제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

적응적 규칙 추정에 의한 네트워크기반 침입탐지시스템 우회공격 방지 기법

최병철*, 서동일*, 손승원*
*한국전자통신연구원 네트워크 보안 연구부
e-mail: corea@etri.re.kr

Adaptive Rule Estimation (ARE) Algorithm against Eluding NIDS

Byeongl-Cheol Choi*, Dong-Il Seo*, Sung-Won Sohn*
*Dept of Network Security, ETRI

요 약

본 연구는 네트워크 기반 침입탐지 시스템(NIDS)의 우회공격 방지를 위한 적응적 규칙 추정 알고리즘(ARE: Adaptive Rule Estimation)을 제안한다. 네트워크 기반 침입탐지 시스템에서 가장 많이 사용하는 침입탐지 방법은 규칙 기반의 패턴 매칭 기법이며, 이 방법은 삽입과 삭제에 의한 우회 공격에 많은 취약성을 가지고 있다. 본 연구에서는 이러한 삽입과 삭제에 의한 우회 공격을 방지 하고자 하는 취지에서 제안된 알고리즘이다. 적응적 규칙 추정에 의한 침입 탐지 알고리즘은 두 개의 과정으로 구성되며, 전처리 부분에서는 최적의 규칙을 선택하고, 주처리 부분에서 적응적으로 규칙 패턴의 변형된 위치를 찾아서 비교 판단하는 과정으로 이루어져 있다. 제안된 적응적 규칙 추정 알고리즘은 기존의 규칙 기반 패턴 매칭에서 우회공격이 가능한 것들이 탐지되며, 미탐지 확률을 줄일 수 있다.

1. 서론

최근에 침입탐지시스템은 네트워크 보안의 강화를 위해서 방화벽과 함께 중요한 장치로 등장하였다. 또한, 방화벽과 침입탐지시스템 상호간의 연동으로 침입자의 연결 상태를 차단하는 방법도 개발되었다. 하지만, 방화벽 뿐만아니라 침입탐지시스템도 공격자에 의한 우회공격에 대해서는 아직 상당부분 방어할 수 없다. 특히 최근에 상용화되고 있는 많은 침입탐지시스템이 규칙 기반의 패턴매칭 방법을 사용하는 네트워크 기반 침입탐지시스템 (NIDS: Network -based IDS)이다. 이러한 네트워크 기반 침입탐지 시스템에 주로 사용하는 우회 방법이 패킷에 임의의 문자를 삽입 혹은 삭제하는 공격 및 서비스 거부 공격 (DoS)을 사용한다. 특히 삽입과 삭제에 의한 공격은 프로토콜의 특성을 이용하여 쉽게 우회공격을 할 수 있다.

본 논문에서는 이러한 네트워크 기반 침입탐지시스템의 취약점에 대해서 분석하고, 이러한 취약점을 극복할 수 있는 우회 공격 방지 기술에 대해서 적응적 규칙 추정 (ARE) 알고리즘을 제안하였다.

본 논문에서 우회공격을 위해 두 개의 문서에 대해서 분석을 하였다. "Insertion, Evasion and DoS: Eluding NIDS"와 Phrack57호에 게재된 "NIDS Evasion Method Named SeolMa"이다.

본 논문에서 제안한 알고리즘은 최근에 많이 문제가 되고있는 네트워크 기반 침입탐지시스템의 보안성을 강화하는 데에 초점이 맞추어져 있으며 IDS 개발자들에게 도움이 될 것이다.

2. 네트워크 기반 침입탐지시스템의 취약성

본 장에서는 침입탐지시스템에서 사용하는 일반적인 탐지 방법인 오용 탐지에 대해서 간략히 언급하고, 특히 네트워크 기반 침입탐지시스템에서 주로 사용하는 탐지 방법 중에 본 논문에서 기존의 방법으로 문제시되는 규칙 기반의 패턴 매칭 방법에 대해서 언급하였다. 또한, 이러한 네트워크 기반 침입탐지시스템의 규칙 기반의 패턴 매칭에 의한 탐지 방법에서 주로 사용되는 우회 공격 기술에 대해서 분석한다.

2.1 침입탐지시스템의 탐지 방법

침입탐지시스템은 탐지 방법에 따라 크게 오용탐지와 이상탐지로 구분한다. 현재 상용화 제품에서 가장 많이 사용되고 있는 방법이 오용탐지 방법이

다. 특히, 본 논문에서 초점을 두고 있는 방법은 규칙 기반의 패턴 매칭, 혹은 전문가 시스템이다.[2]

침입탐지시스템에서 주로 사용되고 있는 오용탐지 방법은 다음과 같다.

☞ 전문가 시스템 (규칙 기반 패턴 매칭)
☞ 상태 전이 분석
☞ 조건부 확률
☞ 패턴 매칭
☞ 키 스트로크 모니터링
☞ 모델 기반 침입 탐지 등

전문가 시스템은 가장 널리 사용되는 침입탐지 방법이며, 네트워크 기반 침입탐지시스템의 경우 네트워크 상에서 수집한 패킷을 분석하여 규칙 데이터베이스의 내용과 1대 1 패턴 매칭을 통하여 침입 여부를 판단하게 된다. 특히, 전문가 시스템의 경우에는 If-then 방법을 사용하여 공격 여부 판단 및 경고를 보내게 된다.

2.2 네트워크 기반 침입탐지시스템의 취약성

네트워크 기반의 침입탐지시스템의 취약성에 대해서는 많은 문서에서도 언급하고 있지만, 본 논문에서는 Insertion, Evasion and DoS : Eluding NIDS와 Phrack 57호의 Line- noise section에 기술된 NIDS Evasion Method Named SeolMa라는 문서를 바탕으로 삽입 및 삭제 공격에 그 초점을 맞추고 있다.[4,5]

NIDS의 문제점을 두 가지의 형태로 분류할 수 있으며, 각각은 insertion, evasion과 DoS이다. 처음은 복잡한 프로토콜의 처리상에서 일어나는 재구성된 패킷을 읽는데에 따른 불충분한 정보 수집에 기인한다. 두 번째는 IDS가 가지는 일반적인 취약점인 denial of service 공격에 취약한 것을 들고 있다.

삽입 공격 (Insertion Attacks)

End-system이 reject하는 패킷을 IDS는 받아들이는 경우이다. 일반적으로, Insertion 공격은 IDS가 end-system보다 패킷을 처리하는 것이 덜 엄격할 경우에 발생한다.

삭제 공격 (Evasion Attacks)

IDS가 reject하는 패킷을 end-system이 받아들이는 경우이다. 위에서 언급한 Insertion attacks의 반대의 경우로써, 패킷의 처리가 너무 엄격할 경우에 발생하며, IDS를 쉽게 exploit 시킬 수 있으며, IDS의 정확성을 파괴시킨다.

서비스 거부 공격 (DoS Attacks)

서비스 거부 공격은 NIDS에서 심각한 문제이다. 이 공격은 countermeasure capability가 실제 발생하지 않은 공격에 반응을 함(false positive)으로써 IDS가 마비 상태가 된다. 본 논문에서는 DoS는 고려하지 않았다.

NIDS 우회 공격 - SeolMa

이 공격방법은 TCP 프로토콜의 urgent mode의 취약점을 이용한 것이며, RFC1122에 기술된 점과 일반적으로 많이 사용하고 있는 BSD 계열에서의 약간의 차이가 있다.

☞ TCP 프로토콜에서 urgent mode의 이용으로 NIDS에서의 패턴 매칭 방법을 우 회할 수 있다.

☞ RFC1122에서 정의하는 urgent mode 와 전통적인 BSD시스템에서 파생된 것과 의 차이가 있다.

- RFC1122에서는 urgent pointer는 마지막 urgent data를 가르킨다.
- BSD 계열에서는 urgent pointer는 urgent data 다음에 따라오는 데이터를 가르킨다.

여기에서 중요한 사실은 대부분의 시스템이 BSD 계열에서 사용하는 urgent mode 방식을 사용한다.

3. 적응적 규칙 추정 (ARE) 알고리즘

본 논문에서 제안한 적응적 규칙 추정 (ARE: Adaptive Rule Estimation) 알고리즘은 네트워크 기반 침입탐지시스템의 규칙 기반의 패턴 매칭 방법의 취약점을 극복하기 위해서 제안된 것이다. ARE 알고리즘은 두 개의 과정으로 구성되어 있으며, 그것은 최적의 규칙을 찾아내는 전처리부분과 변조된 패킷의 복원을 하여 패킷 검사를 수행하여 침입 여부를 판단하는 주처리 부분으로 구성되어 있다.

ARE 알고리즘에서는 문자표를 사용하게 되는데, 이 문자표란 패킷의 ASCII 문자의 1대 1 문자 레벨을 표기한 것이다. 문자표는 본 논문에서 제안한 ARE 알고리즘을 수행하는데 새로이 도입한 것이다.

ARE의 전처리 부분에서는 LMSE (Least Mean Square Error)를 이용하여 수집된 패킷에 해당하는 최적의 규칙을 찾게 된다. ARE의 주처리 부분에서는 문자 비교와 변조된 패킷의 문자 위치 추정을 통하여 원래의 패킷을 복원하여 패턴 매칭을 수행하게 된다.

그림 1은 제안된 ARE 알고리즘의 순서도이다. ARE는 전처리 부분과 주처리 부분으로 구분되며, 전처리 부분에서는 LMSE를 사용하여 최적의 규칙

을 찾는 작업을 수행하고, 주처리 부분에서는 변조된 패킷을 복원하여 패턴 매칭을 수행하여 침입을 판단하게 된다.

다음은 본 논문에서 제안한 ARE 알고리즘의 상세 기술을 설명한 것이다.

ARE 알고리즘 전처리 부분

1. 유사한 규칙 찾기

패킷에 해당하는 규칙 찾기

처음에 수집된 패킷에 해당한 규칙 데이터 베이스를 선정하고 해당 규칙을 찾아야 한다. 예를 들면, 80 포트로의 공격 패킷이 수집되면, WEB 형태의 규칙 라이브러리를 설정하며, 우회 공격을 고려하여 네트워크에서 외부로 나가는 응답 패킷을 통해서 규칙을 판단하게 된다. 이러한 방법은 최근의 우회 공격을 고려하여 사용되는 것이다. 하지만 이것만으로 완전히 공격자의 침입 여부를 판단할 수는 없다. 따라서, 다음의 일련의 과정으로 검정을 거쳐야 한다.

패킷과 규칙에 대한 정의 및 정규화

$$\begin{cases} Packet : P_1, P_2, \cdots, P_n \ (n-bits) \\ Rule : R_1, R_2, \cdots, R_m \ (m-bits) \end{cases} \quad (1)$$

각 패킷과 규칙에 대한 비트 단위로 분석을 할 수 있도록 변경한다.

$$Type\ I\ (n > m)$$
$$\begin{cases} Packet : P_1, P_2, \cdots, P_n \ (n-bits) \\ Rule : R_1, R_2, \cdots, R_m, R_{m+1}, \cdots R_n \ (n-bits) \end{cases}$$
$$Type\ II\ (m > n)$$
$$\begin{cases} Packet : P_1, P_2, \cdots, P_n, P_{n+1}, \cdots R_m \ (m-bits) \\ Rule : R_1, R_2, \cdots, R_m \ (m-bits) \end{cases} \quad (2)$$

Type I은 삽입된 형태이며, Type II는 삭제된 형태의 경우이다. 이것은 문자표를 기준으로 비트 단위로 레벨이 결정되며, 추가된 임의의 비트는 0으로 설정된다.

MSE (Mean Square Error) 계산

$$MSE = \sum_{k=0}^{l-1} (R_k - P_k)^2 \quad (3)$$

우선 패킷과 규칙에 대한 MSE (Mean Square Error)를 구한다. 물론 MSE의 임계값을 설정하여야 한다[3]. 이 임계값은 시스템의 처리 속도 및 성능의 한 요소이다.

2. 최적의 규칙 설정

ARE 알고리즘에서 이번 단계는 LMSE (Least MSE), 즉 최소의 MSE의 값이 산출되는 규칙을 찾아내는 작업이다. 만약 LMSE 0인 경우는 기존의 규칙 기반의 패턴 매칭 방법이 그대로 적용되는 경우이다.

$$Example\ of\ WEB-CGI\ attack:$$
$$similar\ rule1: test-cgi,\ similar\ rule2: test$$
$$packet: tesYt-cXgi$$
$$MSE1 = MSE(rule1\ and\ packet)$$
$$MSE2 = MSE(rule2\ and\ packet)$$
$$if\ threshold\ of\ MSE = T_{MSE}\ and$$
$$T_{MSE} > MSE1\ and\ T_{MSE} > MSE2$$
$$\therefore LMSE \equiv MSE1$$

ARE 알고리즘 주처리 부분

1. 정규화 (NC 계산)

$$NC(Norm\ Count) = length(Rule) - length(Packet)$$
$$= \begin{cases} +n : n-bits\ inseting\ in\ packet \\ -n : n-bits\ inserting\ in\ rule \end{cases} \quad (4)$$
$$단,\ n \le \max\left|n \in (2^m \le [length(Rule)])\right|$$

여기에서, n은 규칙의 변형률 및 침입탐지시스템 처리 속도와 상관이 있다.

2. 변조된 패킷 위치 판단

문자 레벨화 (Character Bit Leveling)

패킷의 각 비트 단위의 문자들은 문자표에 따라서 값이 설정된다.

문자 비교 (Character Bit Projection)

패킷과 문자는 각각의 비트 단위로 레벨화된 값으로 비교가 되며, 이 단계에서 변조된 문자의 위치를 파악할 수 있다.

문자 이동 (Character Bit Shifting)

문자 비교를 통해서 변조된 패킷의 문자 위치를 파악한 후 문자의 이동이 1 비트씩 이루어진다. 이것은 NC를 고려하여 수행되어져야 하며, NC를 초과하는 경우는 시스템의 처리 능력 밖으로 판단한다. 만약 NC가 2이면, 문자 이동의 과정이 2번 수행한 후 패턴 매칭 및 침입 판단을 수행한다.

3. 패턴 매칭 및 침입 판단

이 부분에서는 삽입과 삭제의 경우에 따라 다르게 수행된다. 삽입 공격인 경우에는 변조된 패킷을 제외시킨 후 비교 판단을 하며, 삭제 공격인 경우는 문자를 복원하여 비교 판단을 하여 침입 여부를 결정하게 된다.

4. 실험 및 결과

우회 공격 실험에서는 Linux 시스템, Snort NIDS, Apache 웹 서버를 사용하고, 공격 툴로써 Phrack 57호에서 소개된 SeolMa를 사용하였다.

실제 테스트베드 상에서 SeolMa로 Snort가 설치된 네트워크를 우회 공격을 하였을 때, 예를 들어 GET /test-cgi /HTTP 1.0이라는 메시지를 TCP urgent mode를 사용하면 공격자는 시스템 정보를 유출할 수 있지만, Snort에서는 test-cgi 공격으로 판단을 할 수 없다.

그림 2는 본 논문에서 사용한 문자표이며, 표 1은 ARE 알고리즘의 2 비트 변조된 패킷으로 공격하였을 때의 시뮬레이션 결과이다. 이것을 그림으로 분석한 것이 그림 3이다. 표 2는 본 논문에서 제안한 ARE 알고리즘과 기존의 규칙 기반의 패턴 매칭 방법의 검출 확률의 가능성을 비교한 것이다. 만약 패킷의 길이가 8 비트이면 3 비트가 변조된 패킷도 복원 혹은 추정하여 침입 여부를 판단할 수 있다.

5. 결론

본 논문에서는 네트워크 기반 침입탐지시스템에 대한 삽입과 삭제에 의한 공격을 방지할 수 있는 적응적 규칙 추정 (ARE: Adaptive Rule Estimation) 알고리즘을 제안하였다.

제안한 ARE 알고리즘은 기존의 규칙 기반의 단순한 패턴 매칭 방법에서 기인하는 삽입과 삭제에 대한 공격을 네트워크 내부에서 외부로의 반응하는 응답 패킷을 통한 규칙의 추정 및 LMSE를 통한 최적의 규칙 찾기의 전처리 부분과 삽입과 삭제 현상에 의한 변조된 패킷의 추정 및 복원을 통하여 정확한 패턴 매칭 과정을 수행하는 주처리 부분으로 구성되어 있다. 실험 및 결과에서도 알 수 있듯이 8비트 패킷의 경우 3 비트가 변조된 경우에도 침입 여부를 판단할 수 있다.

앞으로의 연구 과제는 삽입 및 삭제에 의한 공격뿐만 아니라 서버스 거부 공격에 대한 고려도 하여야 할 것이다.

참고문헌

[1] W. R. Stevens, TCP/IP Illustrated Volume I: The Protocols, Addison -Wesley, 1994

[2] S. Northcutt, J. Novak, Network Intrusion Detection An Analysts Handbook, New Riders, 2001

[3] A. Leon-Garcia, Probability and Random Processes for Electrical Engineering, Addision - Wesley, 1994

[4] T. H. Ptacek, T. N. Newsham, Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection, Secure Networks, Inc., 1998

[5] Y. J. Ko, NIDS Evasion Method Named SeolMa , Line-noise Section, Phrack57, 2001

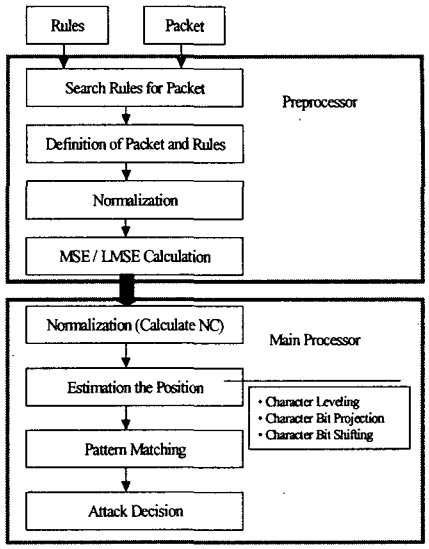

그림 1. ARE 알고리즘 순서도

0	1	2	3	4	5	6	7	8
etc	A	B	C	D	E	F	G	H
9	10	11	12	13	14	15	16	17
I	J	K	L	M	N	O	P	Q
18	19	20	21	22	23	24	25	26
R	S	T	U	V	W	X	Y	Z

level character

그림 2. 문자 레벨

표 1. 2 비트 변조된 패킷 탐지

Steps	Bit Count									
	1	2	3	4	5	6	7	8	9	10
Origin	t	e	s	Y	t	–	c	X	g	i
	20	19	5	25	20	0	3	24	7	9
1 step	t	e	s	t	–	c	g	i		
	20	19	5	20	0	3	7	9		
2 step	t	e	s	Y	t	–	c	g	i	
	20	19	5	25	20	0	3	7	9	
3 step	t	e	s	Y	t	–	c	X	g	i
	20	19	5	25	20	0	3	24	7	9

표 2. 성능 비교 (검출 확률)

Packet Length=8 Max(n)=3

Method	NC=0	NC=1	NC=2	NC=3
Pattern Matching	O	X	X	X
ARE	O	O	O	O