정보보호 시스템간 보안메시지 교환 아키텍쳐

이병희°[1] 김성원[1] 권영찬[1] 윤호상[2] 장희진[2] 이성기[2]
[1](주)안철수연구소
{bhlee, magician, yckwon}@ahnlab.com
[2]국방과학연구소
{yunhs, janghj, leesk}@add.re.kr

SMEA (Secure Message Exchange Architecture) for Message Exchange between Information Security Systems

Byoung-Hee Lee°[1] Sung-won Kim[1] Young-chan Kwon[1]
Ho-sang Yun[2] Hee-jin Jang[2] Seongkee Lee [2]
[1]AhnLab, Inc.
[2]Agency for Defense Development

요 약

본 논문은 정보보호 시스템들간에 교환해야 하는 침입탐지 정보 및 대응 수행 명령들을 효율적으로 전달 및 처리 응답을 수신 받기 위한 아키텍쳐에 대한 정의를 수행한다. 각 시스템에서 제공하는 서비스를 제공받고자 하는 시스템 측에서는 원격호출 메시지를 전송함으로써 서비스를 제공받으며, 시스템 간의 동적 연결을 위한 '시스템 분포맵' 의 정의, 메시지 교환 시에 인가되지 않은 시스템 또는 인가되지 않은 메시지의 교환을 차단하기 위한 '시스템 서비스맵' 의 정의, 각기 다른 분리된 네트워크 상에 존재하는 시스템간 통신을 위한 Proxy 기능을 수행하는 시스템을 통한 메시지의 Tunneling 기능, 교환되는 XML 메시지의 교환 성능 향상을 위해 'Fast InfoSet' 을 적용한 보안메시지 교환 아키텍쳐(SMEA)를 정의한다.

1. 서론

통신망에서 사용되고 있는 보안 프로토콜의 종류는 사용자 인증을 위한 프로토콜과 메시지보안을 위한 프로토콜의 분류만으로 봤을 때 그 종류와 수는 다수가 존재하며, 각 프로토콜 및 방법론들은 각각의 장점이 존재하지만, 취약성 및 성능 부분 등에서 단점 역시 보유하고 있다.

일례로 기존의 IDXP(RFC 4767)는 수신되는 채널의 생성요청에 대해(정해진 프로파일 URI가 동일하며, 표준 인증서 기반의 handshake 만 성공한다면) 불순한 의도에 의한 무조건적인 채널 할당을 수락하는 형태로 구성되어 있어서 허가되지 않은 전송 패킷에 의한 부하를 감수해야 하며, 또한 기존의 IDMEF 등과 같은 메시지 형태의 경우 단순 메시지 형태를 수신 측에서 IDMEF 형태의 XML에 대한 파싱 작업 등을 처리해야 하는 부하를 안고 업무를 수행하게 되며, 원격의 분산되어 있는 서비스, 객체 또는 서버에 접근할 수 있다는 장점을 가진 SOAP 프로토콜은 응답시간이 느리다는 단점을 가지고 있다.

본 논문에서는 이러한 한계를 극복하고, '원하지 않는 접근' 및 '부적절한 컨텐츠'에 대한 거부를 수행하여 '부적절한 트랜잭션 처리량' 을 감소시킴과 동시에 원격 시스템의 서비스를 직접적으로 접근하여 서비스를 제공받을 수 있는 메시지 교환 아키텍쳐를 제시하고자 한다.

본 논문의 목적은 정보보호 시스템간의 메시지 교환 전 채널생성 시점에 인가(Authorization)된 메시지여부의 검증, 교환되는 XML 메시지를 Binary화하여 적은 양의 데이터를 전송할 수 있도록 Fast Infoset을 적용한 '시스템 구조' 기법 및 SMEA 내에서 사용하는 BEEP(RFC 3080) 기반의 원격 시스템 서비스(Function) 단위로 호출을 수행 함으로써 각 시스템들이 해당 업무들을 효율적으로 수행할 수 있는 구조를 갖는 'SMEP 프로토콜'을 적용한 '보안메시지교환 아키텍쳐 (SMEA)' 의 구조를 설명할 것이다.

* 이 논문은 국방과학연구소 사이버침입탐지 및 대응기술 사업의 지원으로 작성되었음

2. 연구동향

2.1. IDMEF

IDS에서 침입탐지의 결과를 관리자에게 통보하기 위해 IDS 로그 표준 형식안인 IDMEF(Intrusion Detection Message Exchange Format)가 표준안으로 제안되어 사용되고 있다. IDMEF에서는 UML 클래스 다이어그램을 사용하여 로그의 데이터 모델을 정의하였고, 데이터 모델을 실제 표현하는 방법으로 XML을 이용하였다.

기존의 IDMEF는 침입탐지, 대응시스템, 관리 시스템간의 통신을 위한 이론적 근거와 정보 교환에 필요한 고수준의 요구사항을 기술하고 있지만, 메시지만으로는 특정 절차 및 업무를 수행하기 위한 Mechanism은 가질 수 없다는 즉, IDMEF 형태의 메시지는 단순한 교환되는 메시지 형태를 가질 밖에 없다는 단점을 갖고 있다.

2.2. IDXP

IDMEF 기반의 XML 경보 데이터를 관리자로 통보하기 위한 프로토콜로서 IDWG에서는 BEEP(Block Extensible Exchange Protocol)기반의 IDXP(Intrusion Detection Exchange Protocol)를 사용하고 있다. BEEP 프로토콜은 IDXP가 TCP/IP 상에서 사용될 수 있도록 해주는 기본 프로토콜이며, TCP/IP 계층에서 동작하는 모든 프로토콜을 블록화하여 프로파일 형태로 제공한다.

IDXP는 BEEP상에서 상호 인증, 기밀성 등을 보장하는 프로토콜로서, BEEP 세션 형성 이후 프로파일 협상 그리고 IDXP 프로파일을 통해서 통신을 수행한다.

기존의 IDXP는 메시지 교환을 위한 채널에 대한 인가(Authorization) 기능이 결여되어 있다. 채널 할당 요청에 대해 인가 기능을 제공하지 않을 경우, 인증서 기반으로 인증된 시스템이라 할지라도 인가되지 않은 메시지 채널 생성 요청을 수렴하게 될 것이며, 그로 인해 발생될 수 있는 트래픽 증가량을 감수할 수 밖에 없는 상황이 될 것이다.

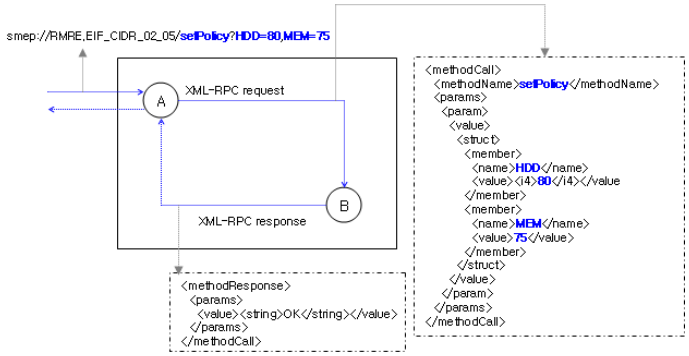
3. 보안메시지 교환 프로토콜 (SMEP, Secure Message Exchange Protocol)

본 논문에서는 IDMEF에서 제시하는 단순 메시지의 형태를 탈피하기 위해 BEEP(RFC 3080) 기반의 원격 시스템이 제공하는 기능을 서비스(Function) 단위로 호출을 수행 함으로써 각 시스템들이 해당 업무들을 효율적으로 수행할 수 있는 구조를 갖는 보안메시지교환 프로토콜 (SMEP, Secure Message Exchange Protocol)을 사용한다.

3.1. 교환 메시지

SMEP 프로토콜을 이용한 메시지 교환 구조는 원격지의 서비스를 제공하는 시스템의 서비스를 호출하는 방식을 갖기 위해 XML-RPC 형식을 따른다. SMEP 프로토콜은 기본적으로 "smep://수신시스템식별자.메시지식별자/함수명?인자=값,인자=값" 구조의 URI schema를 갖는다.

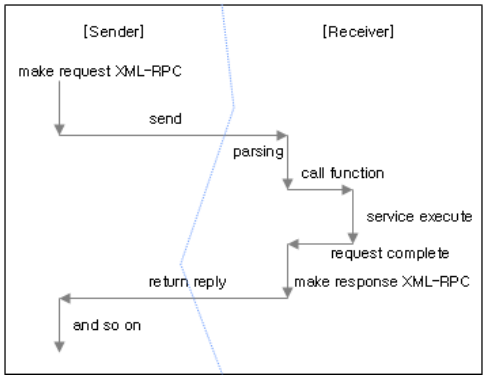
입력된 URI schema 정보에서 '수신시스템 식별자'와 '메시지 식별자'는 채널생성요청 및 채널유지를 위한 식별자로 사용되며, 함수명과 인자(parameter) 및 값은 [그림 7]과 같은 XML-RPC 구조의 데이터의 생성을 위해 사용된다.



[그림 1] smep URI schema 기반의 메시지 교환

3.1.1. 원격함수호출 (Remote Function Call)

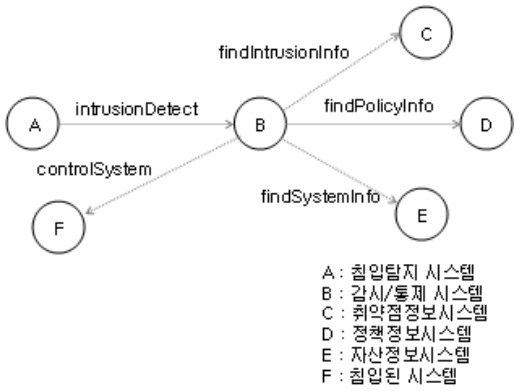
서버 측으로 호출된 XML-RPC 메시지는 SMEP 프로토콜 내부적으로 변환되어 Function Call 기능을 수행한다. 전달된 메시지의 파라미터들을 이용하여 함수 호출을 수행하며, 처리 결과 값을 제공받는 구조를 갖는다.



[그림 2] 원격함수호출의 절차

3.1.2. 메시지 Modeling 및 Mechanism 의 예

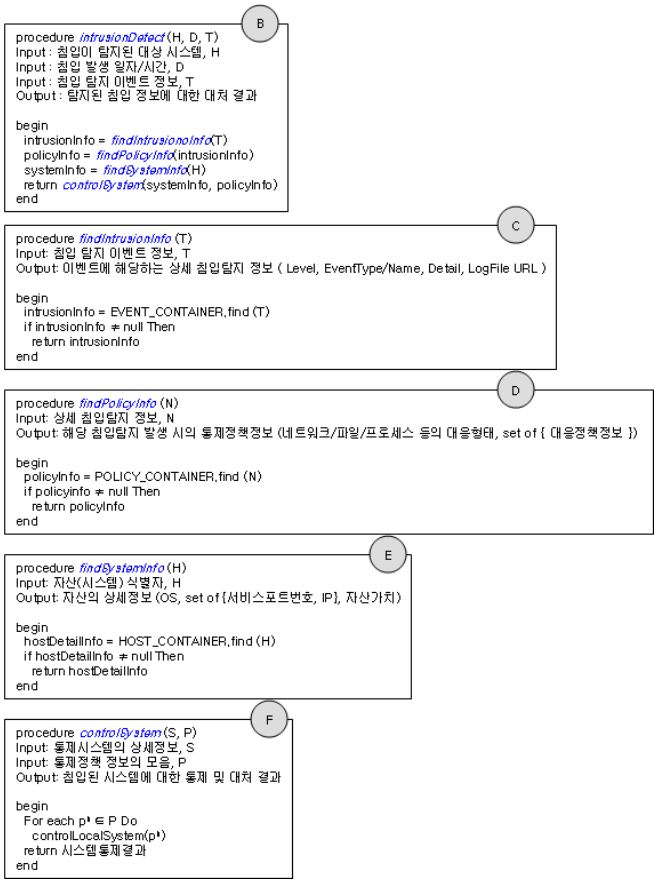
SMEP에서 사용될 수 있는 XML-RPC 형태의 메시지들은 함수 호출 구조를 가지며, 단순 메시지의 정의 및 교환이 아닌 아래의 예와 같은 함수 호출의 및 응답을 제공받는 즉, Mechanism 을 갖는 메시지 흐름의 형태로 표현될 수 있다.



[그림 3] 침입탐지 및 통제 시스템의 예

예로 들은 [그림 9]의 '침입탐지 시스템(A)'은 'F 시스템'이 침입된 것을 감지하여 '감시/통제 시스템(B)'에게 침입정보를 제공하면, 침입정보를 이용하여 상세한 침입정보의 획득, 해당 침입에 대한 통제정책, 침입 당한 시스템의 상세정보를 각 '취약점정보시스템(C)', '정책정보시스템(D)', '자산정보시스템(E)' 으로부터 획득하여 '침입된 시스템(F)'에 대해 통제정책을 수행하도록 명령을 내리는 절차의 예를 표현하고 있다.

예로 들은, [그림 9]와 같은 시스템이 존재할 때, SMEP 프로토콜을 이용한 메시지 Modeling 및 Mechanism의 예는 아래와 같이 표현될 수 있다.

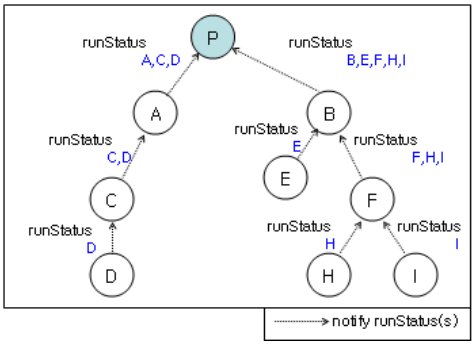


[그림 4] 메시지 Modeling & Mechanism의 예

[그림 10]과 같이 각 형태의 메시지 Modeling 정보는 XML-RPC 형태의 함수 호출 구조를 가지며, 업무를 처리하기 위해 필요한 정보(함수인자)를 함께 제공하여 업무를 처리할 수 있는 Mechanism을 가질 수 있다. 처리결과(Output)는 IDMEF 표준의 메시지 형식을 따르도록 한다.

3.2. 동작상태(runStatus) 정보의 전송

상위 시스템과 통신을 수행하는 하위 시스템들은 상위 시스템으로 주기적인 동작상태(runStatus)정보를 전송한다. 이 정보는 '시스템 분포맵' 상에 존재하는 모든 시스템들의 동작상태를 모니터링 할 수 있는 정보를 제공한다. 각 시스템은 하위 시스템이 존재할 경우 하위 시스템의 runStatus 정보와 함께 본 시스템의 runStatus를 전송한다.



[그림 5] 동작상태(runStatus) 정보의 전송

[그림 4] 에서 '시스템 B'의 경우 자신의 동작상태 정보뿐 아니라 관리하고 있는 하위 시스템들(E, F, H, I)의 동작상태 정보도 함께 전송한다.

아래는 '동작상태(runStatus)'의 메시지 구성요소를 표현한다.

동작상태는 '초기화 중', '구동 중', '연결 끊김' 상태로 표현된다.

```
runStatus ::= SEQUENCE
{
   netId    STRING
   sysId    STRING
   ip       STRING
   status   CHOICE OF {initializing, running, disconnected}
}
```

4. 시스템 구조 (System Architecture)

본 논문에서 제시하는 시스템 구조는 IDXP에 결여되어 있는 인가(Authorization)된 메시지만이 교환될 수 있도록 함으로써 '원하지 않는 접근' 및 '부적절한 컨텐츠'의 차단을 수행하여 '부적절한 트랜잭션의 처리량'을 감소시키기며, 각 시스템간의 식별자만을 이용한 연결, Text 기반의 XML 데이터를 바이너리화 하여 XML 데이터의 교환 성능의 향상, 각기 다른 네트워크 상에 존재하는 시스템간의 원활한 메시지 교환을 위한 SMEA의 시스템 구현적인 측면에서의 구조를 설명한다.

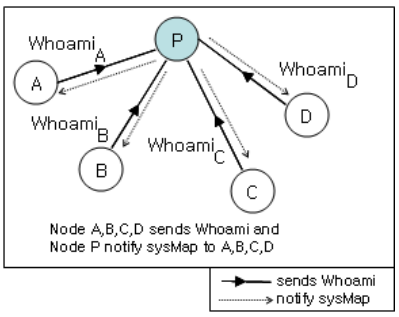
4.1. Access Control

시스템간의 메시지 교환을 위해 '접근제어' 업무를 수행하기 위한 SMEA 내의 기능을 설명한다.

식별자만을 이용한 시스템간의 연결 및 '원하지 않는 접근', '부적절한 컨텐츠'의 차단 업무를 수행하기 위해 필요한 Component (shakeholder[1])인 관리 및 Router (본 논문의 각 'P'들) 로서의 기능을 갖는 시스템이 존재한다.

4.1.1. 시스템 분포맵 (sysMap)

일반적으로 시스템간 연동을 위한 연결은 각 시스템의 IP 및 서비스 제공 Port번호를 이용하여 수행하지만, SMEA 기반의 프레임웍에서는 상위 시스템(P)에서 '시스템 분포맵' 정보를 유지함으로써, 각 시스템간의 연결을 IP 및 Port에 대한 인지 없이도 시스템 식별자만으로도 접속이 가능할 수 있도록 한다.



[그림 6] 시스템 분포맵 유지 및 배포

[1] 일반적으로 '투자자', '이해관계자'를 의미하지만, 본 논문에서는 필수요소(참가 시스템)로서의 의미를 부여한다.

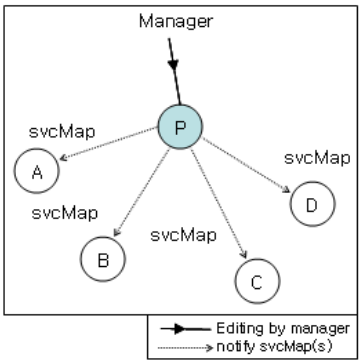
실제 교환되는 Whoami 메시지의 구성요소는 아래와 같다.

```
Whoami ::= SEQUENCE
{
   netId     STRING,
   sysid     STRING,
   ip        STRING,
   svcPort   INTEGER
}
```

시스템(P)에서 제공하는 sysMap 정보는 Whoami 메시지와 동일한 구조를 가지며, 다수의 Whoami 데이터를 가질 수 있다. 제공받은 sysMap 정보를 이용하여 각 A, B, C, D 시스템은 시스템의 IP, Port가 아닌 시스템식별자(ID)를 이용하여 접속 수행이 가능하다.

4.1.2. 시스템 서비스맵 (svcMap)

SMEA는 메시지 교환을 위한 채널 생성 시, '시스템 서비스맵' 데이터를 이용하여 인가된 메시지에 대해서만 채널을 생성할 수 있는 매커니즘을 제공함으로써, '원하지 않는 접근'에 대한 거부를 수행할 수 있다. '시스템 서비스맵'은 상위 시스템의 관리자에 의해 설정될 수 있으며, 각 하위 시스템들은 상위 시스템으로부터 '시스템 서비스맵'을 조회하여 유지할 수 있다.



[그림 7] 시스템 서비스맵 관리 및 배포

아래는 '시스템 서비스맵'의 메시지 구성요소를 표현한다.

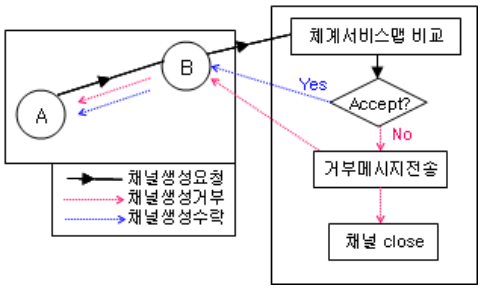
```
svcMap ::= SEQUENCE
{
   srcNetId      STRING,
   srcSysid      STRING,
   targetNetId   STRING,
   targetSysId   STRING,
   interfaceId   STRING,
   accept        BIT STRING
}
```

메시지의 근원지인 Source 시스템으로부터 발생되는 Target 시스템으로의 해당 인터페이스ID의 메시지에 대한 수락여부(Accept)를 결정지을 수 있는 정보를 갖는다.

각 A, B, C, D 시스템은 제공받은 '시스템 서비스맵(svcMap)'을 이용하여 '채널생성 인가 (Authorization)' 업무를 수행한다.

4.1.3. 채널생성 인가 (Authorization)

각 시스템은 유지하고 있는 '시스템 서비스맵' 정보를 이용하여 타 시스템으로부터 수신되는 메시지 교환을 위한 채널 생성 요청에 대해 인가(Authorization) 업무를 수행한다. 인가가 거부될 경우, 채널생성은 거부되며, 메시지의 교환도 자연히 수행되지 않는다.



[그림 8] 채널생성 인가 (Authorization) 절차

아래는 메시지를 전송하고자 하는 측(그림의 A)측에서 수신 측(그림의 B) 측으로 채널생성을 요청하는 메시지를 나타낸다.

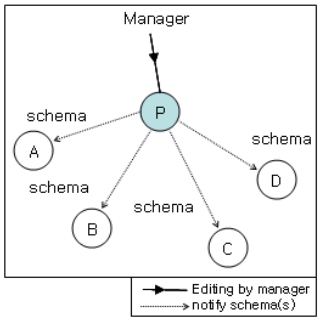
```
MSG 0 1 . 49 106
Content-Type: application/beep+xml
<start number='1'>
    <profile uri='http://iana.org/beep/SMEP'>
        <![CDATA[<SMEP-Greeting                    msgId='
NET1.A.NET1.B.EIF_CIDR_01_01' authCheck='true' />]]>
    </profile>
</start>
END
```

기본 BEEP 표준의 채널생성시 piggyback data [2] 를 이용하여 채널생성을 위한 인가 정보를 제공한다. 위 예는 NET1의 A라는 시스템이 NET1의 B라는 시스템으로의 채널생성을 요청하며 생성이 완료되면 교환될 메시지의 식별자 및 인가(Authorization)요청을 수행한다. NET1 망 내의 B라는 시스템은 유지하고 있는 '시스템 서비스맵' 정보를 이용하여 채널생성에 대한 인가 수락 여부를 결정 후 수락/거부를 수행한다.

4.1.4. 메시지 스키마 관리

SMEA 내의 시스템간에는 XML-RPC 형태의 메시지 교환을 수행하며, 메시지 교환 시 정해지지 않은 '부적절한 컨텐츠'에 대한 처리 거부를 수행하기 위한 메시지에 대한 스키마 검증(메시지 유효성 검증, Validation Check)을 수행한다.

스키마는 상위 시스템의 관리자에 의해 편집될 수 있으며, 각 시스템은 변경된 스키마에 대한 수신을 수행할 수 있다.



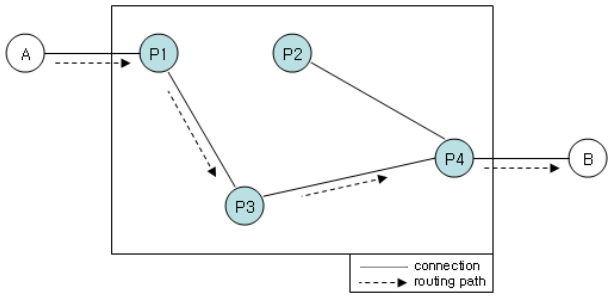
[그림 9] 스키마관리 및 배포

[2] 메시지 전송을 위한 채널을 생성시 전송하는 채널생성요청 메시지에 같이 실어서 보내지는 메시지로 SMEP에서는 이 데이터를 이용하여 해당 채널을 통해 메시지를 송신하는 시스템의 식별 정보 및 메시지 식별자를 획득할 수 있다.

각 시스템은 스키마 정보를 수신 후, 해당 XML 스키마들을 이용하여 메시지에 대한 유효성검증(validation check)을 수행한다.

4.1.5. 채널 Tunneling

메시지를 교환하고자 하는 시스템이 서로 다른 네트워크상에 존재하거나 정책상 단일 포트를 이용한 방화벽을 사이에 둔 시스템간의 메시지 교환이 가능할 경우 application-layer 기반의 메시지 Tunneling 업무를 수행해야 한다. 이러한 환경에서 SMEA 를 적용하는 구현되는 시스템은 채널의 Tunneling 기능을 통해 메시지 교환을 수행하는 proxy/router로서의 업무를 효율적으로 수행할 수 있다.



[그림 10] 메시지 routing 의 예

[그림 12]는 A 시스템이 B 시스템으로 메시지 전송 시 Proxy 기능을 갖는 시스템들을 통해 routing 되는 방식의 예를 나타낸다. 각 SMEA를 적용하는 시스템들은 시스템간 '시스템 분포맵' 정보를 교환하여 아래와 같은 '시스템 분포맵'을 유지한다.

Node Name	IP	Port	Connected Router
A	192,168,1,100	12072	P1
B	192,168,1,101	12073	P4
P1	192,168,1,10	11072	P3
P2	192,168,1,11	11073	P4
P3	192,168,1,12	11074	P1, P4
P4	192,168,1,13	11075	P2, P3

[시스템 분포맵]

최초 전송대상 메시지를 수신받은 P1은 P4에 이르기까지의 채널들에 대한 Tunneling을 수행하기 위해 채널 생성 요청에 대한 piggyback data를 '시스템 분포맵' 기반으로 작성하며 아래와 같은 형태의 데이터를 갖는다.

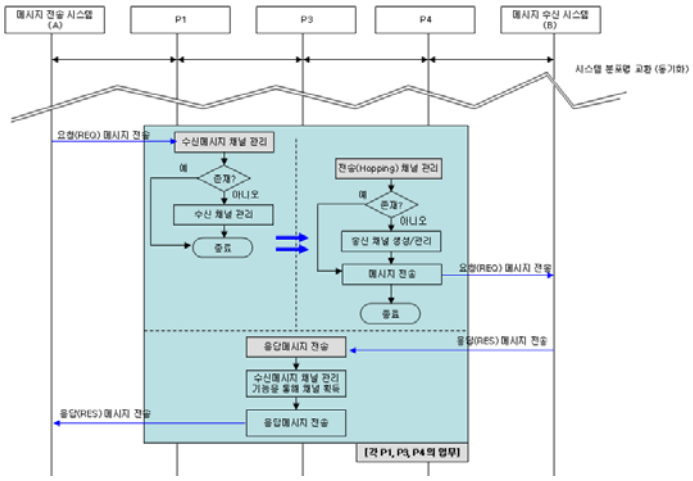
[표 1] P1에서의 P3로의 채널생성 요청의 예

```
MSG 0 1 . 49 106
Content-Type: application/beep+xml
<start number='1'>
    <profile uri='http://iana.org/beep/SMEP'>
        <![CDATA[
          <tunnel fqdn='P3' port='11074'>
            <tunnel fqdn='P4' port='11075'>
              <tunnel fqdn='B' port='12073' />
            </tunnel>
          </tunnel>
        ]]>
    </profile>
</start>
END
```

P3 에서는 [표 1]의 채널생성 요청 수신 후, P3 (현 시스템)에 대한 채널생성 요청 내용만을 삭제한 'P4' 및 'B' 시스템으로의 Tunneling 정보만 남겨 두고 'P4'로의 채널생성 요청을 수행한다.

즉 채널 Tunneling을 위해 각 시스템은 해당 시스템의 FQDN[3] 만을 삭제 후, 인접된 시스템으로 채널 Tunneling을 위한 정보를 제공한다.

아래는 Tunneling 기반의 메시지의 전송 및 응답을 수신받기 위해 SMEA 내부적으로 유지하게 되는 표준 BEEP 채널들에 대한 유지 관점의 흐름도를 표현한다.

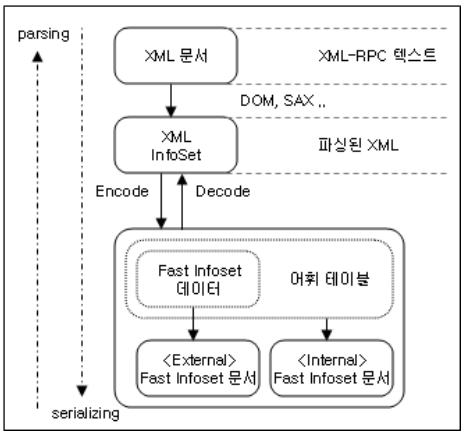


[그림 11] 메시지 routing의 흐름도

4.2. 바이너리 XML 데이터의 교환

서로 다른 환경에서 사용되는 어플리케이션들은 플랫폼과 프로그래밍 언어에 종속적이기 때문에 상호 운용성이 결여되어 있으며 이러한 상호운용성의 확보를 위해 XML 기반의 서비스들이 등장하게 되었다. 하지만 텍스트 데이터로 이루어져 있는 XML 기반의 메시지를 주고받기 때문에 네트워크상의 부하가 예상될 뿐 아니라 전체 응용 프로그램의 성능을 저하시키는 요소로 작용이 될 수 있다. 이에 본 논문에서는 SMEP 프로토콜에서 사용되는 XML-RPC 메시지의 크기를 줄임으로써, 전송과 처리에 소요되는 시간을 줄이고자 "바이너리 XML 인코딩 표준"을 적용하기 위한 Fast Infoset 알고리즘의 도입하여 바이너리 XML-RPC 데이터의 교환을 수행하도록 한다.

[3] Full Qualified Domain Name, 전체주소 도메인명



[그림 12] Fast Infoset en/decoding 흐름도

위 그림은 Fast Infoset 알고리즘을 이용한 바이너리 인코딩 과정을 보여준다. 먼저 XML 문서를 파싱하여 XML 정보셋으로 변환 후, Fast Infoset 알고리즘에 따라 각각의 정보셋을 어휘 테이블(Vocabulary Table)에 저장하고 인덱스로 대체하여 Fast Infoset 데이터로 재구성한다.

재구성된 어휘 테이블과 Fast Infoset 데이터는 Fast Infoset 문서로 Serializing되며, 어휘 테이블의 포함 여부에 따라 External [4] 과 internal[5] 로 구분된다.

4.3. 그 외

SMEA 내에서의 시스템간 인증 및 교환되는 메시지에 대한 암호화, 메시지 유효성 검증은 표준을 따른다.

교환되는 메시지에 대한 암호화 업무는 표준 TLS(Transport Layer Security) 기반으로 수행하며, 시스템간 인증은 인증서 기반의 TLS handshake 과정을 통해 수행하도록 한다.

5. 결론

본 연구는 기존의 메시지 교환 방식의 형식을 벗어나 조직 내의 시스템간 메시지를 교환하기 위한 절차적 모델을 제시하였다.

본 연구는 시스템간의 효과적인 상호운용성을 확보하여 서로 다른 플랫폼 환경을 통합하는 XML 문서 기반의 교환방식에서 발생하는 성능상의 문제점을 개선하는 모델을 제시하였으며, 채널에 대한 인가 기능을 통한 '원하지 않는 접근'으로부터의 원천적인 채널생성의 거부 및 메시지에 대한 유효성 검증을 수행함으로써 '부적절한 컨텐츠'에 대한 거부를 수행할 수 있는 모델을 제시하였다. 또한, 다중 네트워크로 구성된 망상에서의 채널의 Tunneling 기능을 통한 메시지의 전달 모델을 제시하였으며, 원격 함수 호출(RPC) 모델을 적용한 SMEA의 기능 및 구조를 제시하였다.

향 후 연구는 본 논문에서 제시하는 시스템간 메시지 교환 시 '시스템 분포맵'을 이용한 시스템 식별자 정보 기준의 접속 및 메시지 교환이 아닌, Service Repository의 구축 및 연동을 통한 "서비스의 등록" 및 "등록된 서비스의 동적 발견(discovery)" 기술을 적용하는 기능이 될 것이다. 각 시스템에서는 타 시스템들의 노드정보를 알지 못하더라도, 서비스 제공 정보가 발견될 경우 해당 서비스를 제공받을 수 있는 즉, 시스템간 서비스 정보만으로도 유동적인 접속 및 메시지 교환을 수행하는 방식으로 연구가 진행이 되리라 예상한다.

이를 위하여 SOAP 기반의 UDDI(Universal Description, Discovery and Integration)에 연구가 진행되어야 하며, SMEA 구조 내에 적용시킬 수 있는 방법에 대한 연구가 진행이 되어야 할 것이다.

[4] Fast Infoset 문서에 URI를 저장하여 어휘 테이블을 외부에서 참조하는 방식으로 구성

[5] 내부에 어휘 테이블과 인덱스를 모두 포함하는 방식으로 어느 곳에서나 쓰일 수 있지만 External에 비해 파일 크기가 크다는 단점을 가짐

참고문헌

[1]. M. Rose, The Blocks Extensible Exchange Protocol Core, RFC 3080

[2]. M. Rose, Mapping the BEEP Core onto TCP, RFC 3081

[3]. H. Debar, The Intrusion Detection Message Exchange Format (IDMEF), RFC 4765

[4]. T. Dierks, The Transport Layer Security (TLS) Protocol, RFC 4346

[5]. B. Feinstein, The Intrusion Detection Exchange Protocol (IDXP), RFC 4767

[6]. W. Harold, Using Extensible Markup Language-Remote Procedure Calling (XML-RPC) in Blocks Extensible Exchange Protocol (BEEP), RFC 3529

[7]. D. New, The TUNNEL Profile, RFC 3620

[8]. ITU-T Rec. X.891 / Fast InfoSet, http://www.itu.int/rec/T-REC-X.891-200505-I/en