Home Network Security Description Language

김건우*, 한종욱

*한국전자통신연구원

Home Network Security Description Language

Geon-woo Kim*, Jong-wook Han

*Electronics and Telecommunications Research Institute

E-mail : kimgw@etri.re.kr

요 약

홈네트워크 시스템의 안전성을 보장하기 위한 다양한 보안 기술들에 관한 연구가 활발히 진행되고 있다. 현재 개발이 진행되고 있는 보안 기술로는 디바이스 인증, 사용자 인증, 접근 제어, 방화벽과 같은 비교적 단순한 보안 메커니즘을 지원하고 있다. 하지만 이러한 보안 메커니즘들을 효과적으로 관리하고 수행하기 위해서는 홈네트워크를 효율적으로 정의하고 보안을 기술할 수 있는 표현 방식이 필요하다. 따라서 본 논문에서는 홈네트워크 보안 요소를 정의하고 기술하기 위한 xHDL 언어의 문법을 정의하고 각 구성 요소가 가지는 의미를 분석한다.

ABSTRACT

There are a lot of on-going researches on various security technologies for guaranteeing the safety of home network systems. Until now, a few security technologies such as device authentication mechanism, user authentication mechanism, access control mechanism and firewall are generally deployed, and they are just simple. However, we need some representation skills in order to efficiently define the home network and describe the security for managing and performing these security mechanisms. So, in this paper, we define the xHDL language to define and describe the security components of home network and analyze the semantics of each components.

키워드

홈네트워크 보안, 접근 제어, 보안 정책, Home network Security Language, xHDL

Ⅰ. 서 론

홈네트워크는 다양한 네트워크 프로토콜과 서로 다른 홈 디바이스를 사용하기 때문에 기존 공용망에서 발생하는 모든 보안상의 허점에 드러난다. 또한 아직 서비스를 제공하는 단계에 있기 때문에 안전성보다는 서비스 사용의 편리성과 필요성에 중점을 두고 사업과 연구가 전개되고 있는 실정이다.

하지만 홈네트워크 서비스가 점점 확대되고 일반화되면, 예상하지 못했던 다양한 침해 사고가 발생해서 자칫 홈네트워크 사업의 확장에 자칫 걸림돌이 될 수도 있다.

따라서 홈네트워크이 안전성을 보장할 수 있는 보안 메커니즘에 대한 연구와 개발이 절실한 상황이다. 현재, 인증서 기반의 디바이스 인증 메커니즘, 생체 정보와 같은 사용자 편리성과 안전성이 강조된 사용자 인증 메커니즘, 및 실시간 접근 제어 메커니즘들이 있다.

하지만 이들 보안 메커니즘들을 정의하기 위해서는 이들 메커니즘에 사용되는 보안 정책을 저장하고 기술하기 위한 방법이 필요하다.

따라서 본 논문에서는 홈네트워크의 보안 정책

을 기술하고 저장하기 위한 홈네트워크 보안 언어(xHDL: eXtensible Home Security Description Language)를 제안한다.

II. 본 론

홈네트워크의 안전성을 보장하기 위한 방법으로는 기본적인 인증 방식과 접근 제어 메커니즘이 있을 수 있다. 이들 보안 메커니즘은 각 댁내마다 설치되어 있는 홈 게이트웨이/홈 서버를 기반으로 동작하며, 각 댁마다 서로 다른 정책을 설정해서 사용할 수도 있다. 이러한 다양성과 편리성을 지원하기 위해서는 각 홈 게이트웨이에 홈네트워크 보안 정책을 설정할 수 있는 방법을 제공해야 한다.

이러한 홈네트워크 보안 정책을 저장하기 위한 방법으로는 상용 데이터베이스 시스템을 사용하거나 파일 시스템을 사용할 수도 있다. 상용 데이터베이스를 사용하는 방식은 비교적 안전하고 편리하게 사용할 수 있는 장점은 있으나 비용이 많이 소모되고, 일관성 있는 보안 정책을 위해서는 동일한 데이터베이스 시스템을 사용해야 하는 단점이 있다. 파일 시스템을 사용하는 방식은 시스템 자원을 적게 사용하기 때문에 비용이 적게 드는 장점이 있으나 사용하기 불편한 단점이 있다.

하지만, 현재 일반적으로 개발되어서 사용되고 있는 홈 게이트웨이 시스템은 비용에 대한 부담으로 인하여 낮은 시스템 성능과 자원만을 지원하기 때문에, 파일 시스템을 기반으로 보안 정책을 설정하고 저장하는 것이 바람직하다.

본 논문에서 제안하는 xHDL 언어는 XML을 기반으로 정의되어 있으며 다양한 보안 요소와 이들 간의 연관 관계를 기술하는데 효율적이다. 또한, 접근 제어 정책뿐만 아니라 다양한 보안 정책을 기술할 수 있는 장점이 있다.

홈네트워크 보안을 기술할 수 있는 기존 기술로는 각 미들웨어별 보안 언어와 , UPnP Security) XACML과 같은 언어들이 있다. 각 미들웨어별 보안 기술은 각 미들웨어에 종속되어 있으며, 다양한 서비스를 기술하는데 어려움이 있다. 또한 XACML(eXtensible Access Control Markup Language)은 XML document에 대한 접근 제어를 목적으로 하지만 거의 모든 컴포넌트들을 정의할 수 있다. 하지만 XACML의 최대 장점인 일반성은 홈네트워크라는 특정 시스템에 사용되기에는 불필요한 요소들을 많이 포함하고 있으며, 접근 제어를 제외한 다른 보안 요소(보안 정책, 상황 인지 정책 등)를 정의하기에는 어려움이 있다.

따라서 홈네트워크 보안을 기술하는데 적합한 xHDL 언어의 문법을 정의하고 사용을 제안한다.

xHDL에서 정의하는 홈네트워크 보안 요소를 보면 다음과 같다.

- User element
- Object element
- *Object-group element*
- Role element
- Rule element

이들 각 element에 대한 설명을 보면 다음과 같다.

2.1 User Element

User element는 홈네트워크의 사용자를 정의하기 위한 element로서, 크게 mandatory sub-element와 optional sub-element로 구성된다.

각 사용자가 가져야할 mandatory sub-element를 보면 다음과 같다.

- 사용자 ID
- 사용자 이름
- 주민등록번호
- 보안 레벨
- 포함되어 있는 Role

사용자 ID는 ID/Password 사용자 인증 방식에 사용되어 보안 인증 메커니즘과 접근 제어의 기반 정보가 되는 필드이며, 사용자 이름은 실제 이름을 의미한다. 주민등록번호는 부가적인 접근 제어를 위해서 사용되는데, 예를 들어 접근 제어 정책과 관련 없이, 미성년자는 19이상 등급의 VOD 서비스를 사용할 수 없도록 하는 기능에 적용될 수 있다. 보안 레벨의 해당 사용자의 보안 레벨을 지정하기 위해서 필드로서 s0~s3까지의 등급이 있다. s0 등급은 가장 높은 등급을 가지며 s3는 등급이 가장 낮다. 마지막으로 포함되어 있는 Role은 해당 사용자에게 주어진 권한을 의미하며, 포함되어 있는 Role이 가지는 모든 권한이 해당 사용자에게 부여된다. 이러한 Role 정보를 통해서 접근 제어를 기술한다.

User element를 문법을 보면 다음과 같다.

```
users ::= "<users>", {user}, "</users>"
user ::= "<user name=", string, ">", realname, id,
        password, certificate, bio, rfid, nick_name,
        sex,ssn, e-mail, birthday, address, phone,
        mobile, fax, employer, marriage,
assignedRoles, security_level, "</user>"
realname ::= "<realname>", string, "</realname>"
id ::= "<id>", string, "</id>"
password ::= "<password>", string, "</password>"
certificate ::= "<certificate>", string, "</certificate>"
bio ::= "<bio>", string, "</bio>"
rfid ::= "<rfid>", string, "</rfid>"
nick_name ::= "<nick_name>", string,
```

```
        "</nick_name>"
sex ::= "<sex>", ("mail" | "femail"), "</sex>"
ssn ::= "<ssn>", string, "</ssn>"
e-mail ::= "<e-mail>", string, "</e-mail>"
birthday ::= "<birthday>", string, "</birthday>"
address ::= "<address>", string, "</address>"
phone ::= "<phone>", string, "</phone>"
mobile ::= "<mobile>", string, "</mobile>"
fax ::= "<fax>", string, "</fax>"
employer ::= "<employer>", string, "</employer>"
marriage ::= "<marriage>", ("yes" | "no"),
             "</marriage>"
assignedRoles ::= "<assignedRoles>", string,
                  "</assignedRoles>
```

2.2 Object Element

Object element는 홈네트워크 시스템에서 정의될 수 있는 다양한 자원(디바이스, 서비스, 센서)을 정의하기 위한 요소로서, 문법을 보면 다음과 같다.

```
object ::= "<object object_type=", ("device" |
           "service" | "sensor"), ">",
           object_body, "</object>"
object_body::= object_device | object_service |
               object_sensor
object_device ::= code, containedGroup,
                  security_level, location,
                  coordinates, operations
object_service ::= code, containedGroup,
                   security_level
object_sensor ::= code, containedGroup,
                  security_level, location,
                  coordinates
code ::= "<code>", string, "</code>"
containedGroup ::= "<containedGroup>", string,
                   "</containedGroup>"
security_level ::= "<security_level>", ("s1" | "s2" |
                   "s3" | "s4"), "</security_level>"
location ::= "<location>", string, "</location>"
coordinates ::= "<coordinates>", integer, ",", integer,
                "</coordinates>"
operations ::= "<operations>", {operation},
               "</operations>"
operation ::= "<operation>", value, [direction],
              [negation], arguments, "</operation>"
value ::= "<value>", string, "</value>"
arguments ::= "<arguments>", {argument},
              "</arguments>"
argument ::= "<argument>", value, [direction],
             [negation], "</argument>"
direction ::= "<direction>", ("IN", | "OUT" |
              "ALL"), "</direction>"
negation ::= "<negation>", ("yes" | "no"),
             </negation>
```

2.3 Object-group Element

Object-group element는 정의된 object들의 그룹을 지정하기 위한 요소로서, 접근 제어 메커니즘에 사용될 수 있다. 즉, 접근하고자 하는 대상을 각 디바이스나 서비스가 아니라 그룹화하면 보다 효과적으로 정의할 수 있다.

Object-group element의 문법을 보면 다음과 같다.

```
object-groups ::= "<object-groups>", {object-group},
                  "</object-groups>"
object-group ::= "<object-group name=", string, ">",
                 {containingObject},
                 security_level, "</object-group>"
containingObject ::= "<containingObject type=",
                     ("device" | "service" | "sensor"),
                     ">", string, </containingObject>
```

2.4 Role Element

Role element는 역할 기반 접근 제어 정책을 기술하기 위한 요소로서 user element와 object element/object-group element간의 연관 관계를 규정한다. 즉, 해당 role element에 포함된 사용자는 해당 role element가 가지는 object에 관한 모든 권한을 가질 수 있다

Role element의 문법을 보면 다음과 같다.

```
roles ::= "<roles>", {role}, </roles>
role ::= "<role name=", string, ">", export, import,
         assignedUsers, security_level, permissions,
         "</role>"
export ::= "<export>", string, "</export>"
import ::= "<import>", string, "</import>"
assignedUsers ::= "<assignedUsers>", string,
                  "</assignedUsers>"
permissions ::= "<permissions>", {permission},
```

```
        "</permissions>"
permission ::= "<permission>", resource, direction,
        negation, operations, "</permission>"
resource ::= "<resource type=>", string, ">, string,
        </resource>"
```

Role간의 상속 관계를 통해서 다른 Role의 모든 권한을 상속받을 수 있어서 대형 네트워크에서 효율적으로 관리할 수 있는 장점이 있다.

2.5 Rule Element

Rule Element는 홈네트워크에 적합한 다양한 보안 정책을 표현하고 기술하기 위한 요소이다. 즉, 미리 정의되어 있는 조건을 만족하면, 해당하는 과정을 수행하는 보안 정책을 기술하며, 특정 서비스에 국한되지 않고 다양한 정책을 기술할 수 있도록 문법이 정의되었다.

Rule element의 문법을 보면 다음과 같다.

```
rules ::= "<rules>", rule, "</rules>"
rule ::= "<rule name=", string, ">", condition_stmt,
      actions_stmt, "</rule>"
condition_stmt ::= "<condition>", intersection_stmt
        | union_stmt, "</condition>"
intersection_stmt ::= "<intersection>", [event],
        [date], [day], [time], [union_stmt],
        "</intersection>"
union_stmt ::= "<union>", [event], [date], [day],
        [time], [intersection_stmt], "</union>"
date ::= "<date>", string, "</date>"
day ::= "<day>", string, "</day>"
time ::= "<time>", string, "</time>"
event ::= "<event>", [subject], [location], [resource],
      [duration], "</event>"
subject ::= "<subject type=", ("user" | "role"), ">",
        string, "</subject>"
duration ::= "<duration cycle=", ("forever", | "year"
      | "month" | "week" | "day"), " unit=",
      ("month" | "week" | "day" | "hour" |
      "minute"), ">", string, "</duration>"
actions_stmt ::= "<actions>", {action}, "</actions>"
action ::= "<action>", resource, enforcements,
      "</action>"
enforcements ::= "<enforcements>", {enforcement},
      "</enforcemnets>"
enforcement ::= "<enforcement>", value,
      arguments, "</enforcement>"
```

III. 결 론

본 논문은 홈네트워크의 다양한 보안 정책을 기술하고 표현하기 위한 보안 정책 언어인 xHDL을 정의하고 제안한다. xHDL은 XML을 기반으로 작성되어 확장성을 지원하며, 사용자 인증, 디바이스 인증, 접근 제어 정책 및 다양한 상황 인지 정책 등을 효율적으로 기술할 수 있는 장점이 있다.

xHDL을 통해서 사용자는 각 댁내별로 구동되는 홈 게이트웨이에 자신의 댁 특성을 최대한 반영한 보안 정책을 설정해서 적용할 수 있다. 또한 최소한의 시스템 자원만을 사용하기 때문에, 기존 상용 홈 게이트웨이에도 용이하게 적용할 수 있으며, readability가 뛰어나 비 IT 전문가도 쉽게 접근할 수 있는 장점이 있다.

본 논문에서 제안하는 xHDL은 generality 특성을 보다 강화하고 새로운 보안 메커니즘을 위한 정책을 더욱 용이하게 적용할 수 있도록 많은 연구과 개발이 따라야 한다.