액티브네트워크 환경에서 정책기반 침입탐지 구조☆

홍 충 선* 허 용 준**

◈ 목 차 ◈

1. 서 론

액티브네트워크는 망 관리 그리고 QoS제어와 같은 새로운 서비스 구조에 많이 사용되어지고 있는 프로토콜로서 점차고 관심이 증대되고 있는 현실이다. 하지만 기대한 만큼 널리 사용되고 있지는 못한데, 그 주된 이유중의 하나는 보안성에 관한 문제점들 때문이다[1][2]. 그리고 네트워크 노드 보안에 있어서 중요한 이슈중 하나인 침입탐지에 대한 연구도 수행되어지고 있는데, 침입탐지 시스템은 네트워크를 통해 들어오는 악의적인 데이터를 차단하고 새로운 공격 방식의 증가에 대처하기 위한 방화벽의 취약성 때문에 보안 구조에 추가되어 왔다. 최근에 이러한 시스템 요구를 만족시키기 위해 Snort[3], NFR[4], Bro[5] 그리고 Stat[6]과 같은 침입탐지 시스템들이 발표되었다. 그러나 침입탐지 방법의 집중적인 연구에도 불구하고 (a)IDS에 의해서 생성되는 오류 알람율을 줄이는 것, (b)망 관리자가 공격 신호를 기술하기 위한 하이레벨의 표시법을 제공하는 것, 그리고 (c)이미 존재하고 있는 망 관리 구조에 침입탐지 시스템을 접목시키는 방안들이 과제로 남아 있다[7].

침입탐지 시스템은 악의 탐지 방식이나 시그네처 탐지 방식을 사용하여 공격을 탐지하는데, 악의 탐지 방식은 통계값과 일반적인 데이터 트래픽을 비교하는 방법이고, 시그네처 탐지 방식은 공격이 기술된 데이터 소스와 유입하는 패킷을 매칭시킴으로써 행해지는 방식이다. 하지만 IDS에서 생성되는 알람의 90%는 오류이다(false positives)[8]. 악의 탐지 방식을 사용하는 IDS의 단점은 일반적인 경우와 비교하는데 어려움이 있다는 점과, 분산된 서비스 거부 공격과 같은 경우에 있어서 상당한 수준의 네트워크 트래픽을 사용하여 공격이 행해진다면 오류가 거의 나오지 않는다는 것이다. 이런 공격 상황에서 오류 알람의 수는 허용한계(threshold)에 따라서 많거나 적거나 할 수 있다.

또한 시그네처탐지 방식을 사용하는 IDS의 경우는 높은 오류 알람율이 문제점인데, 이런 문제점들은 공격 신호를 표현할 수 있는 언어의 한계와 불완전한 신호의 표현으로 인해 야기되는 것이다. 일반적으로 시그네처는 프로토콜의 상호관계가 아닌 패킷 영역을 조사하는 것과 관계있다. 이러한 제약 효과의 예로 TCP SYN/TCP RST 포트 스캐닝 기술을 살펴볼 수 있는데, 여기에서 사용되는 시그네처는 TCP 패킷이 RST 플래그를 갖고 있는지 아닌지의 조사를 한다. 그러나 이러한 접근 방식의 문제점은 TCP RST 패킷은 같은 종류의 서비스를 사용하지 않는 호스트에서는 생성되지 않는다는 점과, 패킷이나 신호를 병합할 수 없는 IDS는 TCP RST 패킷과 알람을 구별하는데 충분하지 못하다는 것이다. 그리고 오류 알람을 생성하는 문제에 있어서는 우리가 위험을 내포하고 있는 징후

☆ This work was supported by University ITRC Project of MIC.
* 경희대학교 전자정보학부
** 그린정보통신(주) 연구원

와 관계된 정보를 항상 얻을 수 있는 것은 아니라는 점이다. 침입자에 의해서 사용되어지는 기술과 그들에 의해서 발생되는 위협은 시스템 성능을 저하시킬 수 있다. 몇몇 경우에는 침입과 비슷한 경우로 오인하여 오류 알람율을 높이는 동기가 되기도 한다[9][10].

통합적인 보안 메커니즘과 현존하는 망 관리 시스템을 볼 때 아직도 보안과 망 관리 사이에는 많은 격차가 있다. 즉 침입탐지에 사용될 수 있는 MIB가 부재한다는 점이다[7]. 몇몇 망 관리 시스템은 침입탐지 시스템을 형성하는 인터페이스를 제공하고 이를 통해 침입탐지에 사용되는 이벤트를 받을 수 있지만 Qin은 IDS에 의해서 전달된 알람 정보를 효율적으로 분석하고 관리하는 시스템이 아직은 부족하다고 말한다[11][12].

본 기고에서는 침입탐지와 관련하여 시스템 침입탐지 시 성능향상을 위한 방안과 액티브 네트워크 환경에서 액티브 패킷에 대한 보안성 향상 방안을 제시하고, Policy-based security를 접목함으로써 보안성이 향상된 안전한 액티브 네트워크 기반 보안관리시스템을 설계하고자 한다.

본 기고의 구성은 다음과 같다. 2장에서는 본 고와 관련된 연구로서 침입탐지 시스템과 각각의 특징을 살펴보고 3장에서 본 고에서 제안하는 구조를 소개한다. 4장에서는 정책기반 보안 관리에 사용되는 정책과 각 object에 대해 소개하고, 5장에서는 정책 템플릿의 예와 보안 정책을 통해 생성된 액티브 패킷을 소개한다. 그리고 마지막으로 6장에서 결론으로 끝을 맺는다.

2. 관련연구

2.1 침입탐지 SNMP 에이전트

본 절에서는 침입상태 탐지를 위한 SNMP 에이전트를 소개하고, 공격신호를 High-level로 표현하며, 오류 알람수를 줄이는 방법을 제시한 시스템을 소개하고자 한다[7].

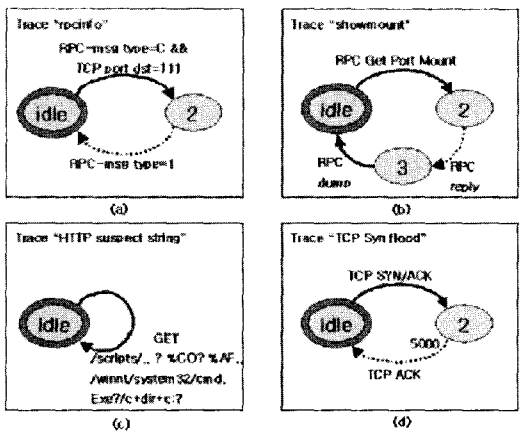
본 관련 연구에서 망 관리자는 모니터되는 공격 신호를 PTSL (Protocol Trace Specification Language)[13]이라 불리는 상태기계어(state machine language)의 사용을 통해 표현하고, 에이전트에서 사용되어지는 신호는 IETF의 Script MIB를 통해서 망 관리자가 표현한다. 일단 프로그램되면 에이전트는 네트워크 트래픽을 통해 신호가 발생되는지를 모니터하고 모니터된 통계값을 확장된 RMON2 MIB로 저장한다. 이러한 통계값들은 SNMP기반의 관리 어플리케이션으로부터 얻어질 수도 있고, 공격 신호를 분석하는데 사용되어질 수 있다.

2.1.1 PTSL을 사용한 공격 신호의 표현

PTSL은 Finite State Machine(FSM)의 개념을 기반으로 하는 프로토콜 Trace를 표현하기 위해 개발된 언어이다. PTSL은 그림(Graphical PTSL)이나 문자(Textual PTSL)로 구현되는 언어이며 서로 같은 내용을 포함하지는 않는다. 즉 문자로 구성된 PTSL은 FSM의 기술과 상태천이가 되는 사건을 포함하여 모든 Trace를 표현 가능하지만, 그림으로 나타내는 방법은 문자로 나타내는 방법의 일부이다.

그림 2-1은 PTSL을 그림으로 나타낸 방법을 보여주고 있다.

(a)에서는 rpcinfo 명령을 감지하는 신호를 볼 수 있다. 이 명령은 RPCs(Remote Procedure Calls)를 서버가 수신하는 과정의 목록을 돌려줌으로써 침입자에게 유용한 정보가 되는 것을 보여준다. 비슷한 방법으로 (b)는 showmount 명령을 감지하는 것을 보여준다. 그리고 그림 (c)에 묘사된 신호는 공격자가 /scripts/..\CO\%AF../winnt/system32/cmd.exe?/ c+dir+c:\의 문자



(그림 2-1) PTSL에서 그림으로 나타낸 공격신호

열을 이용하여 HTTP 응답을 요구한다. 여기에서 URL은 공격자가 HTTP 서버에서script나 CGI를 실행시켜 서버에 위치한 파일들의 리스트를 얻어 내려는 것을 알려주고 있다. 그리고 (d)에서는 SYN flood attack을 감지하기 위한 신호를 표현하고 있다.

그림 2-2는 공격 신호를 문자로 표현한 예를 보여주고 있다.

```
1  Trace "TCP SYN - TCP RST"
2  Version: 1.0
3  Description: Trace to detect port scanning.
4  Key: TCP, SYN, RST, port scanning
5  Port:
6  Owner: Luciano Paschoal Gaspary
7  Last Update: Tue, 16 Aug 2000 15:30:58 GMT
8
9  MessagesSection
10
11 Message "TCP SYN"
12 MessageType: client
13 BitCounter Ethernet/IP 110 1 1 ="Field SYN - 1 means TCP Connect"
14 EndMessage
15
16 Message "TCP RST"
17 MessageType: server
18 BitCounter Ethernet/IP 109 1 1 ="Field RST"
19 EndMessage
20
21 EndMessagesSection
22
23 StatesSection
24 FinalState idle
25
26 State idle
27 "TCP SYN" GotoState 2
28 EndState
29
30 State 2
31 "TCP RST" GotoState idle
32 EndState
33
34 EndStatesSection
35
36 EndTrace
```

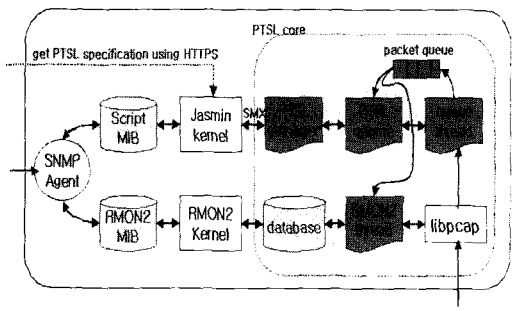
(그림 2-2) PTSL에서 문자로 나타낸 공격신호

문자로 표현된 PTSL은 Trace keyword로 시작하여 EndTrace Keyword로 끝마치게 되고, Catalog와 version control 정보는 Trace keyword 다음에 오게 된다. 그리고 이어서 MessagesSection, GroupsSection, States-Section의 세 개의 영역이 오게 된다.

2.1.2 침입탐지 SNMP 에이전트의 구조

침입탐지 에이전트는 입력값으로 PTSL로 기술된 공격 신호를 요구한다. 그리고 주어진 시간에 모니터되는 신호는 Script MIB를 통하여 망 관리자에 의해서 만들어진다. 일단 프로그램되면 에이전트는 네트워크 트래픽에서 신호의 발생을 모니터하고 통계치들을 신호의 발생에 따라서 확장된 RMON2 MIB로 저장하게 된다. 이 통계치들은 주기적으로 에이전트를 폴링하는 것에 의해서 SNMP 기반의 관리 어플리케이션을 통해 얻어질 수 있다. 그림 2-3은 침입탐지 SNMP 에이전트의 구조를 나타내고 있으며, Linux 환경에서 구동되고 C 언어, POSIX 스레드 library, NET-SNMP 프레임과 Jasmin으로 구현된다. 여기에서 PTSL관리자 스레드는 Script MIB와 PTSL core를 접목시키는데 사용되어지고, 모니터하는데 필요한 새로운 신호를 생성하거나 에이전트에서 기존의 신호를 제거하는 명령이 있을 때마다 PTSL core와 RMON2 protocolDir 테이블에 의해서 사용되어지는 데이터 구조를 업데이트 한다.

세 개 이상의 스레드(큐, PTSL engine, RMON2)가 생성자와 소멸자 사이에서 작업을 수행한다. 첫 번째 스레드는 libpcap library를 사용하여 네트워크 인터페이스 카드에 도착하는 모든 패킷을 모니터하여 원형 큐에 집어넣는다. 두 번째 스레드는 큐에 저장된 모든 패킷을 처리하고 state machine에 작용될 수 있는 신호들이 있는지를 감별하게 된다. 만일 패킷에 특별한 특징이 추가되었다면, 마지막으로 RMON2 스레드가 특징에 따라서 큐로부터 모든 패킷을 제거하고 mySQL 데이터베이스에 저장된 RMON2 테이블을 업데이트하는 구조로 되어있다.



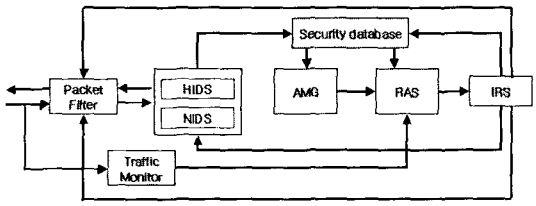
(그림 2-3) 침입탐지 SNMP 에이전트의 내부구조

2.2 프로토콜 악의 탐지와 데이터마이닝을 이용한 NetShield 시스템

본 절에서는 망 공격으로부터 네트워크 서버, 라우터, 클라이언트 호스트등을 보호하기 위한 Netshield system[14][15]에 대해서 소개하고자 한다.

2.2.1 Netshield 보안 시스템의 구조

Nesshield 보안 시스템은 모든 종류의 악의적인 네트워크 웜이나 공격을 방어하기 위해 개발된 시스템으로 데이터마이닝과 프로토콜 악의 탐지[9]기법을 사용한다.



(그림 2-4) NetShield 시스템 구성도

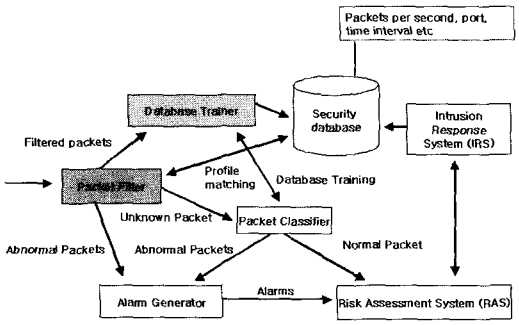
그림 2-4는 이러한 Netshield 시스템의 구성도를 나타내고 있다. 시스템은 크게 IDS, AMG, RAS, IRS로 네 개의 구조로 구성되어 있다. 여기에 패킷 필터, 트래픽 모니터, datamining 개체, security 데이터베이스가 추가된 구조이다. 여기에서 데이터베이스는 모든 공격과 공격에 대응하는 정보를 저장하게 된다.

Netshield 시스템에서 IDS는 침입을 탐지하기 위해 사용되어지고, HIDS(host intrusion detection system)는 네트워크 트래픽상에서 handler나 zombie 호스트를 검출하기 위해 사용되어지며, NIDS(netso가 intrusion detection system)는 시스템 내부로 유입하는 flooding 공격을 검출하기 위해 사용된다. 그리고 traffic monitor는 유입되는 traffic을 검사하여 DDoS 공격을 인식하는데 사용된다.

2.2.2 악의 탐지를 위한 데이터마이닝 메커니즘

Netshield 시스템에서 공격이 탐지된 경우, 이에 대한 대응은 데이터베이스로 저장되게 된다. 저장된 정보는 다시 공격정보(attack profiles)로 쓰이게 되는데, 공격정보는 사용된 프로토콜과 port, 초당 패킷 수, 패킷 전송시간 등으로 구성된다.

그리고 Security 데이터베이스는 짧은 시간동안 구현되지 않고 보통 오랜 시간에 걸쳐 정보를 얻게 된다. 그림 2-5는 네트워크 공격으로부터 공격정보를 얻는 과정을 보여주고 있다. 그림에서 데이터베이스 trainer는 향후에 새로운 공격을 검출하는데 쓰이는 공



(그림 2-5) NetShield 시스템에서 데이터마이닝을 통한 공격정보의 수집과정도

격정보를 갱신하고, 분리하기 위해 사용되어진다.

그리고 security 데이터베이스는 시스템으로 유입되는 패킷으로부터 표 1과 같은 정보를 얻는데, 표에 나와있는 security 데이터베이스 정보는 공격을 검출하는데 사용되는 트래픽값이나 시스템 매개변수이다. 그리고 데이터베이스 엔트리는 서로 다른 공격 패턴을 나타낸다. 여기에서 CPS는 특정 호스트로부터 들어오는 패킷/초이며, RNC는 새로운 연결설정 요구수를 나타낸다. 그리고 NOC는 연결되어있는 호스트의 수를 나타낸다. IPinsec은 안전하지 않은 host나 Zombie의 IP 주소 리스트를 나타내고, MAXCON은 시스템이 제어할 수 있는 연결의 수를 나타낸다.

(표 1) Security 데이터베이스 설계에서 사용되는 네트워크 트래픽과 system 매개변수

Profiles (pps)	CPS	RNC	NOC	MAXCON	(IPinSec)
Entry 1	100 pps	10	75	1000	130.110.x.x
Entry 2	190 pps	23	50	1000	132.23.34.x
Entry 3	1000 pps	100	768	1000	123.x.x.x

표에서 가령 매우 높은 비율로 새로운 연결을 시도하는 경우, security 데이터베이스에 저장되어 있는 정보와 비교하여 RNC값이 커지게 된다. 그러므로 이는 많은 연결설정을 통해 공격을 시도하는 경우로 가정할 수 있다.

2.3 ALAN(Application Layer 액티브 네트워크) 서버를 위한 정책기반 관리구조

2.3.1 ANDROID 관리 구조

Application Layer 액티브 네트워크(ALAN)은 액티브 서버에 사용자 특성화된 서비스(user-customised services : proxylets))들을 빠르고 효율적으로 등록하기 위하여 개발되었다[16][17]. 이는 프로그래밍 가능한 어플리케이션 영역을 이용함으로써 가능하다. 여기에서 Proxylet은 서비스질을 높이거나 사용자에게 새로운 서비스를 소개하는 기능들을 제공한다. IST 프로젝트인 액티브 네트워크 DistRibuted Open Infrastructure Development (ANDROID)는 이러한 ALAN 서버 관리를 목적으로 유연한 정책기반 시스템을 개발하는 것이며, 액티브 서버 정보를 효율적으로 관리하는데 있다.

ANDROID 시스템은 이벤트에 따른 정책을 통해 시스템을 관리하는 구조로 되어있다. 이벤트는 시스템의 상태가 변한 경우 발생하게 되며, 특별한 이벤트가 발생하게 되면 정책이 적용되도록 설정되었다.

2.3.2 보안과 정보관리를 위한 XML 정책예

본 절에서는 ANDROID 시스템에서 적용되는 정책의 예와 정책을 구성하는 요소들을 살펴보도록 한다.

먼저 ANDROID 시스템에서 사용하는 XML 스키마는 그림 2-6과 같이 6개의 엘리먼트로 구성되어있다. 이 6개의 엘리먼트를 통해 이벤트 발생시 적절한 정책을 적용하도록 구성되어 있다.

각각의 엘리먼트에 대한 소개는 다음과 같다.

- creator : 정책의 근원지를 나타낸다.
- info : 정책에 어긋나는 정보를 나타낸다.
- sender : 정책이 적용되는 곳을 나타낸다.
- subject : 정책이 적용되기 위한 엔티티를 나타낸다.
- trigger : 정책이 바뀌게 되는 상태를 나타낸다.
- action : 정책이 바뀌게 될 경우 취하게 되는 동작을 나타낸다.

그림 2-7은 이러한 스키마를 보안 정책에 적용시킨 예이다. 그림에서 aLDPrx(보안 관리자가 proxylet을 실

```
<xsd:schema xmlns="http://www.android.org/policy"
targetNamespace="http://www.android.org/policy"
Xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
elementFormDefault="qualified">
<xsd:element name="policy">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="creator"/>
            <xsd:element ref="info"/>
            <xsd:element name="sender" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="subject"/>
            <xsd:element name="trigger" minOccurs="0"/>
            <xsd:element name="actions" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

(그림 2-6) ANDROID 시스템의 XML 스키마 예

```
<?xml version="1.0" encoding="UTF-8"?>
<policy xmlns="http://www.android.org/policy" xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-
instance" xsi:schemaLocation="http://www.android.org/policy file:///C:/docs/policy.xsd">
    <creator>
        <authority>
            <admin-domain>EE</admin-domain>
            <role>Admin</role>
        <authority>
        <identity>/AS/ADMIN</identity>
        <reply_address>127.0.0.1</reply_address>
    </creator>
    <info>
        <policy-id>270920011237</policy-id>
        <modality>obligation</modality>
    </info>
    <subject>
        <domain>
            <role>Security</role>
        </domain>
    </subject>
    <trigger>
        <event-id>eLDPrx</event-id>
    </trigger>
    <actions>
        <condition>
            <operand>pAuthDeployer</operand>
            <operator>Equals</operator>
            <operand>True</operand>
        </condition>
        <action>
            <target>
                <domain>
                    <role>Resource-Manager</role>
                </domain>
            </target>
            <data>
                <method>aLdPrx</method>
            </data>
            <target>
                <domain>
                    <role>Security Manager</role>
                </domain>
            </target>
            <data>
                <method> aLocSPU</method>
            </data>
        </action>
    </actions>
</policy>
```

(그림 2-7) 보안 관리 정책의 예

행하도록 한다.)와 aLocSPU(보안 관리자가 정책을 생성하거나 업데이트 할 수 있도록 한다.)가 실행되기 전에 pAuthDeployer(정책 배포자가 인증되는지 아닌지에 관한 이벤트)상태를 만족시켜야 한다는 것을 보여주고 있다. 상태가 만족된 경우, 첫 번째 동작은 정보관리자가 proxylet을 실행하도록 하고, 두 번째 동작은 proxylet과 관련된 java.policy 파일을 업데이트 하도록 한다.

3. 제안 구조

지금까지 본 기고와 관련된 연구들에 대해서 살펴보았다. 본 장에서는 관련 연구를 토대로 액티브 네트워크 환경에 적합한 보안구조를 설계하고자 한다.

3.1 설계 이슈

본 고의 목표는 액티브 네트워크 환경에서 네트워크 침입탐지를 통한 보안성 향상 방안의 연구에 있다. 제안 사항을 소개하기에 앞서 제안하는 구조의 설계 목적은 다음과 같다.

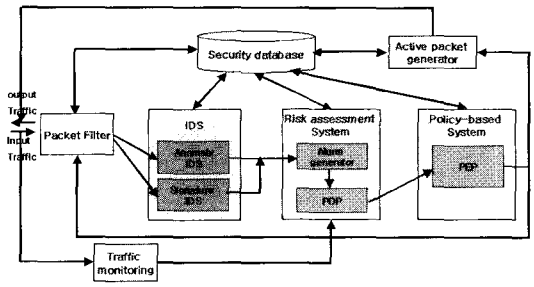
- 공격 탐지 과정에서 발생할 수 있는 시스템의 오버헤드를 줄임
- 시그네처 탐지 기법과 악의 탐지 기법의 접목을 통한 공격 대응방안 향상
- 분산 패킷 필터링을 통해 DDoS 공격시 victim host의 오버헤드를 줄임
- 액티브 네트워크를 이용하여, 서로 다른 도메인간의 공격에 대응할 수 있는 방안 제공
- 정책기반 보안 기법을 접목하여 네트워크 공격에 대하여 유연성 제공

위와 같은 사항을 고려하여 시스템을 설계하였으며, 다음절에서는 제안하는 액티브 네트워크 환경에서의 침입탐지 시스템을 소개하고자 한다.

3.2 제안 구조

3.2.1 악의 탐지 기법과 시그네처 탐지기법을 접목한 제안구조

지금까지 기존 침입탐지 시스템의 소개를 통해 악의 탐지[18] 기법과 시그네처 탐지 기법의 특성을 살펴보았다. 앞서도 소개되었듯 악의 탐지 기법은 공격정보를 수집하고 이를 침입탐지 필터에 적용함으로써 공격 정보의 주기적인 업데이트가 필요없는 장점이 있는 반면 고려될 만한 트래픽에서 공격이 행해진다면 공격으로 인식하지 못하고 오류를 발생하는 단점이 있다. 또한 시그네처 탐지기법은 트래픽 양에 상관없이 공격을 탐지하지만 공격정보의 주기적인 업데이트가 필요하다. 본 절에서는 이러한 특성을 통해 액티브 네트워크 환경에서 시스템 성능 향상과 보안성 향상을 위한 IDS의 기본 구조를 소개하고자 한다.



(그림 3-1) 액티브네트워크 환경에서 보안을 위한 침입탐지 시스템 구조

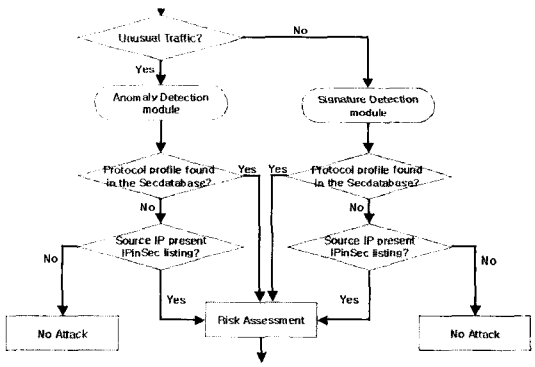
먼저 그림 3-1은 제안된 침입탐지 시스템의 구조를 보여주고 있다. 제안하는 구조는 Packer filter, security 데이터베이스, Active packet generator, Policy-based security 등으로 구성되어 있다. 여기에서 특징적으로 살펴보아야 할 부분은 IDS 시스템에 Anomaly 기법과 시그네처기법을 따로 적용하였고, Active packet generator를 통해 서로 다른 도메인간 공격정보를 공유할 수 있도록 하였다는 점이다.

그럼 제안된 구조의 컴포넌트에 대해 살펴보도록 하겠다.

- Packet filter: victim Host로 유입되는 패킷중 DDoS 공격 패킷과 같은 시스템 성능 저하를 유발하는 패킷을 우선적으로 차단하는 기능을 하며 동시에 시스템으로 유입되는 트래픽을 분류, 각 트래픽 정보를 security 데이터베이스로 전달하는 기능을 한다.
- IDS system: 침입탐지 기능을 하며 트래픽 양에 따라 Anomaly IDS와 시그네처 IDS로 분류, 침입탐지 과정에서 생길 수 있는 시스템 성능 저하를 줄이기 위한 구조이다. 가령 시스템 내부로 유입하는 패킷의 양이 많은 경우 시그네처 Filter를 사용하게 되면 attack profile을 매칭시키는 과정에서 많은 양의 패킷을 매칭해야하므로 시스템 성능 저하를 일으킬 수 있다. 이러한 경우 Anomaly IDS를 사용하여 많

은 트래픽 양을 처리할 수 있도록 하였다.

- Security 데이터베이스: 보안 관련 정보를 저장하는 데이터베이스 server. 또한 시스템으로 유입하는 패킷에 대한 정보를 저장하고, 저장된 정보를 Anomaly Filter에 self-update 하도록 한다.
- Risk assessment system: 침입탐지 과정에서 침입이 탐지된 경우 Alarm generatork에서 알람을 생성하고, PDP에서 침입에 따른 적절한 정책을 결정 Policy-based system으로 보내게 된다.
- Active packet gnerator: 서로 다른 도메인간 보안 정보를 공유하기 위해 Active packet을 생성하는 component이다. 가령 다른 도메인에 있는 zombie Host, handler에 대한 정보나 Attack profile을 알려주어 다른 도메인 서버가 공격정보를 통해 보안기능을 설정할 수 있다.
- Traffic monitoring: 실시간 네트워크 트래픽 모니터링 기능을 하며, 이를 통해 Packet Filter에서 유연성 있는 보안기능을 수행하게 된다.



(그림 3-2) 제안하는 구조에서 DDoS 공격을 탐지하기 위한 순서도

그리고 그림 3-2의 시퀀스 다이어그램은 제안하는 IDS 구조에서 Anomaly Filter와 시그네처 Filter를 이용한 침입탐지 과정도의 예이다.

먼저 시스템으로 유입되는 패킷의 양을 비교하여 패킷의 양이 많은 경우, 악의 탐지 모듈을 사용하도록 하였고, 패킷의 양이 많지 않아 시스템 성능저하에 지장이 없는 경우에는 시그네처 Detection module을 사용하여 악의 탐지 기법의 단점인 고려될 만한 수준의 공격에 대해서도 보다 정교하게 침입탐지를 할 수 있도록 하였다.

여기에서 그림 3-2는 일반적인 DDoS 공격을 탐지하기 위한 예이며, 새로운 공격 방식의 출현으로 보다 정교한 탐지 과정을 필요로 하는 경우에는 공격 정보의 업데이트를 통해 필요한 사항들은 추가가능하다.

4. 정책기반 보안을 위한 정책기반 템플릿

본 장에서는 네트워크 공격과 관련하여 침입탐지 시스템이 공격을 탐지한 경우 정책기반 보안에 사용되어지는 정책템플릿과 각각의 정책오브젝트 예를 보여주도록 한다. 본 기고에서 정책기반 네트워크관리는 침입탐지 시스템의 Alarm Generator와 Risk Assessment System 역할을 하며, 또한 SNMP와 같은 망 관리 프로토콜과의 확장성을 위함이다. 또한 정책을 구현함에 있어 XML의 사용으로 확장성과 보안성, 그리고 유연성 같은 여러 장점들을 활용할 수 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<xsd:element name="Policy_Schema">
        <xsd:complexType>
        <xsd:sequence>
        <xsd:element ref="Dtech"/>
        <xsd:element ref="Event"/>
        <xsd:element name="Condition" type="comment"/>
        <xsd:element name="Action" type="Direct"/>
        </xsd:sequence>
        </xsd:complexType>
</xsd:element>
                <xsd:complexType name="comment">
                <xsd:sequence>
                <xsd:element name="Pamout" type="xsd:positiveInteger"/>
                <xsd:element name="IPinsec" type="xsd:string"/>
                <xsd:element name="RiskType" type="xsd:string"/>
                </xsd:sequence>
                </xsd:complexType>
                <xsd:complexType name="Direct">
                <xsd:sequence>
                <xsd:element name="Target" type="xsd:string"/>
                <xsd:element name="data" type="xsd:string"/>
                </xsd:sequence>
                </xsd:complexType>
</xsd:schema>
```

(그림 4-1) XML 정책 스키마의 예

그림 4-1은 XML 정책 템플릿의 예를 보여주고 있다. Policy template에 적용된 각각의 요소 정의는 다음과 같다.

- Dtech : 침입탐지 시스템에서 사용된 공격 탐지 기법을 나타낸다.
- Event : 공격의 유형을 나타낸다.
- Condition : 공격에 대한 위험도와, 세부적인 공격정보를 나타내며 PDP에서 적용가능 한 정책을 결정

하도록 한다.

- Action : 공격 유형과 위험도에 따라 PDP에서 결정하는 정책들을 보여준다.

그리고 다음의 표는 이러한 정책기반 네트워크에서 침입탐지 리소스들을 보여주고 있다. 검출 기법에서 침입탐지 관련 사용된 탐지 기법을 알려주고 이에 따른 공격 유형을 선택하여 위험도에 따라 동작을 취하도록 설계되어 있다. 가령 Target element에서 적용하고자 하는 대상을 결정하여 data element에서 적절한 대응법을 적용하게 된다. 표에 나타나 있는 리소스들은 각각 객체들로 구현되어 확장가능하며, 새로운 보안사항들이 요구될 경우 정책 관리자가 보안 사항에 따라 적절한 리소스 객체들을 구현할 수 있다.

〈표 2〉 정책기반 네트워크의 침입탐지 리소스

Detection Technique	Event	Condition	Action	
			Target	Data
anoTech	buffOver intAttack dosAttack	aVerify aSuspi highRisk midRisk lowRisk	rateFilter tarDomain vicHost siteMan	pRemoval geAlarm actPacGen
sigTech	intAttack			geAlarm pRemoval

표 3은 침입탐지 리소스에서 검출 기법과 관련한 사항에 대해 설명하고 있다.

〈표 3〉 침입탐지 리소스의 검출기법

리소스명	설 명
anoTech	공격 탐지시 유입하는 패킷양에 따라 패킷양이 많은 경우, 악의 탐지 탐지 기법을 적용하기 위한 resource
sigTech	공격 탐지시 패킷양이 적은 경우, 시그네처 Detection 탐지 기법을 적용하기 위한 resource

그리고 표 4는 공격유형에 대한 리소스들을 설명하고 있다. 여기에서 공격 유형은, 액티브 패킷을 통하여 서로 다른 도메인간 공격정보의 공유에 사용될 수 있다.

〈표 4〉 침입탐지 리소스의 공격유형

리소스명	설 명
buffOver	시스템 성능 저하를 일으킬 수 있는 Buffer overflow 공격인 경우 발생되는 resource
dosAttack	DDoS 공격시 발생되는 resource
intAttack	문자열 침입과 일반적인 침입시 발생되는 resource

표 5와 6은 공격에 대한 대응법과 각각에 해당되는 요소들을 나타내고 있다.

〈표 5〉 침입탐지 리소스의 네트워크 공격에 대한 대응(1)

리소스명	설 명
rateFilter	DDoS 공격이나 Buffer overflow 공격시, 시스템으로 유입하는 패킷들을 신속히 폐기하기 위해 Rate Filter에 공격에 대응하기 위한 명령을 보냄
tarDomain	다른 Domain으로부터 유입하는 공격 패킷들을 처리하고 zombie Host를 처리하기 위해 해당 Domain Server에 Active packet을 통해 공격 정보를 전달하게 된다.
vicHost	공격 탐지시 Victim Host에게 알람을 발생한다.

〈표 6〉 침입탐지 리소스의 네트워크 공격에 대한 대응(2)

리소스명	설 명
pRemoval	DDoS공격과 Buffer overflow 공격시 Rate Filter에서 Attacker로부터 유입하는 패킷들을 신속히 제거하기 위한 명령
actPacGen	공격 정보를 다른 Domain Server에게 전달하기 위한 Active 패킷 생성 명령
geAlarm	공격 탐지시 victim Host에게 알람을 생성하기 위한 명령

5. 운용 시나리오

5.1 XML 정책 템플릿의 예

본 장에서는 침입탐지 과정에서 네트워크 공격을 탐지한 경우 생성되는 XML 정책 템플릿을 보여주고

정책 템플릿을 통해 다른 도메인으로 전달하게 될 액티브 패킷의 예를 보여주도록 한다.

먼저 그림 5-1은 네트워크 공격이 탐지된 경우 XML 정책 템플릿이 생성된 예를 보여주고 있다.

그림에서 나타내고 있는 정보는 다음과 같다.

"악의 탐지로 DDoS 공격을 탐지한 경우, 한 근원지 주소로부터 들어오는 패킷양이 1200이 넘어 시스템 오버헤드를 발생시킬 수 있으므로 Rate Filter에서 시스템으로 유입하는 공격패킷을 바로 폐기한다."

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Security_policy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="D:\XML\schema.xsd">
  <DTech>anoTech</DTech>
  <Event>dosAttack</Event>
- <Condition>
    <Pamount>1200</Pamount>
    <IPinsec>no</IPinsec>
    <RiskType>highRisk</RiskType>
  </Condition>
- <Action>
    <Target>rateFilter</Target>
    <data>pRemoval</data>
  </Action>
</Security_policy>
```

(그림 5-1) 네트워크 공격에 대한 XML policy 템플릿 생성예

5.2 액티브 패킷의 생성 예

그림 5-2는 HTTP_request 메시지를 통해 공격을 가해온 경우, 다른 도메인 서버에게 공격 정보를 알려주기 위한 Active Packet의 예를 보여주고 있다.

그림의 액티브 패킷은 공격의 근원지 주소와 사용된 공격 유형을 알려주어 해당 도메인 서버에서 Attacker나 Zombie Host들을 관리할 수 있도록 하였다.

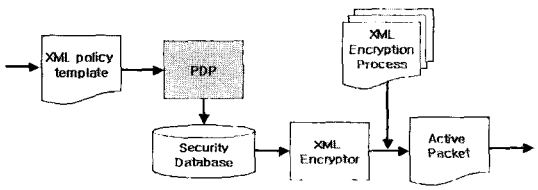
```
<?xml version="1.0" encoding="UTF-8" ?>
- <Attack_Profiles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="C:\xmldir\profiles.xsd">
  <IPinSec>130.110.x.x</IPinSec>
  <Attack_pattern>HTTP_request</Attack_pattern>
  <Message>GET/Scripts/..\%C0\%AF../winnt/system32/cmd.exe?/c+dir+c:\</Message>
  <MessageType>client</MessageType>
  <FieldCounter0>Ethernet/IP/TCP 0 GET=</FieldCounter0>
  <FieldCounter1>Ethernet/IP/TCP 1 /scripts/..\%C0\%AF../winnt/system32/cmd.exe?
    c+dir+c:\=</FieldCounter1>
</Attack_Profiles>
```

(그림 5-2) Active packet 생성예

그리고 그림 5-3는 생성된 액티브 패킷의 보안성을 위해 XML 암호화 알고리즘을 적용한 그림이다. 이를 통해 다른 도메인으로 공격정보를 전달하는 과정에서 발생할 수 있는 보안상 취약점을 해결할 수 있다. XML 암호화는 SUN J2sdk1.4.1, Apache Xerces 2.3.0, Xalan2.4.1, IBM alphaworks의 XSS4j를 이용 작업 수행되었으며 그림에서 메시지 요소를 암호화 하였다.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Attack_Profiles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="C:\xmldir\profiles.xsd">
  <IPinSec>130.110.x.x</IPinSec>
  <Attack_pattern>HTTP_request</Attack_pattern>
- <EncryptedData Id="edi" Type="http://www.w3.org/2001/04/xmlenc#Element"
  xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
- <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
- <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
- <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
  <KeyName>Alice</KeyName>
  </KeyInfo>
- <CipherData>
  <CipherValue>qEqQGDHVI+IOtQM16UoyyfjUMNCjqSFTw9DyJA/uSfH60JSCPGr/nQEQoMEhns
  </CipherData>
  </EncryptedKey>
  </KeyInfo>
  <CipherData>
  <CipherValue>XLq8pNIt/pbX2ZGjYyuIF91rR1DNJnXsE1DqK/6Si1r/CaulPnfhPLDxHdrdJqYv%pSt
  </CipherData>
  </EncryptedData>
  <MessageType>client</MessageType>
  <FieldCounter0>Ethernet/IP/TCP 0 GET=</FieldCounter0>
  <FieldCounter1>Ethernet/IP/TCP 1 /scripts/..\%C0\%AF../winnt/system32/cmd.exe?
    c+dir+c:\=</FieldCounter1>
</Attack_Profiles>
```

(그림 5-3) Active packet 보안을 위한 XML 암호화 적용예



(그림 5-4) Active packet 생성과정도

액티브 패킷을 암호화 하기 위한 과정은 위의 그림과 같다. XML 정책 템플릿으로 생성된 정책에서 액티브 패킷 생성 명령이 있는 경우, PDP에서는 security 데이터베이스에 있는 보안정보들을 추출하여 XML 암호화 과정을 거치고 최종적으로 액티브 패킷을 생성하게 된다.

6. 결론

본 기고의 목적은 액티브 네트워크 환경에서 보안성 향상과 시스템 성능향상의 방안을 제시하는 것이다. 기존 침입탐지 시스템의 소개를 통해 악의 탐지 기법과 시그네처 탐지 기법의 특성을 살펴보았고, 정

책기반 네트워크를 위한 정책 예들에 대해서 살펴보았다. 즉 악의 탐지 기법은 공격 정보를 수집하고 이를 침입탐지 필터에 적용함으로써 공격 정보의 주기적인 업데이트가 필요 없는 장점이 있는 반면 고려될 만한 트래픽에서 공격이 행해진다면 공격으로 인식하지 못하고 오류를 발생하는 단점이 있고, 시그네처 탐지 기법은 트래픽 양에 상관없이 공격을 탐지하지만 공격정보의 주기적인 업데이트가 필요하며 네트워크 트래픽이 많은 경우 시스템 성능이 떨어질 수 있다. 이를 위해 제안하는 구조에서는 먼저 시스템으로 유입되는 패킷의 양을 비교하여 패킷의 양이 많은 경우, 악의 탐지모듈을 사용하도록 하였고, 패킷의 양이 많지 않아 시스템 성능저하에 지장이 없는 경우에는 시그네처 탐지 모듈을 사용하여 악의 탐지 기법의 단점인 고려될 만한 수준의 공격에 대해서도 보다 정교하게 침입탐지를 할 수 있도록 하였다. 그리고 정책기반 네트워크를 적용함으로써 네트워크 공격에 대해 유연성과 확장성을 제공할 수 있으며, 또한 액티브 네트워크를 이용하여 서로 다른 도메인간 보안 정보를 공유하고, zombie Host나 공격자에 대한 정보를 공유함으로써 공격에 대해 보다 신속히 대응할 수 있는 장점을 가질 수 있으리라 판단된다.

향후 연구 과제로는 액티브 패킷을 사용하여 zombie Host나 공격자 검출을 위한 정책 및 각각의 엘리먼트에 대한 연구가 필요하며, Protocol Anomaly 기법 적용시 보안 요구사항들에 대한 연구가 필요하리라 판단된다.

참고문헌

[1] S.Murphy, E.Lewis, R.Puga, R.Watson, and R.Yee, "Strong Security for 액티브 네트워크s", IEEE, 2001

[2] Kou Yanan, Li Zengzhi, Liao Zhigang, "A Prototype of Security for 액티브 네트워크s", International Conference on Algorithms and Architecture for Parallel Processing, 2002

[3] Snort The Open Source Network Intrusion Detection System. http://www.snort.org/

[4] NFR Security. http://www.nfr.net/

[5] V.Paxson.Bro: A System for Detection Network Intruders in Real-time, Computer Networks, 31 Dec 1999.

[6] G.Vigna, S.T.Eckmann, and R.A.Kemmerer, The STAT Tool Suite, In Proceedings of DARPA Information Survivability Conference & Exposition (DISCEX 2000), 2000.

[7] Luciano Paschoal Gaspary, Edgar Meneghetti, Liane Rockenbach Tarouco, "An SNMP 에이전트 for stateful Instusion Inspection", Integrated Network Management VIII 2003.

[8] D.Alessandri, "Using Rule-based Activeity Descriptions to Evaluate Intrusion -Detection Systems", In Proceedings of International Workshop on the Recent Advances on Intrusion Detection (RAID 2000), 2000.

[9] Erwan Lemonnier, "Protocol 악의 탐지 in Network-based IDSs" Defcom 28th June 2001.

[10] Kumar Das, "Protocol 악의 탐지 for Network-based Intrusion Detection", GSEC Practical Assignment version 1,2f , August 13 2001.

[11] J.B.D.Cabrera, L.Lewis, X.Qin, W.Lee, R.K.Prasanth, B.Ravichandran, and R.K.Mehra, "Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables-a Feasibility Study", In Proceedings of IFIP/IEEE International Symposium on Integrated Management(IM2001), 2001.

[12] X.Qin, W.Lee, L.Lewis, and J.B.D.Cabrera, "Using MIB II Variables for Network Intrusion Detection." Data mining for Security Applications, Advances in Computer Security. Kluwer Academic Press, March 2002.

[13] L.P.Gaspary, L.F.Balbinot, and L.R. Tarouco, "Monitoring High-Layer Protocol Behavior Using the Trace Architecture", In Proceedings of Latin American Network Operation and Management Symposium, 2001.

[14] Kai Hwang, Sapon Taanachaiwiwat, Pinalkumar

Dave, "Proactive Intrusion Defense Against DDoS Flooding Attacks", IEEE Security & Privacy Magazine, April 14, 2003.

[15] Kai Hwang, Pinalkumar Dave, Sapon Tanachaiwiwat, "NetShield: Protocol 악의 탐지 with Datamining Against DDoS Attacks", RAID 2003 the Sixth International Symposium on Recent Advances in Intrusion Detection, Mar 2003.

[16] Temitope Olukemi, Ioannis Liabotis, Ognjen Prnjat, Lionel Sacks, "Security and Resource Policy-based Management Architecture for ALAN servers", Net-Con'2002 - IFIP and IEEE Conference on Network Control and Engineering for QoS, Security and Mobility, Paris, France, 2002.

[17] Ognjen Prnjat, Ioannis Liabotis, Temitope Olukemi, Lionel Sacks, "Policy-based Management for ALAN-Enabled Networks", IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[18] Jonathan Werrett, "Review of Anomaly -based Network Intrusion Detection", 26th May 2003

◐ 저 자 소 개 ◑

홍 충 선

1983년 경희대학교 전자공학과 (학사)
1985년 경희대학교 전자공학과 (석사)
1997년 Keio University, Department of Information and Computer Science (박사)
1988년~1999년 KT 통신망연구소 선임 연구원/ 네트워킹연구실장
1999년~현재 경희대학교 전자정보학부 조교수
관심분야 : 인터넷 서비스 및 망 관리 구조, 모바일 IP, 인터넷보안

허 용 준

2002년 경희대학교 전자공학과(학사)
2004년 경희대학교 전자공학과(석사)
2004년~현재 그린정보통신(주) 연구원
관심분야 : 정책기반 보안관리, XML보안