유전자 알고리즘을 이용한 침입탐지 규칙의 자동생성

정현진,° 한상준, 조성배
연세대학교 컴퓨터과학과
{cherishjhj, sjhan, sbcho}@sclab.yonsei.ac.kr

Automatic Generation of Intrusion Detection Rules using Genetic Algorithms

Hyun-Jin Jung,° Sang-Jun Han and Sung-Bae Cho
Dept. of Computer Science, Yonsei University

요 약

침입탐지 시스템 중 하나인 오용탐지 시스템은 축적된 침입패턴 정보를 이용하기 때문에 새로운 침입에 대하여 새로운 정의가 필요하다. 이러한 문제점을 극복하여 새로운 침입에 대하여 일일이 정의하지 않고 자동으로 새로운 규칙을 생성하도록 하는 것이 좀 더 바람직하다. 본 논문에서는 새로운 규칙을 찾기 위한 방법으로 생물의 진화과정을 모델링한 유전자 알고리즘(GA)을 이용하였다. GA는 계산에 의존한 방법에 비하여 전역적인 해를 구할 때 더 효율적이다. GA를 이용하여 규칙을 자동 생성하고 침입을 탐지할 수 있는 규칙을 찾아가는 방식을 제안하였다. 실험 결과에서는 GA를 이용하여 자동 생성된 규칙으로 40~ 60%의 탐지율로 침입을 탐지할 수 있다는 것을 확인하였다.

1. 서 론

오용탐지 시스템은 침입으로 알려져 있는 행위를 규칙으로 정의하고 수집된 검사사건이 미리 정의된 규칙과 일치하는 경우에 이를 침입으로 판정한다[1]. 이는 알려져 있는 침입패턴을 탐지하는데 적합하고 관리가 쉬우며 잘못 탐지할 가능성이 적다.

그러나 오용탐지 시스템은 새로운 침입에 대하여 취약하고 지속적인 침입탐지 규칙 수집이 필요하다. 본 논문에서는 오용탐지 시스템의 단점인 지속적인 침입탐지 규칙의 수집을 자동화하고자 한다. 이를 위해 새로운 침입에 대한 침입탐지 규칙을 찾는 알고리즘으로 유전자 알고리즘(GA)을 적용하였고 공개용 침입탐지 시스템으로 대표적인 Snort를 그 대상으로 하였다.

2. 관련연구

Mart Crosbie는 축적된 지식 기반인 침입탐지 시스템이 갖는 단점인 규칙을 만들고 시스템을 모니터링하는데 전문가가 필요한 것을 극복하고자 침입을 학습하여 새로운 침입을 탐지할 수 있는 시스템을 만들고자 하였다[2]. 학습하는 능력을 가진 시스템을 구현하기 위해 유전자 프로그래밍을 도입하였다. 이 시스템은 정상적인 상태를 학습하고 이와 비교하여 비정상적인 상태를 침입으로 판정하는 비정상탐지 시스템이다.

Dipankar Dasgupta 등은 기존 침입탐지 시스템이 패킷 레벨의 정보나 사용자의 행동 레벨로 침입여부를 판단하는데 반해 동시에 다중 레벨을 모니터링하여 알려지지 않은 침입도 탐지하여 일정한 반응을 하는 시스템을 구현하였다[3]. 이 시스템은 실시간 모니터링하여 분석하고 침입행동에 대하여 적절한 반응을 하도록 한다. GA를 적용하여 적합한 다중 레벨 모니터링 조합을 찾도록 했다.

위 연구에서는 침입정보를 학습하는 능력을 가진 시스템을 구현하기 위해 GA를 적용하거나 다중 레벨 모니터링을 하여 그 결과로 침입을 판정하는 것에 GA를 적용하였다. 이와 달리 본 논문은 실세계에 적용할 수 있도록 GA를 이용하여 침입 패턴에 맞는 규칙을 자동 생성하고자 한다.

3. GA를 이용한 Snort 규칙 생성

본 논문에서는 새로운 침입패턴을 찾는 알고리즘으로 GA를 선택하였고 침입탐지 시스템인 Snort의 규칙에 이를 적용하였다. Snort 규칙집단을 랜덤 생성한 후에 이를 테스트하여 결과를 평가하고 자동 생성된 Snort 규칙 집단에 GA 연산을 적용하여 새로운 Snort 규칙을 생성하는 방법을 반복 수행하였다. 이러한 과정을 도식화하면 다음 그림 1과 같다.

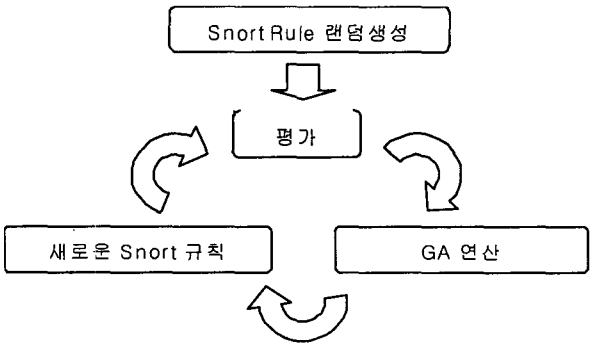

그림 1. 침입패턴 검색과정

3.1 Snort 규칙의 구조

본 논문에서 침입탐지 시스템으로 사용한 Snort는 오용탐지 시스템으로서 규칙 기반이자 네트워크 기반의 시스템이다[4]. 또한 공개용 침입 탐지 시스템으로서 상용 침입 탐지 시스템에 뒤떨어지지 않는다. Snort가 탐지할

규칙을 받게 되면 네트워크 패킷을 규칙으로 검사하여 패킷이 규칙과 일치할 경우에는 침입탐지로 간주하고 로그를 남긴다.

Snort 규칙의 구조는 Rule Header + Rule Option으로 되어 있다. 예를 들어 구조를 살펴보면 다음과 같다.

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (content: "USER root"; msg: "FTP root login")

Snort 규칙의 형태인데 alert은 로그를 남길 형태를 말하고 tcp는 프레임에서의 프로토콜을 말한다.

" $EXTERNAL_NET any -> $HOME_NET 21 " 은 외부에서 내부로 들어온 패킷중 21번 포트로 들어오는 패킷을 탐지한다는 부분으로 Rule Header에 해당한다.

" (content: "USER root"; msg: "FTP root login") "부분이 Rule Option에 해당한다. 즉 이 규칙은 외부에서 내부로 21번 포트로 들어온 패킷중 내용이 "USER root" 이면 탐지가 되고 "FTP root login"이라는 메세지를 보이며 로그를 남긴다.

3.2 유전자 알고리즘

유전자 알고리즘은 생물의 진화과정을 모델링한 알고리즘으로, 현 세대를 구성하는 모집단에서 적합도에 따라 염색체들을 선택하고, 유전 연산자를 적용하여 새로운 세대의 모집단을 생성해 나가는 방법이다[5].

유전 알고리즘이 다른 탐색이나 최적화 방법과 다른 점은 다음과 같다.

- 파라메터를 코딩한 것을 직접이용
- 일점이 아닌 다점 탐색 방법
- 탐색에 비용 정보를 이용
- 결정론적인 규칙이 없고 확률적 연산자를 사용

이와 같은 특징으로 인해 다른 탐색 또는 최적화 방법 중 하나인 계산에 의존한 방법에 비하여 전역적 해를 구할 가능성이 높으며 다른 여러 탐색방법에 비하여 효율적이다.

3.3 인코딩 및 적합도 평가방법

앞에서 살펴본 바와 같이 GA는 전역적 해를 구할 가능성이 높으며 효율적이다. 이러한 GA의 장점을 활용하여 침입에 대한 침입탐지 규칙을 찾고자 하였다. Snort는 규칙을 기반으로 하는 체인매칭 방식의 침입 탐지 시스템이다. 정의된 규칙의 모든 패턴과 일치하는 패킷을 침입으로 판정한다. 정의된 규칙과 하나의 패턴만 달라도 침입으로 판정하지 않는다. 따라서 대부분의 침입 패턴에 적용될 수 있는 옵션을 선정하여야 한다. 본 논문에서는 가장 일반적으로 사용되는 것들 중 Rule Header에서는 포트번호를, Rule Option에서는 contents를 선택하여 데이터 인코딩을 하였다.

데이터 인코딩에 사용된 포트 번호는 Well Known Port를 중심으로 침입이 빈번히 일어나는 포트를 선정하였는데, 선정된 포트중 일부와 그 포트의 사용용도는 표 1과 같다.

표 1. 포트번호와 용도

포트번호	용도	포트번호	용도
7	Echo	67	BOOTP
20	FTP, data	69	TFTP
21	FTP, conrol	79	Finger
22	SSH	80	HTTP
23	Telnet	109	POP2
25	SMTP	110	POP3
49	TACACS	111	Portmapper
53	DNS	113	IP

그림 2는 content에 문자가 2개인 경우 제안하는 개체의 데이터 인코딩을 보여준다.

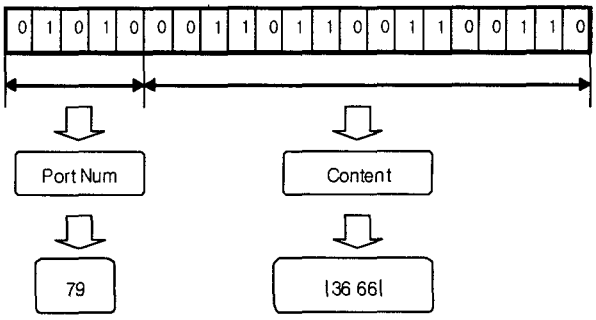

그림 2. 데이터 인코딩

그림 2는 집단중 하나의 개체로 데이터를 인코딩한 예를 보여준다. 이 개체는 테스트 과정에서 다음과 같은 Snort 규칙으로 변환된다.

alert tcp $EXTERNAL_NET any -> $HOME_NET 79 (msg: "eject!";content: "|36 66|";)

위 규칙에서 포트 번호가 79가 된 것은 이진수 01010 값이 10이고 이는 배열로 선언된 포트리스트 중 11번째 포트번호가 79이기 때문이다. content "36 66"은 16진수로 패킷 내용과 비교한다.

그림 2와 같이 자동 생성된 개체를 Snort를 이용하여 테스트하고 그 결과를 평가하기 위해 Snort 실행후 남겨진 로그파일을 읽어 탐지율과 오류율을 계산한다. 이를 평가하는 방법은 다음과 같다.

- 침입이 탐지 되었을 경우
  Fiteness = 탐지된 침입 수 × 10 - 오류율 × 10 + 1 (1)
- 잘못된 탐지만 있을 경우
  Fiteness = 1 - 오류율 (2)
- 탐지된 것이 아무것도 없는 경우
  Fiteness = 1 (3)

침입이 탐지된 경우에는 식(1)과 같이 탐지된 침입 수

에 가중치를 두어 좀더 많은 침입을 탐지한 경우 높은 평가값을 갖도록 하였고 탐지된 침입이 없이 잘못된 탐지가 있는 경우 식(2)와 같이 낮은 평가값을 갖도록 하였다.

4. 실험 및 결과

4.1 실험 환경

본 논문에서 실험 데이터로 사용한 패킷 데이터는 1998년 DARPA Intrusion Detection Evaluation program에 사용된 6주차 목요일 데이터이다[6]. 이 데이터에서 테스트하는 공격이 누출된 호스트로 들어오는 패킷만을 추출하여 데이터 셋을 만들어 사용하였다. 추출된 데이터 셋은 침입연결수가 5개이고 정상연결수가 295,422개이다. 침입연결수 중 2개는 비은닉 침입이고 3개는 은닉 침입이다. Simple GA를 사용하였고 GA설정은 다음과 같다.

- Populations : 64 ~ 512
- Iteration : 1000
- Crossover Rate : 0.70
- Mutation Rate : 0.05
- Roulette Wheel Selection

4.2 실험 결과

실험을 통해 생성된 규칙과 결과는 다음과 같다.

- alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg: "eject!";content: "|50 3b|";)
- alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg: "eject!";content: "|40 3a|";)
- alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg: "eject!";content: "|38 56 51|";)
- alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg: "eject!";content: "|3c 46 51|";)

실험 결과에 따른 탐지율은 정상탐지수 / 전체공격수 × 100 으로 나타내었고 표 2와 같다.

표 2. 자동 생성된 규칙에 따른 결과

port	content	탐지율	false alarm 수	침입특징
23	|50 3b|	40%	0	비은닉 침입
23	|40 3a|	60%	3	은닉 침입
23	|38 56 51|	60%	2	은닉 침입
23	|3c 46 51|	60%	0	은닉 침입

실험은 eject침입을 대상으로 하여 이루어졌다. eject침입은 telnet으로 접속하여 오버플로우를 일으키는 프로그램을 실행함으로써 root 권한을 얻고자 하는 것이다. 표 2에서 보는 바와 같이 침입특징으로 비은닉 침입과 은닉 침입이 있다. 비은닉 침입은 프로그램 소스를 그대로 입력하거나 전송한 경우이고 은닉 침입은 프로그램 소스를 암호화하여 전송한 경우이다.

표 3. 자동 생성된 규칙과 그 의미

생성된 규칙(content)	아스키 코드의 문자
|50 3b|	P ;
|40 3a|	@ 7
|38 56 51|	8 V Q
|3c 46 51|	< F Q

생성된 규칙의 content의 의미는 표 3과 같다. 패킷에서 content를 확인한 결과 비은닉 침입의 경우 변수명의 일부였고 은닉 침입의 경우 암호화된 프로그램 소스의 일부였다.

5. 결 론

본 논문에서는 오용탐지 시스템이 지속적으로 새로운 침입탐지 규칙을 수집해야 한다는 단점을 극복고자 GA를 이용하여 침입탐지 규칙을 자동 생성하는 방법을 제안하였다. 하지만 아직 높은 오류율, 좁은 적용 범위, 테스트 소요시간 등의 문제점이 남아 있다. 또한 생성된 규칙을 평가하기 위해서 패킷이 침입이라는 것을 알고 있어야 한다. 이는 비정상행위 탐지 시스템과 연계하여 자동으로 침입을 판정하고 침입탐지 규칙을 생성하는 진화하는 침입탐지 시스템을 구축하여 해결할 수 있을 것이다. 앞으로 침입탐지 규칙의 적용 옵션을 늘려 오류율을 줄이고 탐지 범위를 넓히는 방향을 모색하는 한편 침입패턴을 종분화하여 탐색을 좀 더 효율적으로 찾도록 개선해야 할 것이다.

감사의 글

본 연구는 대학 IT 연구센터 육성/지원사업의 연구결과로 수행되었음.

참고 문헌

[1] S. E. Smaha, Tools for Misuse Detection, *In proceedings of ISSA'93*, Crystal City, VA, 1993.

[2] M. Crosbie and G. Spafford, "Applying genetic programming to intrusion detection," *In Proceedings of AAAI Symposium on Genetic Programming*, pp. 1-8, November 1995.

[3] D. Dasgupta and F. A. Gonzalez, "An intelligent decision support system for intrusion detection and response," *In Proceedings of International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security*, pp. 1-14, May 2001.

[4] Martin Roesch. et al, Snort Users Manual Snort Release: 2.0.0, 8th, 2003

[5] D. E. Goldberg, *Genegic Algorithms in Search, Optimization, and Machine Learnig*, Addison-Wesley publishing company, *Inc.* 1989.

[6] MIT Lincoln Labs, 1999 DARPA intrusion detection evaluation. In *http:://www.ll.mit.edu/IST/ideval/index.html*