情報保護學會論文誌
第14券 第6號, 2004. 12

암호화 기법을 적용한 침입 탐지 시스템의 룰 보호 기법

손 형 서†, 김 현 성‡, 부 기 동
경일대학교 컴퓨터공학부

A Rule Protecting Scheme with Symmetric Cryptosystem for Intrusion Detection System

Hyung-Seo Son†, Hyun-Sung Kim‡, Ki-Dong Bu
School of Computer Engineering, Kyungil University

요 약

논문[10]에서는 유비쿼터스 환경에서 보안 시스템들의 정책들을 보호하기 위해 단방향 함수를 사용한 룰 보호 기법을 제시 하였으며, 논문[5-6]은 침입 탐지 시스템 중 Snort를 기반으로 해쉬 함수를 사용한 룰 보호기법을 제안하였다. 이러한 기법들을 통해 보안 시스템들의 정책을 보호할 수 있었으나 단방향 함수의 특성상 정책의 모든 부분을 보호할 수는 없었다. 이러한 문제를 해결하기 위해 본 논문에서는 Snort를 기반으로 대칭키 암호 시스템을 이용한 새로운 기법을 제안한다.

정책의 암호화 및 복호화에 사용되는 비밀키의 유출을 예방하기 위해 논문[12]에서 제안한 PCMCIA 암호 모듈을 사용한 키 관리 기법을 사용한다. 본 논문에서 제안한 기법은 일반적인 정책기반의 보안 시스템에 적용될 수 있다.

ABSTRACT

Kvarnstrom et al. in[10] proposed a rule protection scheme by using one-way hash function to protect rules in security systems over ubiquitous environment. Son et al. in[5-6] also proposed a rule protection scheme for Snort, which is one of the most common IDS. These schemes provide security only for the header information but not for its contents. To solve this problem, this paper presents a scheme based on the symmetric cryptosystem over Snort not only for the header information but also contents.

This paper uses the key management based on PCMCIA security module proposed by[12] for the symmetric cryptosystem. Our scheme could be adjusted to other security systems, which use the rule based detection.

Keywords : *Intrusion Detection System, Rule Protection Mechanism, Symmetric Cryptosystem*

Ⅰ. 서 론

침입 탐지 시스템(Intrusion Detection System)은 호스트나 네트워크의 데이터를 감시하여 자동으로 침입을 탐지하는 시스템이다. 이러한 시스템은 공격이 탐지되면 이를 시스템 관리자에게 알리고 시스템 관리자는 이에 따른 대응을 수립한다. 이러한 침입탐지 시스템은 크게 오용탐지(misuse detection) 시스템과 이상탐지(anomaly detection) 시스템으로 나눌 수 있다. 네트워크 기반의 오용탐지 시스템은 룰(네트워크 전문가에 의해서 선정된 몇 가지 특성을

접수일 : 2004년 5월 7일 ; 채택일 : 2004년 12월 6일
† 주저자 : hyeongseo@gmail.com
‡ 교신저자 : kim@kiu.ac.kr

추출해서 만든 정책)을 미리 생성하여 이를 기반으로 침입을 탐지한다[1-4]. 하지만, 이러한 시스템에서 만약 룰들이 공격자에게 노출된다면 공격자는 룰을 우회한 공격을 하게 되고 그 침입 탐지 시스템은 그러한 공격에 대한 탐지를 제대로 수행하지 못할 것이다.

논문[5-6]에서는 잘 알려진 공개 침입 탐지 시스템인 Snort의 룰을 보호하기 위해 해쉬 함수를 이용한 룰 보호기법을 제안하였다. 또한, 논문[10]에서는 유비쿼터스 환경을 위한 보안 매커니즘의 정책을 보호하기 위해, 단방향 함수를 이용한 정책 보호기법을 제시함으로써 정책과 같은 중요한 정보를 보호할 수 있는 기법을 제안하였다. 하지만 이들 연구에서는 단방향 함수의 특성상 고정값의 속성에 대한 룰 보호는 가능하였지만 가변 값과 범위 값 속성의 룰 보호는 어렵다는 문제가 있었다. 이러한 문제를 해결하기 위해 이들 연구에서는 가변 수렴 및 발산 알고리즘과 같은 기법을 도입함으로써 가변 값과 범위 값의 정보에 대한 단방향 함수 적용의 문제점을 해결하였다. 그러나 패킷의 내용(Contents)을 구성하는 정보들은 데이터의 길이나 형태가 일정하지 않고, 탐지 과정에서 패킷의 데이터 중 특정 부분의 패턴을 비교해야 되기 때문에 기존의 기법을 적용하기 위한 문제점이 여전히 존재한다.

따라서, 본 논문에서는 기존의 룰 보호기법의 문제를 해결하기 위하여 Snort를 기반으로 새로운 룰 보호기법을 제안한다. 제안한 기법은 대칭키 암호화 기법 중 하나인 Triple-DES를 사용함으로써 룰의 헤더 정보뿐만 아니라 패킷의 내용에 대해서도 기밀성 및 무결성을 제공한다. 특히, 본 논문에서 제안한 기법은 Snort 뿐만 아니라 다른 룰 기반의 보안 시스템에도 적용되어 일반화 시킬 수 있다. 그리고, 기밀성과 무결성을 위해 사용된 대칭키 암호화 시스템인 Triple-DES도 AES나 기타 시스템에서 사용되고있는 다른 안전한 암호화 시스템으로 대체 될 수 있다.

제안한 시스템에서는 대칭키 암호화 기법을 사용하기 때문에 암호화 시스템에 사용되는 키 관리기법도 고려되어야 한다. 이를 위해 본 논문에서는 기존의 논문[12]에서 제안된 PCMCIA 암호 모듈을 사용한 키 생성 및 관리 기법을 이용한다. PCMCIA 암호 모듈을 이용한 키 관리 기법을 사용할 경우 암호 모듈 자체에 암호에 사용될 키와 고속 암호칩을 포함함으로써, 키 유출의 위험을 줄일 수 있으며, 암호시스템을 이용함으로서 발생되는 성능저하의 문제도 해결할 수 있을 것이다.

본 논문의 구성은 2장에서 룰 보호 기법들에 관련된 연구를 살펴보고, 본 연구와 관련이 있는 Snort 및 Triple-DES에 대하여 간략히 기술한다. 3장에서는 대칭키 암호화 시스템에 기반한 새로운 룰 보호 기법을 제안하고, 제안한 시스템에서 사용된 키 관리 기법을 설명한다. 그리고 제안한 기법이 적용된 침입 탐지 방법을 제시한다. 4장에서는 시뮬레이션 결과를 제시하고 기존의 연구와 본 논문의 기법들을 비교 및 분석하며, 5장에서 결론을 내린다.

II. 배 경

본 장에서는 보안시스템의 룰을 보호하기 위한 기존의 보호 기법들을 살펴보고 이들의 문제점을 살펴본다. 그리고 본 논문의 대상이 되는 오용탐지 모델에 기반한 공개 침입탐지 시스템인 Snort와 대칭키 암호 시스템인 Triple-DES에 대해서 간략히 기술한다.

2.1 관련 연구

기존의 보안 매커니즘들은 침입으로부터 대상 시스템들에 대한 정보보안을 효과적으로 제시해주지만 보안 매커니즘 그 자체에 대한 보안은 고려되지 못했다. 따라서 보안 정책과 같은 중요한 정보들이 침입자에게 노출될 위험이 있고, 그러한 정보가 노출됨으로써 보안 매커니즘을 우회한 공격에 대한 잠재적인 취약성이 있다.

이러한 문제를 해결하기 위한 해결책들이 논문[5-6,10]에서 제시되었다. 논문[10]에서는 침입 탐지 시스템의 탐지 정책이나 방화벽의 필터링 정책과 같은 보안 정책 및 보안과 관련된 정보들을 보호하기 위해 단방향 함수를 사용한 기법을 제안하였다. 이 연구에서는 룰을 단방향 함수를 사용하여 해쉬를 취함으로서 룰에 대한 정보보호를 제공하였다. 그러나 단방향 함수는 입력 값의 작은 변화도 출력 값에 상당한 영향을 미치는 특성이 있고, 이러한 특성은 일반적인 정보를 보호하는데는 유용하지만, 가변적인 값이나 범위 값을 가지는 정보에 대해서는 적용하는데 어려움이 있었다. 이러한 문제를 해결하기 위해서는 모든 가능한 값들에 대해 별도로 해쉬를 수행할 수 있다. 그러나 이러한 방법은 데이터의 범위가 넓을 경우 효율성이 저하된다. 이러한 문제를 해결하기 위하여 논

문 [10]에서는 가변 수렴 및 발산 알고리즘을 사용하였고, 이를 통해 효율성을 높일 수 있었다. 그러나 패킷의 헤더정보가 아닌 내용(contents)을 구성하는 정보의 경우, 대부분 패턴매칭이 적용되어야 하므로 논문[10]에서 제안한 기법을 적용하는데는 어려움이 따른다. 본 논문의 목적은 패킷의 헤더 정보 및 내용 정보에 대한 정보보호를 제공하기 위한 새로운 기법을 제안하는데 있다.

2.2 Snort

Snort는 오용 탐지 모델 중 잘 알려진 공개 침입 탐지 시스템이다. Snort의 기본구조는 그림 1과 같이 전체 4단계로 구성된다. 먼저, 네트워크 상에서 전송되는 패킷을 캡춰(sniff)하여 필터링(filtering)하고, 필요한 정보를 디코딩(decoding)한 후, 침입탐지엔진(intrusion detection engine)에서 침입 탐지 룰을 기반으로 침입 여부를 탐지하고 경보와 로그(alerting and logging)단계에서 시스템 관리자에게 경고를 보내고 로그를 남기고 레포트를 출력한다[4].

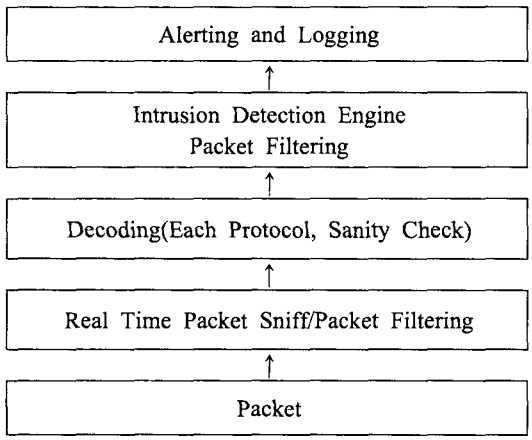


그림 1. Snort의 기본구조

Snort에서 사용하는 룰의 기본 구조는 그림 2와 같다. 본 논문의 목적은 보안 메커니즘의 룰을 보호하는 기법을 제안하는 것이므로, 이러한 룰의 구성을 이해하는 것은 상당히 중요하다.

그림 2. Snort 룰의 기본구조

그림 2에서 Options 필드 이전까지의 값의 정보를 헤더정보(Header)라고 하고, Options 필드 정보를 내용정보(Contents)라고 한다. 다음은 룰의 구성 요소에 대한 개괄적인 설명이다.

- Action : 침입이 탐지 되었을 경우 대응을 위한 행위(action)을 지정
  ex) alert, log, pass, activate, dynamic 등
- Protocol : 사용된 프로토콜을 명시
  ex) TCP/IP, UDP, ICMP 등
- SourceIP, DestinationIP : 패킷의 송신지 및 수신지 지정
  ex) any, 203.230.91.1/32 등
- SourcePort, DestinationPort : 패킷의 포트지정
  ex) 8080, 21 등
- -> : ->를 기준으로 왼쪽이 송신지, 오른쪽이 수신지 정보
  ex) ->, <-, <> 등
- Option : 위의 항목이외의 특정필드나 데이터의 정보를 지정 및 기타 설정
  ex) msg, content, fragbits 등

룰에 유연성(flexibility)을 제공하기 위하여 IP Address, Port 부분은 고정 값(fixed value), 임의 값(wildcard), 가변 값이나 범위 값(variable or interval)으로 표현된다. 실제 Snort 룰의 예는 다음과 같다.

- Destination Port가 고정 값인 경우
  ex) log tcp any any -> 192.168.1.1/32 23
- Source IP와 Source Port가 각각 임의 값인 경우
  ex) log tcp any any <> 192.168.1.1/32 23
- Source와 Destination의 Port가 범위 값을 가진 경우
  ex) log tcp any 23:100 -> any 1024:
- Destination IP가 변수 값인 경우
  ex) alert tcp any any -> $HOME_NET any
- Content 필드를 가지는 경우
  ex) alert ip $EXTERNAL_NET any -> $HOME_NET any (content:"|00 00|");

이렇게 설정된 룰을 기반으로 Snort는 다양한 네

트워크 공격을 탐지한다. 그러나 전술한 바와 같이, 이러한 룰들이 침입자에게 노출된다면 우회공격에 대한 취약성이 존재할 수 있다. 다음 장에서는 이러한 문제를 해결하기 위한 새로운 기법을 제안한다. 특히, Snort의 룰들은 여러 가지 다양한 형태의 속성 값을 가지므로, 룰을 구성하는 속성들의 정보를 분류하고 각 속성에 맞는 룰 보호 기법을 제시한다.

2.3 대칭키 암호 시스템

암호화와 복호화에 모두 같은 키를 사용하는 알고리즘을 대칭키 암호 알고리즘(symmetric key cryptosystem)이라고 한다. 이러한 알고리즘으로는 DES, Triple-DES, AES, SEED등과 같은 여러 기법들이 있으며, 이를 통하여 데이터에 기밀성과 무결성을 제공할 수 있다. 본 논문에서는 이 중 Triple-DES를 사용한다.

Triple-DES는 비교적 널리 알려진 DES에 대한 대안으로서 키 관리 표준 ANS X9.17과 ISO 8732에 의해 채택되었으며 전사공격에 대한 DES의 잠재적인 취약점을 해결하기 위해 세 개의 다른 키로 세 번의 암호화 과정을 사용하는 것이다. 이것은 알려진 평문 공격의 비용을 2112로 크게 한다. 그러나 3×56=168의 다소 큰 키 길이를 요구한다는 결점을 가진다[7]. 선택적으로, Tuchman은 두개의 키를 사용하는 Triple DES를 제안하였다. 그 함수는 암호화-복호화-암호화(EDE)순서로 다음과 같다.

$$C = E_{K_1}[D_{K_2}[E_{K_1}[P]]]$$

본 논문에서는 룰의 기밀성과 무결성을 제공하기 위해서 두개의 키를 사용하는 Triple-DES를 이용한다.

Ⅲ. 룰 보호기법

본 장에서는 보안시스템의 룰을 보호하기 위한 새로운 기법을 제안한다. 본 논문에서는 대칭키 암호화 시스템 중의 하나인 Triple-DES를 이용하여 잘 알려진 공개용 침입탐지 시스템인 Snort의 룰을 보호할 수 있는 새로운 기법을 제안한다. 본 논문에서 제안한 시스템의 전체적인 구성은 그림 3과 같다. 이 시스템은 크게 룰에 기밀성과 무결성을 적용하는 Rule 부분과 네트워크에 입력되는 패킷을 필터링하는

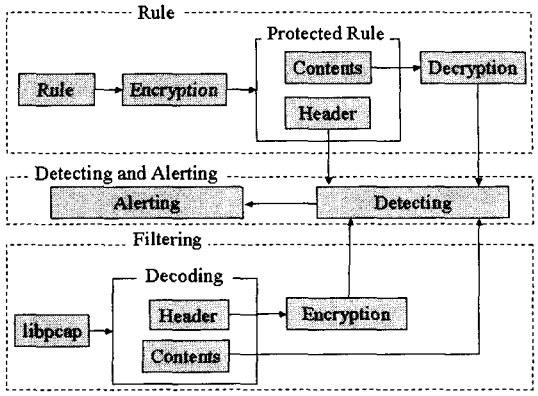


그림 3. 시스템 전체 구성

Filtering 부분, 그리고 침입의 여부를 탐지하고 경고를 출력하는 Detecting and Alerting 부분으로 구성된다. 패킷의 정보와 룰은 프로토콜, 송수신지 주소, 송수신지 포트 번호로 구성된 Header 부분과 실제 데이터를 포함하는 Contents 부분으로 구성된다.

침입탐지시스템의 룰을 보호하기위해서는 룰을 암호하고 패킷이 도착하면 암호화된 룰을 복호하여 침입인지의 여부를 확인하는 방법이 가장 쉬운 방법이다. 그러나 이러한 방법은 매번 패킷이 입력될 때마다 모든 룰이 복호되어야 하기 때문에 시스템에 많은 부하가 발생한다. 이러한 문제를 해결하기 위하여 본 논문에서 제안하는 시스템에서는 침입탐지를 위하여 룰을 암호하고 네트워크에 패킷이 발생하면 이 패킷의 헤더정보는 암호하여 저장된 암호된 룰과 비교하고, 내용정보는 복호된 룰과 비교함으로서 침입을 검사한다.

본 장에서는 먼저 룰을 보호하기 위한 암호학적 요구 사항에 대해서 살펴보고 이러한 요구사항을 만족하는 새로운 룰 보호 기법을 제안한다. 이를 위해서 먼저 2장에서 살펴본 Snort의 룰을 종류별로 분류하고, 각 분류에 따른 보호 기법을 제시한다. 그리고 대칭키 암호 시스템에서 사용된 키의 관리 기법에 대해서 살펴본다. 마지막으로, 제안한 기법을 기반으로 실제 침입탐지 시스템에서의 동작 과정을 보인다.

3.1 보안 요구사항

침입탐지 시스템의 룰을 보호하기 위해서는 기밀성(Confidentiality)과 무결성(Integrity)이 제공되어야 한다. 이를 위한 보안 요구 사항은 다음과 같다[7].

- **기밀성**은 정보의 소유자가 원하는 대로 정보의 비밀이 유지되어야 한다는 원칙이다. 정보는 소유자의 인가를 받은 사람만이 접근할 수 있어야 하며, 인가되지 않은 정보의 공개는 금지 되어야 한다. 기밀자료는 비밀성이 노출되지 않도록 반드시 인가된 자에 의해서만 접근이 가능해야 한다.
- **무결성**은 정보는 정해진 절차에 따라, 그리고 주어진 권한에 의해서만 변경되어야 한다는 것이다. 정보는 항상 정확성을 일정하게 유지하여야 하며, 인가 받은 방법에 의해서만 변경되어야 한다. 무결성을 보장하기 위한 정책에는 정보 변경에 대한 통제뿐만 아니라 오류나 태만 등으로부터의 예방도 포함되어야 한다. 즉, 정보는 의도적이던, 우발적이던 간에 허가 없이 변경되어서는 안 된다.

본 논문에서는 보안 시스템의 룰에 이러한 보안 요구사항을 제공하기 위하여 대칭키 암호시스템의 하나인 Triple-DES를 사용한다.

3.2 보호된 룰 생성

본 절에서는 룰에 기밀성과 무결성을 제공하기 위한 기법을 제안한다. 이를 위해 먼저 룰을 구성하는 데이터의 특성에 따라 룰을 분류하고, 이에 따른 보호기법을 제안한다. 그림 4는 보호된 룰의 기본 형식을 보여준다. 보호된 룰은 필드의 특성에 따른 분류를 위해 추가적인 Flag 필드를 사용한다. 마지막 필드인 Contents 필드는 항상 고정 값을 가지므로 Flag 필드는 두지 않는다.

F	Src IP	F	Src Port	F	Dst IP	F	Dst Port	Contents

-F : Flag -Src : Source -Dst : Distination

그림 4. 보호된 룰의 형식

2.2절에서 제시되었던 룰의 필드는 그 필드를 구성하는 데이터의 형태에 따라 크게 다음과 같이 세 가지 형태로 분류할 수 있다.

- 고정 값을 갖는 필드
- 범위 값을 갖는 필드
- 가변 값과 임의 값을 갖는 필드

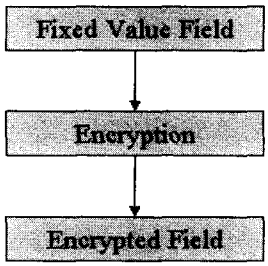


그림 5. 고정 값을 갖는 필드의 룰 생성과정

각각의 속성에 따른 처리 방법은 다음과 같다.

가. 고정 값을 갖는 필드

룰을 생성하는데 있어서 고정 값의 필드는 그림 5와 같이 바로 암호화를 적용한다.

나. 범위 값을 갖는 필드

범위 값을 갖는 필드의 경우 Snort에서는 룰 포맷에 정의된 특수한 형태로 기술된다. 따라서 필드의 값을 암호 할 경우, 그 결과 값은 실제 룰이 가지는 의미를 잃을 수 있다. 따라서 룰의 범위에 속하는 모든 값에 대한 암호화가 필요하다. 하지만 이러한 방법은 룰의 범위가 넓을수록 룰의 개수가 증가하고 효율성이 떨어진다. 이러한 문제를 해결하기 위해서 범위 값을 갖는 필드에 대해서 하나의 대표 값으로 대응시키기 위해 가변수렴 알고리즘과 이를 확인하기 위한 가변발산 알고리즘을 이용한다[9].

```
가변수렴(MinValue, MaxValue){
  Interval = MaxVal - MinVal + 1;
  CommitVal1 = Random() * Interval;
  DecommitVal = MinVal - CommitVal1;
  return Interval, CommitVal1, DecommitVal;
  }
```

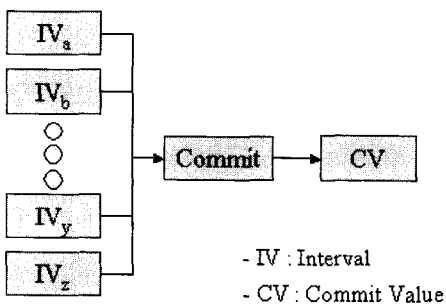


그림 6. 가변수렴 알고리즘의 동작방식

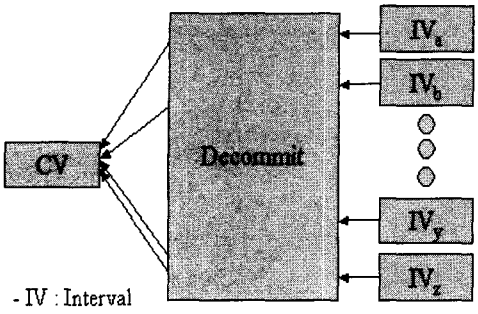


그림 7. 가변발산 알고리즘의 동작방식

```
가변발산(CaptureNum, Interval, DecommitVal){
  CommitVal2=CaptureNum-DecommitVal;
  CommitVal2 -= (CommitVal2 % Interval);
  return CommitVal2;
 }
```

다음은 가변수렴과 가변발산 알고리즘의 이해를 돕기 위한 예제이다.

예제 - 룰에서 IP 범위가 203.230.91.200~203.230.91.220로 설정되어 있고, 네트워크 상에서 캡춰한 패킷의 IP가 203.230.91.205일 경우

(1) 가변 수렴: IP의 범위는 220 - 200 + 1 = 21이므로 Interval은 21이 된다. 21의 정수배 값을 가지고, Random()에의해 선택된 값이 10이라고 가정하면 CommitVal1은 210이 된다. DecommitVal은 범위의 최소값(MinVal)과 CommitVal1의 차(뺄셈연산)로 계산되므로 -10이 된다.

(2) 가변 발산: 입력된 패킷에서 체크될 범위에 해당하는 IP인지 확인하기 위해서는 입력된 IP인 203.230.91.205와 가변수렴에서 계산된 DecommitVal을 이용하여 CommitVal2 = CaptureNum - DecommitVal = 205 - (-10) = 215를 구한다. 그리고 CommitVal2 = CommitVal2 - {(CommitVal2% Interval)} = 215 - (215 % 21) = 210을 계산한다.

이렇게 생성된 CommitVal1과 CommitVal2가 일치하는지 확인함으로서 패킷의 주소가 룰이 가지는 주소 범위에 포함되는지의 여부를 확인할 수 있다. 위의 예제에서는 패킷의 주소 203.230.91.205는 CommitVal1과 CommitVal2가 동일한 값(210)을 가지므로 패킷의 값이 룰의 주소 범위에 포함됨을 확인할 수 있다. 범위 값을 갖는 필드는 그림 8과 같은 처리과정을 통하여 단일한 값으로 매핑 시킨 후 암호화를 수행한다.

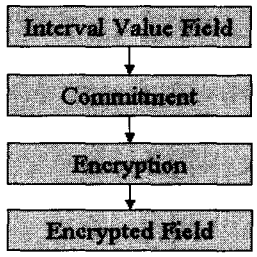


그림 8. 범위 값을 가진 필드의 룰 생성과정

다. 가변 값과 임의 값을 갖는 필드

가변 값의 필드는 각 가변 값이 가질 수 있는 값들을 하나의 룰로 변환하여 고정 값의 필드일 경우와 같은 형태로 Triple-DES를 적용한다. 또한, 임의 값을 갖는 필드의 경우에는 모든 값이 허용되어야 하므로 그 필드를 공백으로 두어서 처리한다.

이러한 세 가지 형태의 필드처리 방법을 통하여 룰에 기밀성과 무결성을 제공할 수 있다. 본 논문에서 제안한 보호된 룰 생성기법을 이용한 시스템의 전체적인 룰 생성과정은 그림9와 같다. 본 논문에서 제안한 룰 보호기법을 통하여 다양한 속성의 룰

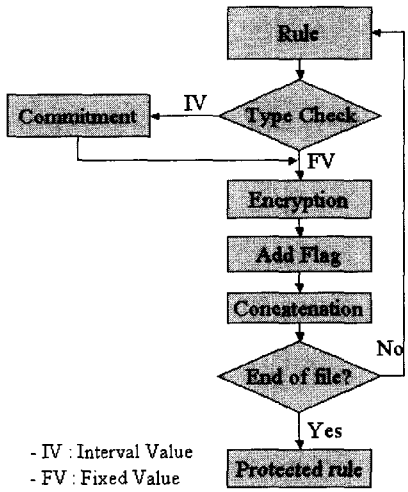


그림 9. 전체 룰 생성과정

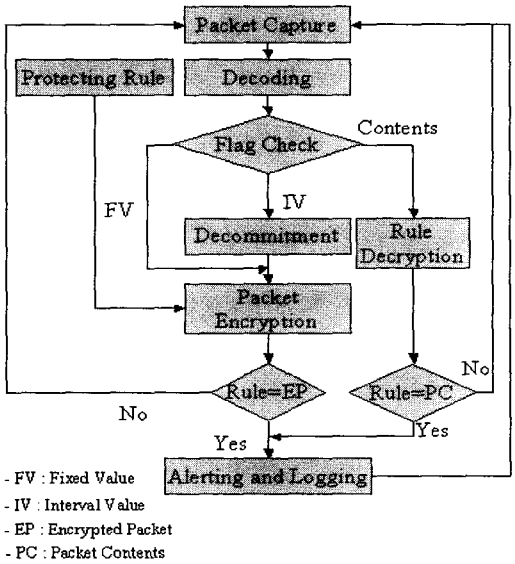


그림 10. 침입 탐지 시스템의 침입 탐지과정

기반 보안 시스템의 룰을 효율적으로 보호할 수 있을 것이다.

3.3. 침입 탐지

본 절에서는 제안한 룰보호 기법이 적용된 침입 탐지 시스템의 처리과정을 제시한다. 전체적인 수행 과정은 그림 10과 같다. 전체적인 시스템의 처리과정은 룰보호 기법을 적용할 때와 마찬가지로 패킷을 구성하는 속성에 따라 세 가지 형태로 나누어 처리한다.

본 논문에서 제안한 룰보호 기법이 적용된 침입 탐지 시스템의 처리과정은 다음과 같다.

(과정1): 수집된 패킷에서 필요한 데이터를 추출한다.

(과정2): 룰의 Flag를 보고 3.2절에서 제안한 바와 같이 룰을 분류한다. 범위를 가지는 값을 갖는 필드일 경우엔 추출된 데이터에 가변발산 알고리즘을 적용한다. 패킷의 내용 필드일 경우 룰을 복호한다.

(과정3): (과정2)의 결과값을 암호화 한다. 단, 패킷의 내용일 경우 암호화 하지 않는다.

(과정4): 룰의 해당 필드의 값과 (과정3)의 결과값을 비교하여 일치할 경우 경고 메시지를 출력한다.

처리과정에서 살펴본 바와 같이, 본 논문에서 제안한 룰 보호기법은 패킷의 헤더 정보 뿐 만 아니라 내용 정보까지도 보안을 적용할 수 있다. 특히, 본 논문에서 제안한 보호기법은 Snort 이외의 다른 룰 기반의 정보보호 시스템에도 적용할 수 있을 것이다.

3.4 키 관리

본 논문에서 제안한 기법에서는 대칭키 암호 시스템을 사용한다. 이러한 시스템에 있어서 키 관리는 아주 중요한 문제이다. 이를 위하여, 본 논문에서는 논문[12]에서 제안된 PCMCIA 암호 모듈을 사용한 키 생성 및 관리 기법을 이용한다.

가. 키 관리의 필요성

제안한 기법에서 사용한 Triple-DES는 암호화 및 복호화를 위한 하나의 키가 필요하다. 알고리즘 자체의 견고함과는 다른 측면에서 적절한 키의 생성 및 관리는 매우 중요한 문제이다. 또한, 키의 크기가 너무 작다면 사전 공격등과 같은 방법으로 원문을 쉽게 알아낼 수 있고, 반대로 키가 너무 클 경우 키의 저장이나 연산 시간의 증가와 같은 또 다른 문제가 생길 수 있으므로 적절한 키의 크기를 선택 하는 것도 중요하다. 본 논문에서는 56비트의 크기를 갖는 두개의 키를 사용한다.

나. PCMCIA 암호 모듈을 사용한 키 관리 기법

논문[12]에서는 PCMCIA 암호 모듈에 기반한 키 관리 기법을 제안하였다. PCMCIA 암호 모듈의 경우, 자체적인 암호와 복호 연산을 수행할 수 있으며 FIPS PUB 140-1 레벨 3을 만족한다. 키 생성 시에는 PCMCIA 암호 모듈 자체적으로 키를 생성하거나 외부에서 생성한 키를 저장할 수 있다. 이 모듈은 실제로 키가 저장될 암호토큰과 이에 접근하기 위한 PED(Pin Entry Device)로 구성되고, 사용자는 PIN KEY를 사용하며 PED를 통해서 암호토큰에 로그인 한 후 암호토큰을 사용하여 키 생성을 수행한다. 키 생성을 내부적으로 수행할 경우 키가 외부로 노출되는 일이 전혀 없으므로 키를 더욱 안전하게 생성하고 관리할 수 있다. 본 논문에서의 키 관리 기법은 논문[12]의 기법을 따른다.

그림 11은 PCMCIA 암호 모듈을 통한 키의 생성과 사용 과정을 보여준다. 룰의 암호화에 사용될

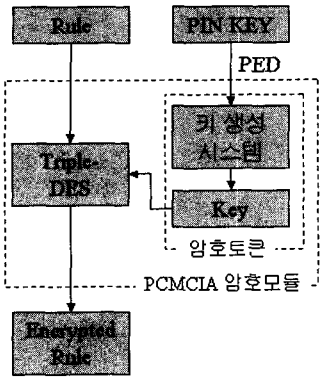


그림 11. PCMCIA를 이용한 키관리

키는 관리자가 입력한 PIN KEY에 의해 모듈 내에서 생성되고 저장된다. 키는 시스템 내부 버스를 통해 직접 Triple-DES 암호화 모듈로 전달된다. 생성된 키에 접근할 수 있는 유일한 경로는 PED이기 때문에 PIN KEY를 가진 관리자 이외의 그 누구도 키를 생성하지도 접근 할 수도 없다. 따라서 룰의 암호화에 사용되는 키의 유출에 대한 문제를 효율적으로 해결할 수 있다.

Ⅳ. 실험 및 분석

이 장에서는 본 논문에서 제안한 새로운 룰 보호 기법의 실험 결과를 보이고, 기존의 연구와 비교분석을 제시한다.

4.1 실험

그림 12는 실험을 위한 네트워크 구성도를 보여준다. 각 시스템의 속성은 다음과 같다.

- 침입 탐지 시스템 : AMD Athlon XP 2000+, Linux Kernel 2.4.9, libpcap, 100MB NIC
- 하위 네트워크 호스트 : Intel Pentium-III 800MHz, Windows 98
- 침입 시스템 : Intel Pentium-III 800MHz, Linux Kernel 2.4.9
- PCMCIA카드 : 침입 탐지 시스템 상에서 thread로 구현

실험의 편의상 침입 탐지 시스템은 게이트웨이와 내부 네트워크 사이에 위치시켰으며, 내부 네트워크

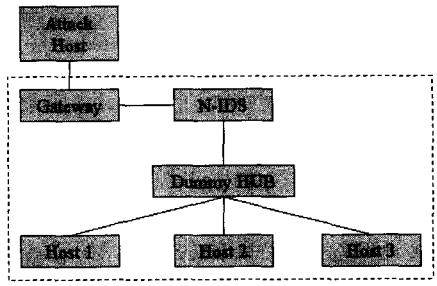


그림 12. 네트워크 구성도

로 들어오는 모든 패킷들을 수집 하였다. 효율적인 실험을 위하여 Snort에서 탐지할 수 있는 100가지의 공격을 수행하였고, 룰 기반의 시스템이므로 0%의 FAR(false accept rate)과 0%의 FRR(false reject rate)을 확인할 수 있었다.

그림 13은 서비스 거부 공격(DOS, denial of service)의 하나인 Jolt 공격의 예를 보여준다. 외부 네트워크에 위치한 공격 시스템으로부터 내부의 서브넷에 속한 임의의 호스트로의 공격을 시도하였다.

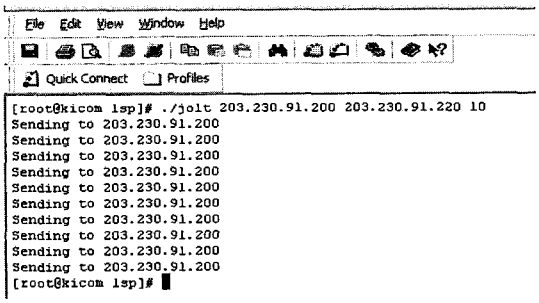


그림 13. Jolt 공격

그림 14는 서비스 거부 공격에 대한 룰의 예제를 보여주고, 이 룰에 대한 보호된 룰은 그림 15에서 보여준다. Snort에서 사용되는 룰은 그림 13에서

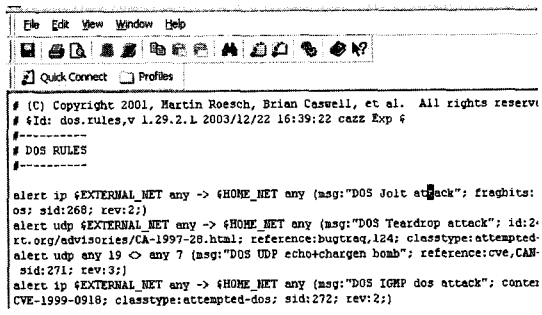


그림 14. Snort의 DOS 공격 룰

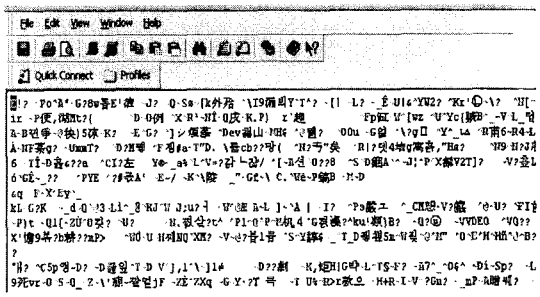


그림 15. 보호된 DOS 공격 룰

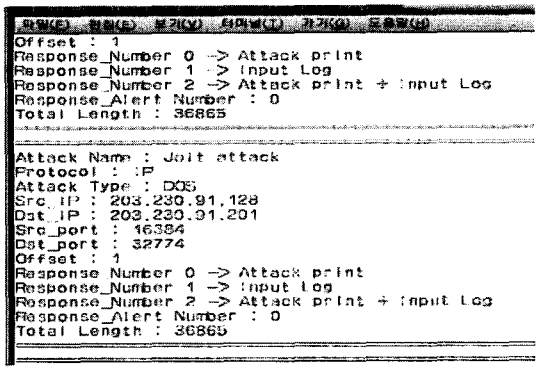


그림 16. 탐지 결과

보여주는 바와 같이 일반적인 텍스트로 저장되어 있기 때문에 그 내용이 유출될 수 있으나, 그림 15의 암호화 된 룰은 암호에 사용된 키를 모르면 정보의 내용을 확인할 수 있는 방법이 없다.

그림 16은 침입 탐지 결과를 보여준다. 전체적인 시스템의 처리과정은 3.3절에서 제시한 침입 탐지 과정을 따른다.

4.2 분석

본 논문에서는 기존 연구의 문제점을 해결하기 위한 새로운 기법을 제안하였다. 논문[5-6]과[10]에서는 단방향 해쉬 함수를 이용한 룰의 보호 기법을 제안하였다.

이들의 기법에서는 헤더정보에 관한 룰의 정보보호는 제시할 수 있었지만, 내용정보에 대한 보호에는 적용하기 어려운 문제가 있었다. 본 논문에서는 대칭키 암호화 시스템을 이용하여 기존의 룰 보호 기법의 문제점을 해결하였다. 표 1은 룰 보호 기법간의 특성 비교를 보여준다.

다음 표 2는 DOS(Denial of service)공격을 이용하여 Triple-DES를 이용한 본 논문에서 제안한 방식과 MD5를 기반으로 한 해쉬함수를 이용한 방법, 그리고 룰을 보호하지 않는 방법 간의 성능분석을 보여준다. 효율적인 분석을 위해서 룰을 보호하지 않는 방법의 수행시간을 0으로 설정하였다.

표 2는 Snort의 DOS공격을 위한 룰이 10개 존재하는 시스템에서의 세 가지 기법간의 시간 차이를 보여준다. 해쉬함수를 이용한 경우보다 본 논문의 방법을 이용한 경우가 침입탐지시간 면에서 약 8배 정도 많이 소요됨을 확인할 수 있다. 이러한 이유는 MD5와 Triple-DES의 수행시간 차에서 발생한다.

그러나 더욱더 주된 원인은 본 논문의 방법에서는 내용정보에 대한 추가적인 보안을 위한 오버헤드가 발생하기 때문이다. 룰 보호 시간은 시스템 설정 과정에서 한번만 요구되므로 전체적인 시스템의 성능에서는 고려되지 않아도 된다. 따라서 시스템의 성능은

표 1. 룰 보호 기법간의 특성 비교

속성 / 기법	적용기법	헤더정보	내용정보	적용 대상
논문[10]	해쉬함수	보호가능	보호 불가능	유비쿼터스 환경의 보안 매커니즘
논문[5-6]	해쉬함수	보호가능	보호 불가능	Snort
제안한 기법	대칭키암호기법	보호가능	보호 가능	Snort 및 정책기반 보안 매커니즘

표 2. 룰 보호 기법간의 오버헤드분석(DOS공격의 예)

기법 / 항목	룰을 보호하지 않는 경우	해쉬함수를 이용한 경우	본 논문의 방법을 이용한 경우
룰보호 시간	0	0.222	3.332
침입탐지시간	0	0.666	4.998

수집된 패킷의 헤더 정보에 대한 암호화에 소요되는 시간과 패킷의 내용에 대한 룰을 복호화 하는데 소요되는 시간에 의존적이다.

이러한 성능 저하 문제는 파이프라인 구조를 사용한 고속 DES 암호 알고리즘[14]이나 VHDL을 이용한 PCMCIA와 같은 같은 별도의 하드웨어적인 칩[12][15]을 사용하거나 수행시간이 상대적으로 빠른 다른 암호알고리즘을 이용함으로써 해결될 수 있을 것이다.

V. 결 론

본 논문에서는 대칭키 암호화 시스템에 기반 한 정책기반의 보안 매커니즘의 룰을 효율적으로 보호하기 위한 새로운 기법을 제안하였다. 논문[5-6, 10]에서는 단방향 함수를 사용한 정책 보호 기법을 제안함으로써, 기존의 정책 기반 보안 시스템들의 정책에 대한 보안을 제공할 수 있었다. 하지만 단방향 함수의 특성상 패킷의 내용과 관련된 정보에 대해서는 기법을 적용하기 어려운 문제점이 있었다. 이러한 문제를 해결하기 위해 본 논문에서는 대칭키 암호화 알고리즘을 사용한 새로운 룰 보호기법을 제안하였다. 본 논문에서는 Snort 룰을 대상으로 Triple-DES를 사용하여 보호된 룰을 생성하고, 생성된 룰을 사용한 간단한 룰 기반의 침입 탐지 시스템을 구현하였다. 제안한 기법은 기존 기법의 문제점을 효율적으로 해결할 수 있었다.

향후 연구로는, 좀 더 다양한 룰 기반의 보안 시스템을 기반으로 제안한 기법을 적용해 볼 필요성이 있고 대칭키 암호화 시스템의 속도 저하를 위한 효율적인 해결책에 대한 연구가 필요하다.

참 고 문 헌

[1] Paul E.Proctor, Practical *Intrusion Detection Handbook*, Prentice Hall, 2001.

[2] 한국 정보보호 센터, "침입탐지 모델 분석 및 설계", 1996.

[3] 한국 정보보호 진흥원 기술문서, "네트워크 공격기법의 패러다임 변화와 대응방안", 2000.

[4] Snort, http://www.snort.org.

[5] 손재민, 김현성, 부기동, "침입 탐지 시스템을 위한 효율적인 룰 보호 기법", 한국 정보 과학회 추계 학술대회 2003, Vol. 30, No. 2(1)호, pp. 898-900, 2003.

[6] 손재민, 김현성, 부기동, "침입 탐지 시스템을 위한 효율적인 룰 보호기법", 한국 산업정보학회 논문지, Vol. 8, No. 4, pp. 8-16, 2003.

[7] William Stallings, *CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE*, Prentice Hall, 2003.

[8] 조완수, "정보 시스템 보안", 홍릉과학출판사, 2003.

[9] A. Juels and M. Wattenberg, "A fuzzy Commitment Scheme", *In Proceeding of the second ACM conference on computer and communication security CCS'99*. Singapore, pp. 28-36, 1999.

[10] H. Kvarnstrom, H. Hedbom, and E. Jonsson, Protecting Security Policies in Ubiquitous Environments Using One-Way Functions, *Lecture Notes in Computer Science 2802*, pp. 71-85, 2003.

[11] 나종근, 김동규, "차세대 고도 정보 통신망 환경에서의 안전체제 연구 -OSI 응용계층에서의 범용 키 관리 모델 설계와 구현을 중심으로-", 정보보호학회 논문지 Vol. 2, No. 2, pp. 40-51, 1992.

[12] 김영백, 이석래, 이재일, 고승철, "전자서명 키 관리 시스템에 대한 고찰", 정보보호학회지 Vol. 10, No. 4, pp. 1-9, 2000.

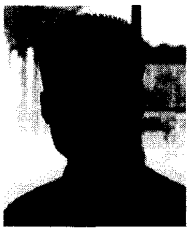
[13] 김대호, 박응기, 김영수, "멀티캐스트 적용을 위한 인터넷 키 관리 프로토콜 SKIP 분석", 정보보호 학회지 Vol. 9, No.4, pp. 25-40, 1999.

[14] 박태규, 황대준, "DES의 고속 암호화를 위한 파이프라인 구조", 정보보호학회 논문지 Vol. 3, No. 2, pp. 41-52, 1993.

[15] 한승조, "VHDL을 이용한 고속 DES 암호칩 설계 및 구현", 정보보호학회 논문지 Vol. 8, No. 3, pp. 79-94, 1998.

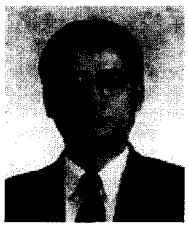
[16] 승기언, 이진우, 곽 진, 양형규, 원동호, "암호 시스템의 키 관리 기술", 정보보호학회지 Vol. 14, No. 4, pp. 45-53, 2004.

〈著者紹介〉



손 형 서 (Hyung-Seo Son) 학생회원
1999년 3월~현재 : 경일대학교 컴퓨터공학과
〈관심분야〉 정보보안, 네트워크 보안, IDS, PKI

김 현 성 (Hyun-Sung Kim) 정회원
1996년 2월 : 경일대학교 컴퓨터공학과 공학사
1998년 2월 : 경북대학교 컴퓨터공학과 공학석사
2002년 2월 : 경북대학교 컴퓨터공학과 공학박사
2002년 3월~현재 : 경일대학교 컴퓨터공학과 교수
〈관심분야〉 정보보안, 암호 알고리즘, 암호 프로세서 설계, IDS, PKI



부 기 동 (Gi-dong Bu) 정회원
1984년 : 경북대학교 전자공학과 전자계산기 전공
1988년 : 경북대학교 대학원 전산 공학전공 공학석사
1996년 : 경북대학교 대학원 전산공학전공 공학박사
1983년~1985년 : 포항종합제철 시스템개발실
2001년 9월~2002년 8월 : 게이오대학 교환교수
1998년~현재 : 경일대학교 컴퓨터공학과 교수
〈관심분야〉 데이터베이스, GIS, 시멘틱 웹