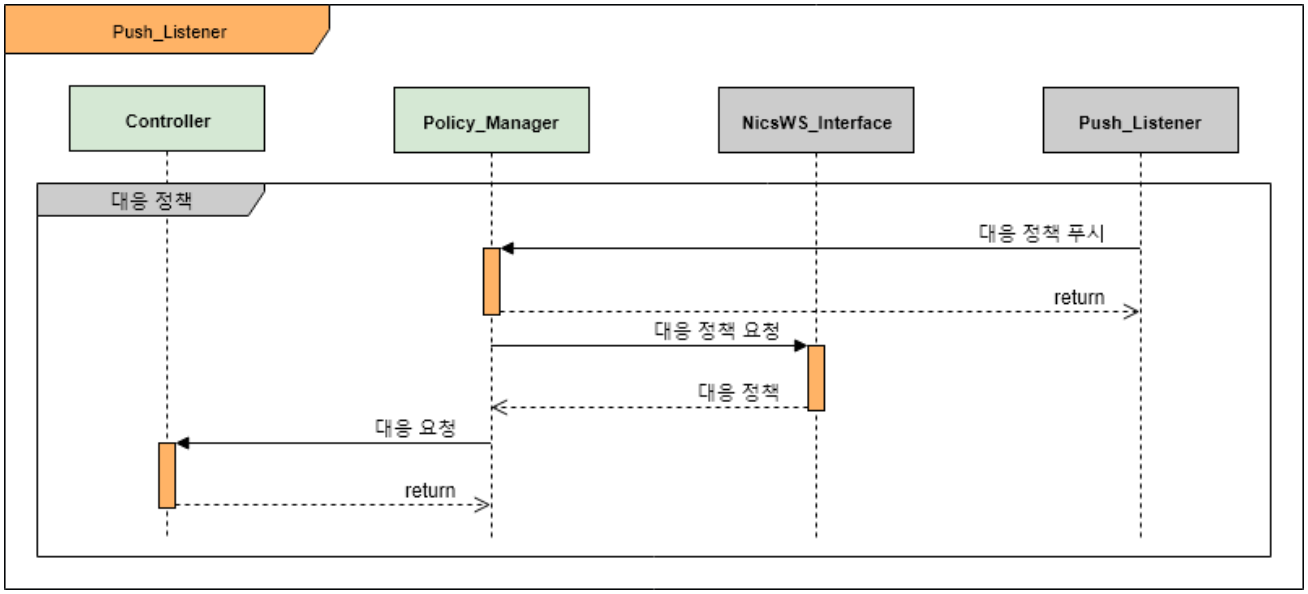
```
{
   "type" : "1",
   "time" : "2018-03-13 08:35:30",
   "id” : "mssong",
   "ip" : "192.168.99.34",
   "mac" : "00:00:00:00:00:00",
   "result" : "START",
}

{
   "type" : "2",
   "time" : "2018-03-13 18:02:10",
   "id” : "mssong",
   "ip" : "192.168.99.34",
   "mac" : "00:00:00:00:00:00",
   "result" : "END",
}
```

- **Push_Manager**

1) Basic Requirements (기본 요구 사항)

- 서버로부터 실시간 푸시 이벤트를 수신할 수 있어야 한다.
- 푸시 이벤트 수신 시 Log_Manager 에게 푸시 이벤트 를 전달한다
- Policy_Manger 에 의해 콜백 함수 등록이 가능해야 한다.
- Policy_Manger 에 의해 등록된 콜백 함수를 통해 푸시 이벤트를 알린다.

2) Sequence Diagram (시퀀스 다이어그램)



[그림 3-2 Push_Manager]

3) Data Struct & Function

① 대응 정책 코드

표. 3-5 Push_Manager 대응 정책 코드	
1000	프로세스 실시간 차단
1010	파일차단
1020	레지스트리 삭제
1030	네트워크 격리/차단

② 콜백 함수 원형

표. 3-6 Push_Manager 콜백 함수 원형	
Function Name	**PushCallbackFunc**
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN INT32 iPushCode) (표.3-5 참고)
Description	Push_Listener 가 호출하는 Policy_Manager 의 콜백 인터페이스

ETC	

③ 외부 함수

표. 3-7 Push_Manager 외부 함수	
Function Name	SetPushCallback
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN VOID* pPushCallbackFunc)(표.3-6 참고)
Description	Push_Listener 에게 제공되는 콜백 인터페이스 설정 함수
ETC	

④ 내부 함수

표. 3-8 Push_Manager 내부 함수	
Function Name	NotifyPushEvent
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN INT32 iPushCode) (표.3-5 참고)
Description	Policy_Manager 에게 Push Event 를 알린다
ETC	

- **Policy_Manager**

1) Basic Requirements (기본 요구 사항)

- 최초 제품 설치 시 기본 정책이 존재 해야 한다 (패키지 내 포함)
- 데이터 요청은 별도의 쓰레드를 통해 이루어 져야 하며, EDR Agent 의 다른 동작이 멈추어서는 안 된다.
- 수신된 정책 데이터는 File 또는 Registry 형태로 저장/유지 되어야 한다.
- 정책은 로그인 된 Agent 계정에 따라 저장 되어야 하며, 계정에 따라 저장된 정책을 적용 하여야 한다.
- 저장된 정책 데이터는 자체 보호 기능에 의해 보호 되어 야 한다. (자체보호 참고)
- 정책은 다음과 같이 분류 된다.

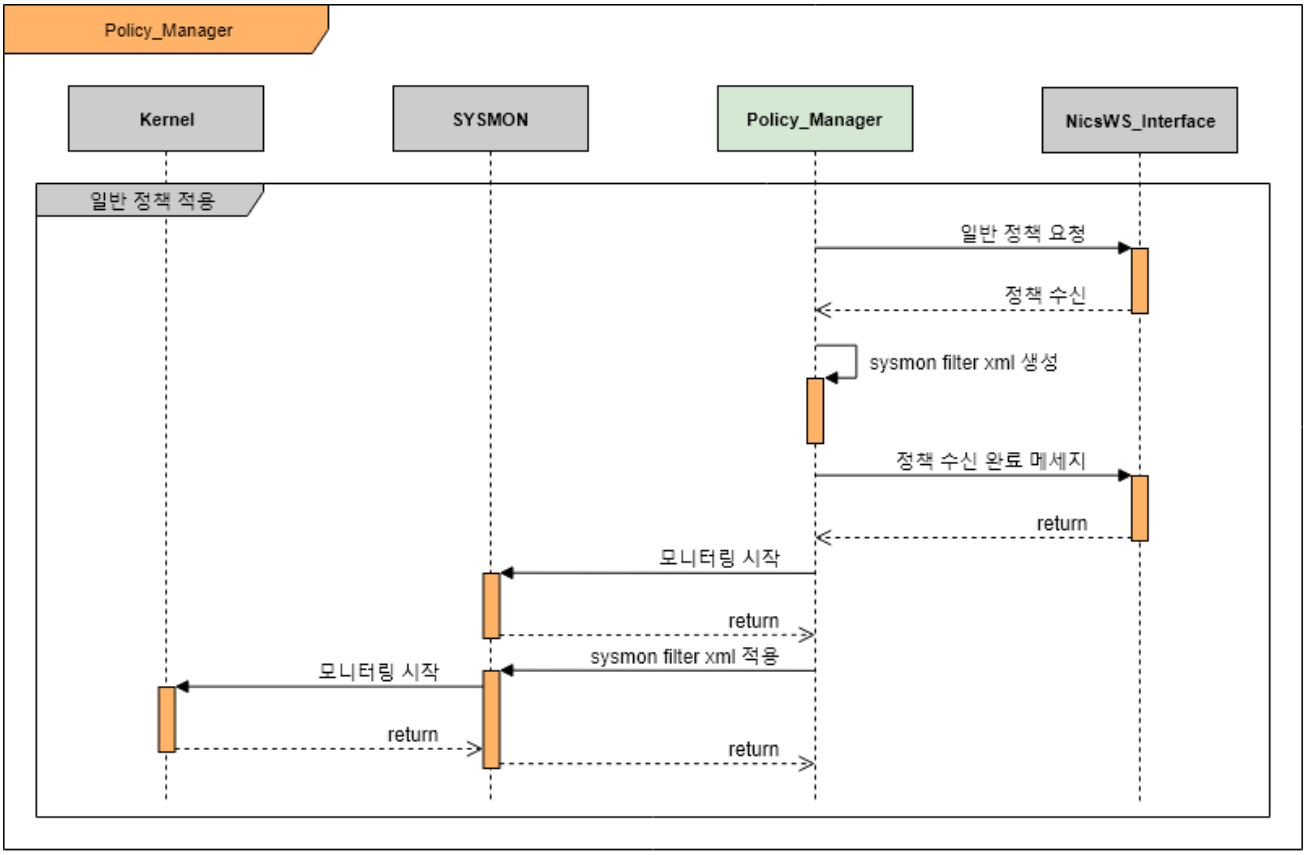
  ① IOC 정책 (Knonw Signature 를 통한 사전 차단 정책)
  ② 대응 정책 (M/L 을 통한 실시간 대응 정책)
  ③ 모니터링 필터 정책 (Sysmon 에 적용할 필터 정책)

- 다음 조건에서 저장된 정책을 적용한다.

  ① Policy Manager 가 다음 번 실행 (재부팅) 시
  ② EDR Agent 가 offline 상태일 때
  ③ 정책 수신에 실패 했을 때

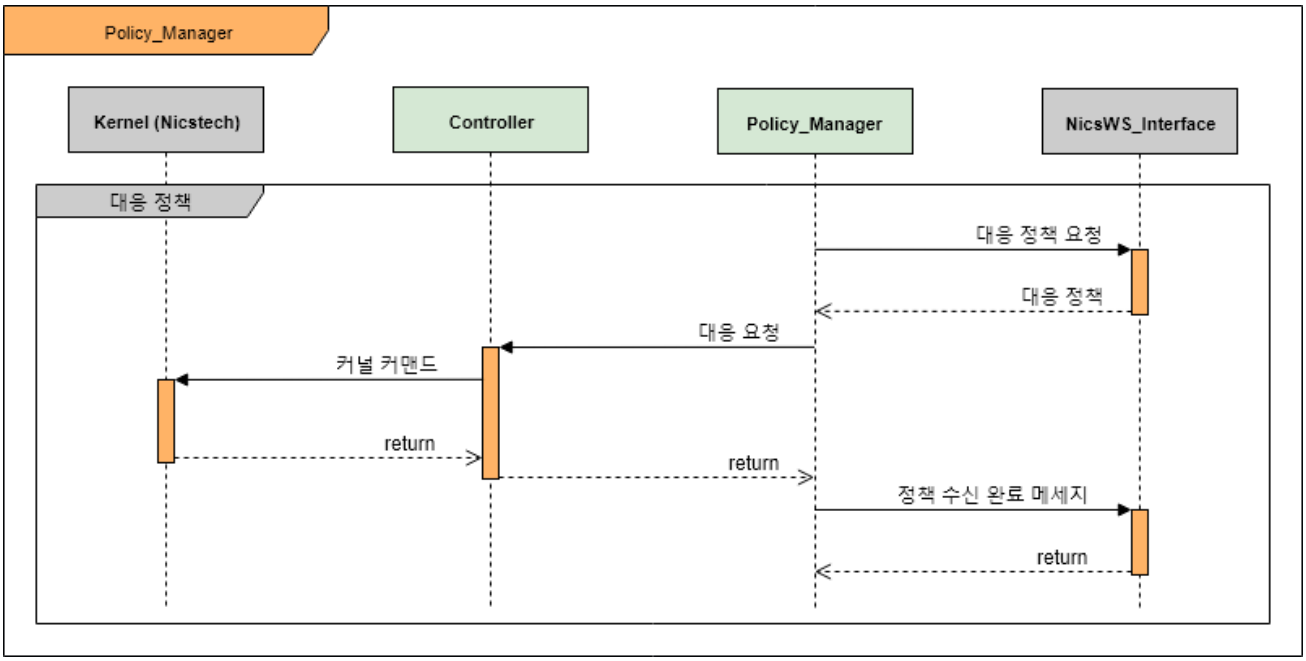
2) Sequence Diagram (시퀀스 다이어그램)

① 일반 정책 요청



[그림 3-3 Policy_Manager 일반 정책 요청]

② 대응 정책 요청



[그림 3-4 Policy_Manager 대응 정책 요청]

3) Data Struct & Function

① 정책 코드

표. 3-9 Policy_Manager 정책 코드	
1040	IOC 정책
1050	Sysmon Filter 정책

② 정책 수신 요청 NICWS 인터페이스

표. 3-10 정책 수신 요청 NICSWS 인터페이스		
프로세스 실시간 차단	GET	https://{DOMAIN}/nicsws/p5/edr/process/20/{{user_id}}
파일차단	GET	https://{DOMAIN}/nicsws/p5/edr/file/20/{{user_id}}
레지스트리 삭제	GET	https://{DOMAIN}/nicsws/p5/edr/reg/20/{{user_id}}
네트워크 격리/차단	GET	https://{DOMAIN}/nicsws/p5/edr/net/20/{{user_id}}

③ 수신 정책 포맷

표. 3-11 수신 정책 포맷	

프로세스 실시간 차단	{
	status: 401,
	detail_code: -1
	message: '요청자가 로그인 중이 아니거나 Agent 설치가 되어있지 않습니다',
	req_user_id: 'dev01',
	result_time: '2018.03.08 13:24:22.64',
	event_list: [
	{group: 'S00001', targets: [{cmd: 'kill',
	ProcessId: "57087",
	ParentProcessId: "24128",
	Hashes: "MD5=22819E...A7138",
	Image: "C:\Program Files\Mozilla Firefox\firefox.exe"},
	...,
	{...}]},
	{group: 'S00002', targets: [{...}, ..., {...}]},
	:
	:
	{group: 'S00099', targets: [{...}, ..., {...}]}
	]
	}

파일차단	{
	status: 401,
	detail_code: -1
	message: '요청자가 로그인 중이 아니거나 Agent 설치가 되어있지 않습니다',
	req_user_id: 'dev01',
	result_time: '2018.03.08 13:24:22.64',
	event_list: [
	{group: 'S00001', targets: ['C:\Program Files\notepad.exe',
	'C:\Program FIles\antivirus.exe',
	…,
	'C:\Program FIles\kakaotalk.exe']},
	{group: 'S00002', targets: [...]},
	:
	:
	{group: 'S00099', targets: [...]}
	]
	}

레지스트리 삭제	{ status: 401, detail_code: -1 message: '요청자가 로그인 중이 아니거나 Agent 설치가 되어있지 않습니다', req_user_id: 'dev01', result_time: '2018.03.08 13:24:22.64',

	event_list: [{group: 'S00001', targets: ['HKLM\\SOFTWARE\\Microsoft\\Windwos\\CurrentVersion\\Run\\key#1', 'HKLM\\SOFTWARE\\Microsoft\\Windwos\\CurrentVersion\\Run\\key#2', …, 'HKLM\\SOFTWARE\\Microsoft\\Windwos\\CurrentVersion\\Run\\key#3']}, {group: 'S00001', targets: [...]}, : : {group: 'S00001', targets: [...]}] }

네트워크 격리/차단	{
	status: 401,
	detail_code: -1
	message: '요청자가 로그인 중이 아니거나 Agent 설치가 되어있지 않습니다',
	req_user_id: 'dev01',
	result_time: '2018.03.08 13:24:22.64',
	event_list: [
	{group: 'S00001', targets: [{cmd: 'block',
	IP: 'XXX.XXX.XXX.XXX',
	Port: 8080,
	Hashes: "MD5=22819E...A7138",
	Bound: "out"},
	...,
	{...}]},
	{group: 'S00002', targets: [{...}, ..., {...}]},
	:
	:
	{group: 'S00099', targets: [{...}, ..., {...}]}
	]
	}

④ 정책 적용 결과 응답 NICWS 인터페이스

표. 3-12 정책 적용 결과 응답 NICSWS 인터페이스		
프로세스 실시간 차단	POST	https://{DOMAIN}/nicsws/p5/edr/process/20/{{user_id}}
파일차단	POST	https://{DOMAIN}/nicsws/p5/edr/file/20/{{user_id}}
레지스트리 삭제	POST	https://{DOMAIN}/nicsws/p5/edr/reg/20/{{user_id}}
네트워크 격리/차단	POST	https://{DOMAIN}/nicsws/p5/edr/net/20/{{user_id}}

⑤ 정책 적용 결과 응답 포맷

표. 3-13 정책 적용 결과 응답 포맷	

프로세스 실시간 차단	{
	status: 500,
	message: 'NUllPointerException: null 항목에 접근 할 수 없습니다',
	apply_time: '2018.03.08 13:24:22.64',
	complete_group_list: ['S00001', 'S00002', ..., 'S00099']
	}

파일차단	{
	status: 500,
	message: 'NUllPointerException: null 항목에 접근 할 수 없습니다',
	apply_time: '2018.03.08 13:24:22.64',
	complete_group_list: ['S00001', 'S00002', ..., 'S00099']
	}

레지스트리 삭제	{
	status: 500,
	message: 'NUllPointerException: null 항목에 접근 할 수 없습니다',
	apply_time: '2018.03.08 13:24:22.64',
	complete_group_list: ['S00001', 'S00002', ..., 'S00099']
	}

네트워크 격리/차단	{
	status: 500,
	message: 'NUllPointerException: null 항목에 접근 할 수 없습니다',
	apply_time: '2018.03.08 13:24:22.64',
	complete_group_list: ['S00001', 'S00002', ..., 'S00099']
	}

	}

⑥ 내부 자료 구조 (내부에 적용될 정책 포맷)

표. 3-14 내부 자료 구조 (설명을 돕기 위해 C/C++ 스타일로 서술)	

프로세스 실시간 차단	typedef struct _EDR_RES_PROCESS_POLICY_ENTRY_ { // 프로세스 차단 항목
	  CString strCmd; // 커맨드 라인
	  DWORD dwPID; // 프로세스 PID
	  DWORD dwPPID; // 부모 프로세스 PID
	  CString strHash; // 프로세스 Hash 값
	  CString strPath; // 프로세스 이미지 경로
	} EDR_RES_PROCESS_POLICY_ENTRY;
	
	typedef struct _PROCESS_GROUP_ENTRY_ { // 그룹 리스트 항목
	  CString strGroupID; // 그룹 아이디
	  DWROD dwPolicyCount; // 프로세스 차단 항목 개수
	  PROCESS_POLICY_ENTRY* stPolicyEntry; // 프로세스 차단 항목 리스트
	} EDR_RES_PROCESS_GROUP_ENTRY;
	
	typedef struct _PROCESS_POLICY_ { // 프로세스 차단 정책
	  CString strApplyTime; // 적용 시간
	  CString strUserID; // 사용자 아이디
	  DWORD dwGroupCount; // 그룹 리스트 개수
	  PROCESS_POLICY_ENTRY* stGroupPolicy; // 그룹 리스트
	} EDR_RES_PROCESS_POLICY;

파일차단	typedef struct _EDR_RES_FILE_POLICY_ENTRY_ { // 파일 차단 항목
	  CString strPath; // 파일 경로
	} EDR_RES_FILE_POLICY_ENTRY;
	
	typedef struct _FILE_GROUP_ENTRY_ { // 그룹 리스트 항목
	  CString strGroupID; // 그룹 아이디
	  DWROD dwPolicyCount; // 파일 차단 항목 개수
	  PROCESS_POLICY_ENTRY* stPolicyEntry; // 파일 차단 항목 리스트
	} EDR_RES_FILE_GROUP_ENTRY;
	
	typedef struct _FILE_POLICY_ { // 파일 차단 정책
	  CString strResultTime; // 적용 시간
	  CString strUserID; // 사용자 아이디
	  DWORD dwGroupCount; // 그룹 리스트 개수
	  PROCESS_POLICY_ENTRY* stGroupPolicy; // 그룹 리스트
	} EDR_RES_FILE_POLICY;

레지스트리 삭제	typedef struct _EDR_RES_REG_POLICY_ENTRY_ { // 레지스트리 차단 항목
	  CString strPath; // 레지스트리 경로
	} EDR_RES_REG_POLICY_ENTRY;
	
	typedef struct _REG_GROUP_ENTRY_ { // 그룹 리스트 항목
	  CString strGroupID; // 그룹 아이디
	  DWROD dwPolicyCount; // 레지스트리 차단 항목 개수
	  EDR_RES_REG_POLICY_ENTRY* stPolicyEntry; // 레지스트리 차단 항목 리스트
	} EDR_RES_REG_GROUP_ENTRY;
	
	typedef struct _EDR_RES_REG_POLICY_ { // 레지스트리 차단 정책
	  CString strResultTime; // 적용 시간
	  CString strUserID; // 사용자 아이디
	  DWORD dwGroupCount; // 그룹 리스트 개수
	  EDR_RES_REG_POLICY_ENTRY* stGroupPolicy; // 그룹 리스트
	} EDR_RES_REG_POLICY;

네트워크 격리/차단	typedef struct _EDR_RES_NET_POLICY_ENTRY_ { // 네크워크 차단 항목
	  CString strIP; // 아이피
	  CString strPort; // 포트
	  CString strHash // Hash 값
	} EDR_RES_NET_POLICY_ENTRY;
	
	typedef struct _EDR_RES_NET_GROUP_ENTRY_ { // 그룹 리스트 항목
	  CString strGroupID; // 그룹 아이디
	  DWROD dwPolicyCount; // 레지스트리 차단 항목 개수
	  EDR_RES_NET_POLICY_ENTRY* stPolicyEntry; // 레지스트리 차단 항목 리스트
	} EDR_RES_NET_GROUP_ENTRY;
	
	typedef struct _NET_POLICY_ { // 레지스트리 차단 정책
	  CString strResultTime; // 적용 시간
	  CString strUserID; // 사용자 아이디
	  DWORD dwGroupCount; // 그룹 리스트 개수
	  PROCESS_POLICY_ENTRY* stGroupPolicy; // 그룹 리스트
	} EDR_RES_NET_POLICY;

전체 대응 정책	typedef struct _EDR_RES_POLICY_ {
	  EDR_RES_PROCESS_POLICY* stEDR_RES_PROCESS; // 프로세스 대응 정책
	  EDR_RES_FILE_POLICY* stEDR_RES_FILE; // 파일 대응 정책
	  EDR_RES_REG_POLICY* stEDR_RES_REG; // 레지스트리 대응 정책
	  EDR_RES_NET_POLICY* stEDR_RES_NET; // 네트워크 대응 정책
	} EDR_RES_POLICY;

표. 3-15 사용자 정보 (설명을 돕기 위해 C/C++ 스타일로 서술)

사용자 정보	typedef struct _EDR_USER_INFO_ {
	  CString strCurrentUserInfo; // 현재 로그인된 사용자 정보
	  CString strCurrentUserName; // 현재 로그인된 사용다 이름
	  CString strHostName; // PC 호스트 명
	  CString strIP; // PC 아이피
	  CString strMAC; // PC 맥
	  CString strCurrentUserGroup; // 현재 로그인된 사용자의 그룹 정보
	  BOOL bIsInternal; // 내/외부 상태
	} EDR_USER_INFO;

⑦ 외부 함수

표. 3-16 Policy_Manager 외부 함수

Function Name	**PushCallbackFunc** (표. 3-2 Push_Manager 콜백 함수 원형 참고)
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN INT32 iPolicyCode) (표.3-5 참고)
Description	Push_Listener 가 호출하는 Policy_Manager 의 콜백 인터페이스
ETC	Push 코드에 따라 NICSWS 인터페이스를 통해 정책 수신을 요청한다.

⑧ 내부 함수

표. 3-17 Policy_Manager 내부 함수

Function Name	**InitPolicyManager**

Return type	BOOL
Argrument(방향, 타입, 이름)	(VOID)
Description	Policy_Manager 초기화 작업을 수행한다.
ETC	* 내부 자료구조 초기화 * Sysmon 설치 유무 판단 후 설치 진행 (서비스 등록) * SafePC 로부터 사용자 정보 획득 * 로컬에 저장된 정책 파일 적용

Function Name	StartPolicyManager
Return type	BOOL
Argrument	(VOID)
Description	Policy_Manager 를 시작한다
ETC	

Function Name	StopPolicyManager
Return type	BOOL
Argrument	(VOID)
Description	Policy_Manager 를 중지한다
ETC	

Function Name	GetUserInfo
Return type	BOOL
Argrument	(OUT EDR_USER_INFO* pEDRUserInfo)
Description	SafePC 로부터 현재 사용자의 정보를 획득한다.
ETC	

Function Name	CheckSysmonInstallStatus
Return type	BOOL
Argrument(방향, 타입, 이름)	(VOID)
Description	Sysmon 설치 상태를 체크한다.
ETC	

Function Name	InstallSysmonService
Return type	BOOL
Argrument(방향, 타입, 이름)	(VOID)
Description	Sysmon 을 설치하고 서비스 등록을 진행한다.
ETC	

Function Name	CheckLogShipperInstallStatus
Return type	BOOL

Argrument(방향, 타입, 이름)	(VOID)
Description	LogShipper 서비스 설치 상태를 체크한다.
ETC	

Function Name	InstallLogShipperService
Return type	BOOL
Argrument(방향, 타입, 이름)	(VOID)
Description	LogShipper 서비스를 설치하고 서비스 등록을 진행한다.
ETC	

Function Name	UpdateSysmonFilter
Return type	BOOL
Argrument(방향, 타입, 이름)	(CString& strSysmonFilter)
Description	Sysmon Filter 정책을 업데이트 한다
ETC	

Function Name	NotifyPolicyApplied
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN INT32 iPolicyCode)
Description	Sysmon 을 설치하고 서비스 등록을 진행한다.
ETC	

Function Name	StartMonitoring
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN BOOL status)
Description	Sysmon 모니터링을 시작(TRUE)/정지(FALST_한다
ETC	

Function Name	StartShipper
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN BOOL status)
Description	LogShipper 서비스 시작(TRUE)/정지(FALST_한다
ETC	

Function Name	ClearRESPolicyData
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN EDR_RES_POLICY* pEDRRESPolicy)
Description	현재 대응 정책을 지운다

ETC	정책 수신시 메모리 할당이 발생하므로 반드시 외부에서 메모리 해제를 해주어야 한다

Function Name	**ClearIOCPolicyData**
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN EDR_IOC_POLICY* pEDRIOCPolicy)
Description	현재 IOC 정책을 지운다
ETC	정책 수신 시 메모리 할당이 발생하므로 반드시 외부에서 메모리 해제를 해주어야 한다

Function Name	**RequestRESPolicyData**
Return type	EDR_RES_POLICY*
Argrument(방향, 타입, 이름)	(CString strUserName)
Description	현재 사용자에 대한 대응 정책을 요청한다.
ETC	내부에서 메모리 할당이 발생하므로 종료시 반드시 외부에서 ClearRESPolicyData 를 호출해 주어야 한다.

Function Name	**RequestIOCPolicyData**
Return type	EDR_IOC_POLICY*
Argrument(방향, 타입, 이름)	(CString strUserName)
Description	현재 사용자에 대한 IOC 정책을 요청한다.
ETC	내부에서 메모리 할당이 발생하므로 종료 시 반드시 외부에서 ClearIOCPolicyData 를 호출해 주어야 한다.

Function Name	**ConvertRESPolicyJsonToPolicyData**
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN CString& strPolicyData OUT EDR_RES_POLICY* pEDRRESPolicy)
Description	수신된 대응 정책을 내부 포맷으로 변환 한다.
ETC	

Function Name	**ConvertIOCPolicyJsonToPolicyData**
Return type	BOOL
Argrument(방향, 타입, 이름)	(IN CString& strPolicyData OUT EDR_IOC_POLICY* pEDRIOCPolicy)
Description	수신된 IOC 정책을 내부 포맷으로 변환 한다.
ETC	

Function Name	**ApplyRESPolicy**
Return type	BOOL

Argrument(방향, 타입, 이름)	(EDR_IOC_POLICY* pEDRRESPolicy)
Description	현재 사용자에 대한 대응 정책을 커널로 내린다
ETC	

Function Name	ApplyIOCPolicy
Return type	EDR_IOC_POLICY*
Argrument(방향, 타입, 이름)	(EDR_IOC_POLICY* pEDRIOCPolicy)
Description	현재 사용자에 대한 IOC 정책을 커널로 내린다
ETC	

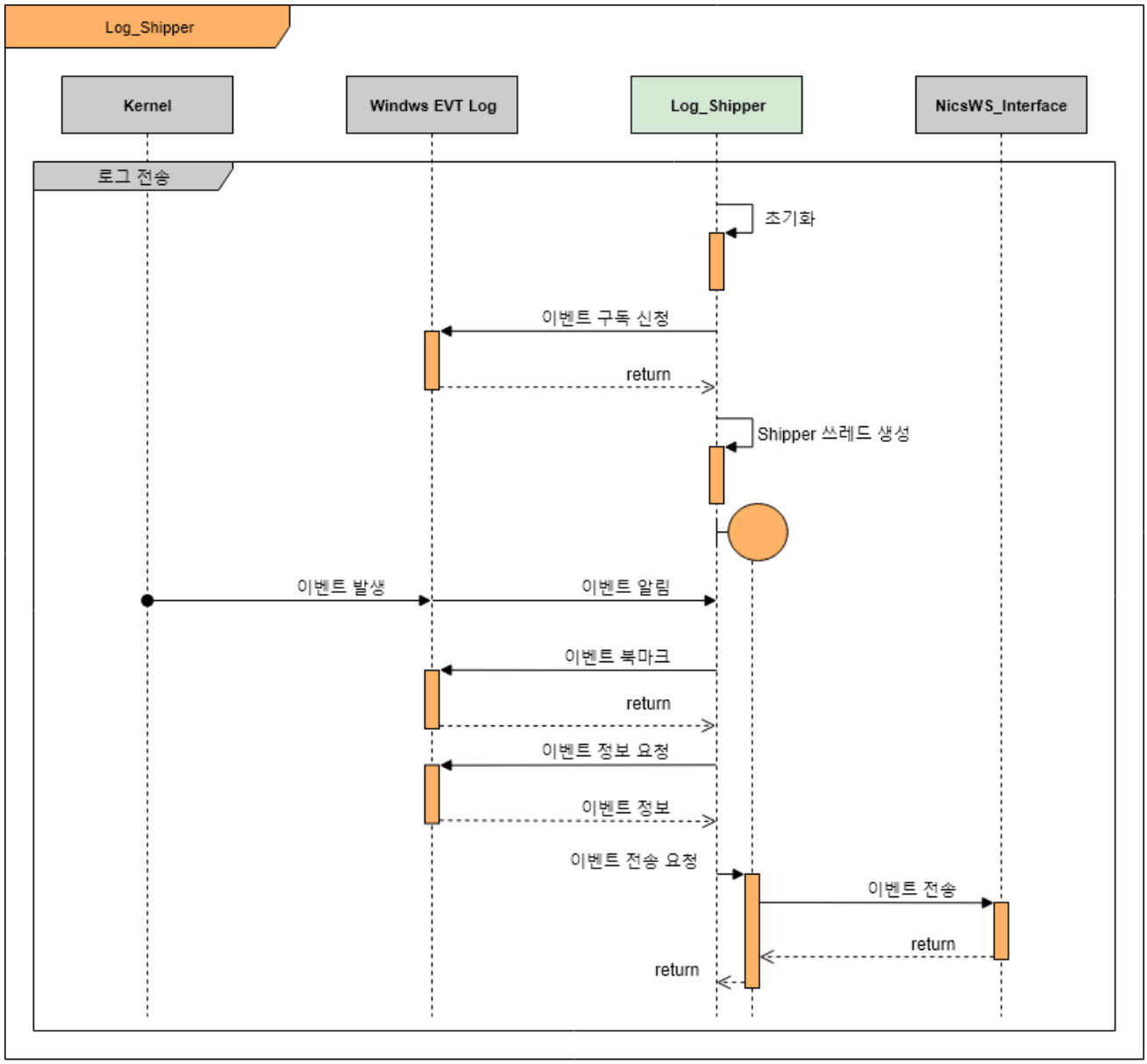
- **Log_Shipper**

1) Basic Requirements (기본 요구 사항)

- Windows EVT Log 를 실시간 구독 가능해야 한다.
- Windows EVT Log 의 종류에 관계없이 구독 가능해야 한다.
- 설정을 통해 Windows EVT Log 를 필터링 할 수 있어야 한다.
- 로그를 실시간 또는 버퍼링을 통해 주기적으로 전송 가능해야 한다.
- 자체적으로 Bookmark 기능이 가능해야 한다
  (재시작 시 마지막 Bookmark 된 이벤트 이후의 이벤트부터 전송)
- Windows EVT Log 전송 시, 별도의 쓰레드로 동작해야 하며, 다른 동작이 멈추어서는 안된다.
- 전송 실패시에 대한 처리가 되어 있어야 한다.

2) Sequence Diagram (시퀀스 다이어그램)

① Windows EVT Log 전송



[그림 3-5 Log_Shipper 로그 전송]

3) Data Struct & Function

① Windows EVT Log 포맷

표. 3-18 Log_Shipper 이벤트 아이디 (종류)

이벤트 아이디	의미	비고
1	Process Creation	프로세스 생성
2	A Process change a file create time	파일 접근 시간 변경
3	Network Connection	네트워크 접근 (IP/PORT)
4	Sysmon service state changed	Sysmon service 상태 변경

5	Process terminated	프로세스 종료
7	Image Loaded	DLL 로드
11	File Create	파일 생성 (접근)
12	Registry Event (Object create and delete)	레지스트리 생성/삭제
13	Registry Event (Value set)	레지스트리 값 설정

표. 3-19 이벤트 아이디별 로그 포맷

로그 기본 포맷 예시


```
{
  "@timestamp": "2018-03-05T07:24:10.194Z",
  "@metadata": {
    "beat": "winlogbeat",
    "type": "doc",
    "version": "6.2.2"
  },
  "message": "Process Create:\nUtcTime: 2018-03-05 07:24:10.147\nProcessGuid: {A376A8F6-F09A-5A9C-0000-
00107EE13802}\nProcessId: 5708\nImage: C:\\Program Files\\Mozilla Firefox\\firefox.exe\nFileVersion: 58.0.2\nDescription:
Firefox\nProduct: Firefox\nCompany: Mozilla Corporation\nCommandLine: \"C:\\Program Files\\Mozilla Firefox\\firefox.exe\"
http://google.co.kr/\nCurrentDirectory: C:\\WINDOWS\\system32\\\nUser: DESKTOP-WJANG\\wook8170\nLogonGuid: {A376A8F6-
ECB5-5A9C-0000-00208AB21300}\nLogonId: 0x13B28A\nTerminalSessionId: 1\nIntegrityLevel: Medium\nHashes:
MD5=22819EB3C648583352AE83003E4C9A56,SHA256=4B69AE5B7D548EFC56FDBE2E650CE2B519AE0C9C928640023077B956C51A7138\nParentProce
ssGuid: {A376A8F6-F098-5A9C-0000-00109D6C3702}\nParentProcessId: 24128\nParentImage:
C:\\Windows\\System32\\RuntimeBroker.exe\nParentCommandLine: C:\\Windows\\System32\\RuntimeBroker.exe -Embedding",
  "user": {
    "identifier": "S-1-5-18",
    "name": "SYSTEM",
    "domain": "NT AUTHORITY",
    "type": "Well Known Group"
  },
  "task": "Process Create (rule: ProcessCreate)",
  "process_id": 8608,
  "version": 5,
  "level": "정보",
  "record_number": "17",
  "opcode": "정보",
  "thread_id": 19220,
  "type": "wineventlog",
  "log_name": "Microsoft-Windows-Sysmon/Operational",
  "event_id": 1,
  "provider_guid": "{5770385F-C22A-43E0-BF4C-06F5698FFBD9}",
  "source_name": "Microsoft-Windows-Sysmon",
  "event_data": {
    "Hashes":
"MD5=22819EB3C648583352AE83003E4C9A56,SHA256=4B69AE5B7D548EFC56FDBE2E650CE2B519AE0C9C928640023077B956C51A7138",
    "FileVersion": "58.0.2",
    "CommandLine": "\"C:\\Program Files\\Mozilla Firefox\\firefox.exe\" http://google.co.kr/",
    "ParentImage": "C:\\Windows\\System32\\RuntimeBroker.exe",
    "ProcessId": "5708",
    "Product": "Firefox",
    "ProcessGuid": "{A376A8F6-F09A-5A9C-0000-00107EE13802}",
    "LogonGuid": "{A376A8F6-ECB5-5A9C-0000-00208AB21300}",
    "ParentCommandLine": "C:\\Windows\\System32\\RuntimeBroker.exe -Embedding",
    "LogonId": "0x13b28a",
    "Image": "C:\\Program Files\\Mozilla Firefox\\firefox.exe",
    "Description": "Firefox",
    "ParentProcessId": "24128",
    "CurrentDirectory": "C:\\WINDOWS\\system32\\",
    "TerminalSessionId": "1",
    "UtcTime": "2018-03-05 07:24:10.147",
    "ParentProcessGuid": "{A376A8F6-F098-5A9C-0000-00109D6C3702}",
    "Company": "Mozilla Corporation",
    "IntegrityLevel": "Medium",
    "User": "DESKTOP-WJANG\\wook8170"
  },
  "beat": {
    "version": "6.2.2",
```

```
   }
}
```

Process Creation – Event ID 1


```
{
    "Hashes":
"MD5=22819EB3C648583352AE83003E4C9A56,SHA256=4B69AE5B7D548EFC56FDBE2E650CE2B519AE0C9C928640023077B956C51A7138",
    "FileVersion": "58.0.2",
    "CommandLine": "\"C:\\Program Files\\Mozilla Firefox\\firefox.exe\" http://google.co.kr/",
    "ParentImage": "C:\\Windows\\System32\\RuntimeBroker.exe",
    "ProcessId": "5708",
    "Product": "Firefox",
    "ProcessGuid": "{A376A8F6-F09A-5A9C-0000-00107EE13802}",
    "LogonGuid": "{A376A8F6-ECB5-5A9C-0000-00208AB21300}",
    "ParentCommandLine": "C:\\Windows\\System32\\RuntimeBroker.exe -Embedding",
    "LogonId": "0x13b28a",
    "Image": "C:\\Program Files\\Mozilla Firefox\\firefox.exe",
    "Description": "Firefox",
    "ParentProcessId": "24128",
    "CurrentDirectory": "C:\\WINDOWS\\system32\\",
    "TerminalSessionId": "1",
    "UtcTime": "2018-03-05 07:24:10.147",
    "ParentProcessGuid": "{A376A8F6-F098-5A9C-0000-00109D6C3702}",
    "Company": "Mozilla Corporation",
    "IntegrityLevel": "Medium",
    "User": "DESKTOP-WJANG\\wook8170"
  }
```


A process changed a file creation time – Event ID 2


```
{
    "CreationUtcTime": "2017-04-24 03:51:05.300",
    "PreviousCreationUtcTime": "2018-03-05 07:25:49.952",
    "UtcTime": "2018-03-05 07:25:49.971",
    "ProcessGuid": "{A376A8F6-ECC2-5A9C-0000-0010E1111D00}",
    "ProcessId": "17568",
    "Image": "C:\\Program Files\\Mozilla Firefox\\firefox.exe",
    "TargetFilename":
"C:\\Users\\wook8170\\AppData\\Roaming\\Microsoft\\Windows\\Recent\\CustomDestinations\\JIMMG587JA65SLJ1SYZD.temp"
  }
```


Network connection – Event ID 3


```
{
    "SourceIsIpv6": "false",
    "DestinationIp": "216.58.200.3",
    "UtcTime": "2018-03-05 07:24:10.536",
    "DestinationPort": "443",
    "ProcessId": "17568",
    "SourceHostname": "DESKTOP-WJANG",
    "SourcePort": "2199",
    "ProcessGuid": "{A376A8F6-ECC2-5A9C-0000-0010E1111D00}",
    "DestinationIsIpv6": "false",
    "DestinationPortName": "https",
    "Protocol": "tcp",
    "SourceIp": "192.168.198.67",
    "Image": "C:\\Program Files\\Mozilla Firefox\\firefox.exe",
    "DestinationHostname": "hkg12s11-in-f3.1e100.net",
    "Initiated": "true",
    "User": "DESKTOP-WJANG\\wook8170"
  }
```


Sysmon service state changed – Event ID 4


```
{
    "Version": "7.01",
    "SchemaVersion": "4.00",
    "UtcTime": "2018-03-05 07:23:14.958",
    "State": "Started"
}
```


Process terminated – Event ID 5


```
{
    "UtcTime": "2018-03-05 07:24:10.322",
    "ProcessGuid": "{A376A8F6-F09A-5A9C-0000-00107EE13802}",
    "ProcessId": "5708",
    "Image": "C:\\Program Files\\Mozilla Firefox\\firefox.exe"
}
```

Image loaded – Event ID 7


```
"Signature": "Microsoft Windows",
    "Signed": "true",
    "ProcessId": "17704",
    "UtcTime": "2018-03-05 07:32:36.738",
    "ProcessGuid": "{A376A8F6-F294-5A9C-0000-001052E48503}",
    "FileVersion": "10.0.16299.15 (WinBuild.160101.0800)",
    "Product": "Microsoft® Windows® Operating System",
    "SignatureStatus": "Valid",
    "Image": "C:\\Program Files\\Mozilla Firefox\\firefox.exe",
    "Company": "Microsoft Corporation",
    "Hashes":
"MD5=222A8E8EA615529B5025DE5782830AF1,SHA256=7C76BB7AEC3C51164A49041E8A710C3C291BA1D30BAE07D126434A52A80D99E1",
    "ImageLoaded": "C:\\Windows\\System32\\kernel32.dll",
    "Description": "Windows NT 기반 API 클라이언트 DLL"
}
```


FileCreate – Event ID 11


```
{
    "ProcessId": "24360",
    "Image": "C:\\WINDOWS\\explorer.exe",
    "TargetFilename": "K:\\cc - 복사본 (2).txt",
    "CreationUtcTime": "2018-03-05 07:27:50.280",
    "UtcTime": "2018-03-05 07:27:50.291",
    "ProcessGuid": "{A376A8F6-EF7D-5A9C-0000-0010537B8601}"
}
```


RegistryEvent (Object create and delete) – Event ID 12


```
{
    "ProcessId": "19128",
    "Image": "C:\\WINDOWS\\regedit.exe",
    "TargetObject": "HKLM\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\새 키 #1",
    "EventType": "CreateKey",
    "UtcTime": "2018-03-05 07:43:03.093",
    "ProcessGuid": "{A376A8F6-F4E7-5A9C-0000-001010A5C804}"
 },
```


RegistryEvent (Value Set) – Event ID 12


```
{
    "Image": "C:\\WINDOWS\\regedit.exe",
    "TargetObject": "HKLM\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\새 값 #1",
    "Details": "(Empty)",
    "EventType": "SetValue",
    "UtcTime": "2018-03-05 07:43:08.639",
    "ProcessGuid": "{A376A8F6-F4E7-5A9C-0000-001010A5C804}",
    "ProcessId": "19128"
 },
```


② 내부 함수

표. 3-20 Log_Shipper 내부 함수

Function Name	InitLogShipper
Return type	BOOL
Argrument(방향, 타입, 이름)	(VOID)
Description	Log_Shipper 초기화 작업을 수행한다.
ETC	

Function Name	StartLogShipper
Return type	BOOL
Argrument(방향, 타입, 이름)	(VOID)
Description	Log_Shipper 를 시작한다.

ETC	

Function Name	StopLogShipper
Return type	BOOL
Argrument(방향, 타입, 이름)	(VOID)
Description	Log_Shipper 를 중지한다.
ETC	

- **Controller**

1) Basic Requirements (기본 요구 사항)

- Policy_Manager 로부터 수신된 정책을 Kernel Driver 에 전달한다.
- 각 드라이버가 담당하는 대응 정책은 다음과 같다

① 프로세스 대응 정책 : NICFSFD (파일 시스템 필터 드라이버)
② 레지스트리 대응 정책 : NICFSFD (파일 시스템 필터 드라이버)
③ 파일 대응 정책 : NICFSFD (파일 시스템 필터 드라이버)
④ 네트워크 차단 : SafeTDI, SafeLWF (네트웍 필터 드라이버, 방화벽 드라이버)

2) Sequence Diagram (시퀀스 다이어그램)

- [그림 3-4 Policy_Manager 대응 정책 요청] 참고

3) Data Struct & Function

- [표. 3-8 내부 자료 구조] 참고