대화형 유전자 알고리즘을 이용한 다양한 침입패턴 생성

구자민°, 조성배
연세대학교
icicle@sclab.yonsei.ac.kr°, sbcho@cs.yonsei.ac.kr

Generation of Various Intrusion Patterns using Interactive Genetic Algorithm

Ja-Min Koo°, Sung-Bae Cho
Dept. of Computer Science, Yonsei University

요 약

전산화로 인해 컴퓨터 시스템에 대한 불법적 침입이 증가하고, 해킹으로 인한 피해가 급증하게 됨으로 인해 침입탐지시스템에 대한 많은 연구와 개발이 이루어지고 있다. 지금까지의 침입탐지시스템 성능은 기존에 알려진 침입패턴을 가지고 평가를 해왔기 때문에 새로운 침입에 대한 탐지능력을 측정할 수 없는 취약성이 있다. 본 논문에서는 침입탐지시스템의 성능 평가를 위해서 새롭고 다양한 침입패턴을 만들기 위한 방법으로 대화형 유전자 알고리즘을 이용하여 침입패턴을 생성하는 방법을 제안한다. 생성된 침입패턴은 사람에 의해 적합도가 평가되고 높은 점수를 가진 침입패턴의 가용성을 검증하기 위해 오용탐지방법을 사용하는 침입탐지시스템인 LinSTAT에 적용하여 보았다. 실험결과 생성된 침입패턴의 약 43%정도는 탐지되는데, 탐지되지 않은 57%는 새로운 침입패턴이라 할 수 있다.

1. 서론

인터넷의 범용화로 정보의 교류 속도가 매우 빨라져 업무의 효율성이 증대됨과 동시에, 이로 인한 보안사고의 수가 급증하고 있어 보안문제의 심각성이 날로 높아지고 있다. 전산 시스템의 보호를 위하여 다양한 방법을 이용한 침입탐지시스템이 연구 및 개발되고 있으며, 이들의 성능을 높이기 위한 연구도 국내외적으로 활발히 진행되고 있다[1,2]. 하지만, 침입탐지시스템의 성능을 평가하기 위해 사용하는 침입패턴들은 주로 기존에 알려진 것들이기 때문에, 변형되거나 혹은 새로운 침입패턴이 시도될 경우의 탐지능력을 측정하기 어렵다. 따라서 침입탐지시스템의 성능을 공정하게 평가하기 위해서는 기존의 침입패턴 뿐만 아니라, 알려지지 않은 다양한 형태의 침입패턴이 필요하다.

본 논문에서는 침입탐지시스템의 성능평가를 위해 새롭고 다양한 침입패턴을 생성하기 위한 방법으로, U2R 공격의 패턴을 일반화 하고, 대화형 유전자 알고리즘을 이용하여 새로운 침입패턴을 생성한다. 그리고 생성된 침입패턴을 Santa Barbara에서 제공하는 침입탐지시스템인 LinSTAT[3]을 이용한 실험으로 그 가용성을 검증한다.

2. 대표적 침입탐지시스템 평가연구

침입탐지시스템 평가관련 연구의 대표적인 기관은 DARPA[4], MIT Lincoln Lab[2], UCDavis[5], ICSA[6] 등이 있다. DARPA는 초기에 적은 수의 시스템에 대해 단순한 트래픽을 포함한 은닉이 아닌 적은 수의 공격만으로 평가하였으나, 1998년 DARPA의 1차 평가에서는 10개의 시스템, 38개의 공격, 충분한 트래픽과 일부의 은닉화된 공격을 통한 평가가 이루어졌다. 2000년 DARPA의 침입탐지평가는 사용자의 편의성과 성능향상에 중점을 두었으나 다양한 공격데이터의 수용에서는 미흡한 면을 보였다.

MIT Lincoln Lab은 1998년부터 DARPA의 지원 아래 침입탐지 관련 연구를 위한 방향을 제시하고 기술에 대한 객관적 개선을 위해서 침입탐지평가를 주제로 과제를 수행하였다. 평가항목은 침입탐지율, 오판율, 자원 사용량 등이며, R2L, U2R, DOS, R2U, Probing 등 공격방법을 유형별로 분류하고 Solaris, Windows NT, Linux 등 운영체제의 특성을 고려한 침입패턴을 생성하여 탐지시험 및 탐지우회 시험을 수행하였다.

UCDavis의 평가항목은 침입탐지율, 오판율, 자원사용량 등이며, 각 평가항목별로 태스트 케이스를 제시하여 침입탐지시스템을 실행하여 침입패턴 실행 및 탐지 결과를 측정하는 침입식별실험, 잡음 또는 CPU 부하 등 스트레스를 가하여 침입탐지 능력을 측정하는 스트레스 시험, 침입차단시스템의 CPU 및 메모리 사용량을 측정하는 자원사용량 시험을 수행한다.

ICSA는 미국의 Trust사에 소속되어 있는 네트워크 정보보호 제품 성능 및 취약성 시험을 중심으로 평가하는 사설 시험기관으로, 작성된 침입탐지 시험기준을 근거로 평가를 한다. 평가는 상용 침입탐지시스템을 설치하여 일정 수준의 환경을 구성한 후 침입패턴을 실행하여 침입탐지율, 오판율, 침입탐지 로그 수준을 기준으로 시행하며 침입 우회 가능성 및 취약성 시험은 별도로 수행한다.

3. 제안하는 방법

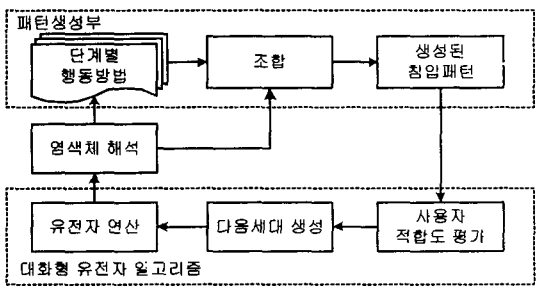

그림 1. 제안하는 방법의 전체 구조도

본 논문에서 제안하는 방법의 구조는 크게 패턴생성부분과 대화형 유전자 알고리즘(Interactive Genetic Algorithm: IGA) 부분으로 이루어져 있으며, 전체 구조는 그림 1과 같다. 패턴 생성부분에서는 각 단계별로 행할 수 있는 행위들을 조합하여 침입패턴을 생성한다. 각 단계별 행위들은 EXPECT 스크립트 언어[6]를 사용한다. EXPECT는 Tck/tk 스크립트의 확장패키지로 프로그램과 자동적으로 반응하기 위한 프로그래밍 가능한 터미널 인터페이스를 제공한다.

전 단계에서 EXPECT를 이용하여 생성된 패턴은 새로운 패턴을 생성하기 위해 대화형 유전자 알고리즘 부분으로 전달된다. 대화형 유전자 알고리즘은 적합도 함수가 명시되지 않는 문제에 사용되는 최적화 방법이다. 이것은 일반적인 유전자 알고리즘에서 적합도 함수 부분만을 사용자 평가로 대체한 것이며, 사용자가 각 개체에 대한 적합도를 직접 평가함으로써 사용자의 생각과 주관을 진화 과정에 적용시킬 수 있다. 각 단계별 행위들간에는 의존성이 있으므로 일반적인 진화 알고리즘을 사용할 경우 실행되지 않는 침입패턴이 높은 적합도를 가질 경우가 발생할 수 있다. 따라서 본 논문에서는 대화형 유전자 알고리즘를 사용하여 의존성의 강도에 따라 적합도 값을 부여한다. 각 유전자 알고리즘 프로그램은 이 값을 반영하여 다음 세대의 집단을 생성하며, 여기에 교차 및 돌연변이 연산을 적용시킨다. 이렇게 만들어진 집단을 해석하여 해당하는 침입패턴을 화면에 표시하고 앞서의 과정을 반복함으로써 사용자는 의존성이 약하고 안정적인 새로운 침입패턴을 얻을 수 있다.

침입 패턴을 생성하기 위해서는 침입이 일어나는 과정을 알아야 하는데, MIT Lincoln Lab에서는 U2R, R2L 등의 침입이 일어나는 있는 과정을 단계별로 나누고, 각 단계별로 행해지는 행동들을 분류하였다[2]. 본 논문에서는 MIT에서 분류한 과정에서 각 행해질 수 있는 행위들을 정리하여 침입패턴을 생성하는데 사용하였으며, 그림 2은 MIT에서 제안한 U2R 침입의 한 종류인 ffbconfig 침입이 일어나는 과정을 보여준다.

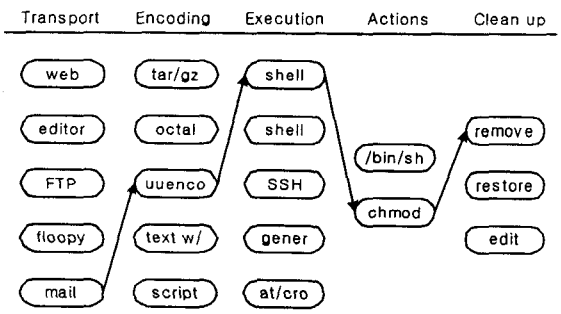

그림 2. ffbconfig 침입과정의 예

uuencoding 방식을 이용해 tar 압축된 파일이 Mail을 이용해 침입하고자 하는 시스템에 전달되며, 쉘 명령을 이용해 전달된 침입코드를 실행시킨다. 침입코드가 실행되면 루트권한을 획득하여 사용자는 원하는 파일의 권한을 변경하여 정보를 얻을 수 있다. 침입의 목적을 달성하면, 전달된 침입코드 및 tar 파일을 삭제하고 빠져나가 흔적을 지운다.

그림 2와 같이 5단계를 거칠 경우 각 단계별 한 부분씩 조합을 하면, 750(5×5×5×2×3)가지 경우의 패턴이 만들어진다. 침입탐지시스템의 성능을 평가하기 위해서는 다양한 종류의 패턴이 필요하다. 따라서 본 논문에서는 새롭고 다양한 종류의 침입패턴을 생성하기 위해서 기존 침입패턴의 행위를 분석하여 각 단계별 행위들을 추가하였으며, MIT에서는 하나로 합합친 암호화 및 복호화 단계를 본 논문에서는 따로 분리 하였으다. 그림 3는 전체 유전자형의 구성을 보여주며 표 1은 각 단계별 행할 수 있는 행위들의 추가된 내용의 일부이다.

암호화	전송단계	복호화	실행	액션	완료

그림 3. 유전자 인코딩

표 1. 단계별 행동 및 인코딩

	내용	인코딩	내용	인코딩
암호화	심볼릭링크사용	001	문자열 변환	010
전송단계	ehco 명령으로 파일생성	101	익명의 FTP 서버로부터 전송	110
복호화	vi 에디터이용수정	000	문자열 변환	001
실행	히스토리파일수정	100	make파일에 추가	000
액션	파일삭제	000	파일전송	001
완료	로그파일수정	011	히스토리파일삭제	100

그림 3과 같이 인코딩 할 경우 유전자형이 가질 수 있는 조합의 수, 즉 탐색 공간의 크기는 8×10×8×8×8×8=327,680이다. 초기 개체들 중에서 침입이 일어날 가능성이 높은 패턴이 선택되고, 선택된 패턴은 다음 세대에 나타날 확률을 높임으로써 좀 더 완성도가 높은 침입패턴을 생성할 수 있다.

4. 실험 및 결과

4.1 실험 환경

대화형 유전자 알고리즘의 교차 연산율과 돌연변이 연산율은 각각 0.6과 0.05로 설정하였다. 적합도 평가를 위해 3명의 사용자들이 직접 의존성 관계 및 침입 가능성을 비교하며 적합도 점수를 0부터 100까지 10점 단위씩 10단계로 나누어 20번에 걸쳐 평가하였다. 또한 제안한 방법으로 생성된 침입패턴의 유용성을 Santa Barbara에서 제공하는 STAT 패키지에 속한 침입탐지 시스템 중 LinSTAT을 설치하여 실험하였다.

4.2 실험 결과

대화형 유전자 알고리즘은 일반적인 유전자 알고리즘과는 달리 주관적 평가에 기반하여 작동하기 때문에 정량적인 분석으로 그 수렴성을 보이는 것은 매우 어렵지만, 가장 간단한 방법으로는 사용자들이 평가한 적합도의 변화를 통해 수렴성을 보이는 방법을 사용한다. 각각의 탐색은 10세대까지로 제한을 두었는데, 이는 대화형 유전자 알고리즘의 특성으로 인한 한계로, 적합도 평가가 사용자에 의해 직접 이루어지기 때문에 집단의 크기와 세대 수를 비교적 작게 제한할 수밖에 없다는 점에 기인한다.

그림 4에서 보는 바와 같이 세대수가 10번으로 제한되어있음에도 불구하고 2, 3세대까지는 적합도가 떨어지다가 4세대 이후로 평균적합도와 최고적합도가 세대를 거듭하면서 꾸준히 오르고 있음을 보인다. 이는, 초반에는 각각 행위에 따른 의존성 문제에 부딪혀 생성될 수 없는 패턴을 만들기 때문이다. 하지만 이후 높은 의존성 문제에 부딪히는 행위들의 결합이 발생하지 않으므로 적합도가 꾸준히 오르는 것이다.

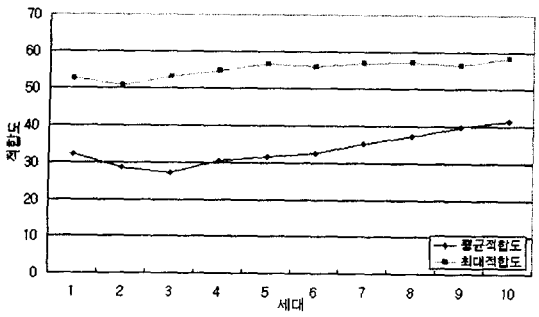

그림 4. 침입패턴 생성의 적합도 변화

```
①
send "lynx zeze.onelove64.com/temp6.htmWr"
expect "Download"
send "dWr"
send "W010W010W010.cWr"
sleep 1
send "y"
send "qy"

②
send "vi temp.cWr"
send ":1,$ s/(064)/a/gWn"
send ":1,$ s/(017)/b/gWn"
send ":1,$ s/(003)/c/gWn"
send ":1,$ s/(041)/d/gWn"
send ":1,$ s/(081)/e/gWn"
send ":1,$ s/(027)/f/gWn"
send ":1,$ s/(055)/g/gWn"
send ":1,$ s/(060)/h/gWn"
send ":1,$ s/(177)/i/gWn"
send ":1,$ s/(361)/j/gWn"
send ":1,$ s/(011)/k/gWn"
send ":1,$ s/(452)/l/gWn"
send ":1,$ s/(913)/m/gWn"
send ":1,$ s/(784)/n/gWn"
send ":wqWn"
sleep 1

③
send "vi MakefileWr"
send ":1,$ s/hello.c/temp.c/gWr"
send ":wqWn"
sleep 1
send "makeWr"
send "./helloWr"

④
send "cp /etc/shadow /etc/shaddowWr"
send "chmod o+r /etc/shadowWr"

⑤
send "rm -rf /etc/shadowWn"
send "cp /etc/shaddow /etc/shadowWr"
```

그림 5. 생성된 침입패턴 코드의 예

그림 5는 제안하는 방법으로 생성된 평균 적합도 이상의 적합도값을 가지는 침입패턴의 EXPECT코드를 보여준다. 생성된 패턴은 ①~⑤단계로 구성되며, ①에서는 이미 인코딩 된 temp6.htm, 010.c 파일을 HTTP 프로토콜을 이용하여 침입하고자 하는 시스템에 전송하는 과정이다. ②는 암호화 된 부분을 복호화하는 과정이며, ③은 침입코드가 실행되도록 하는 부분이다. 여기서는 다른 임의의 코드를 컴파일 하는 make 파일에 침입코드가 컴파일 되어 실행되게끔 하는 명령을 추가하여 실행되도록 하는 과정이다. ④에서 첫줄 코드는 원본파일을 복사해 두는 부분이고, 둘째 줄은 원본 파일의 권한을 변경하는 과정이다. 모든 행위가 끝나면 흔적을 없애기 위해 다시 원래 상태로 파일의 권한을 변경해야 하는데, 미리 복사해 둔 파일의 속성대로 원본 파일의 권한을 변경하는 과정을 ⑤단계에서 수행한다. 잘못 조합되어 낮은 적합도를 가진 패턴들은 ④, ⑤ 단계에서의 의존성 문제와 결부된다. 예를 들면 ④단계에서 파일 권한과 관련된 행위를 하지 않았지만 ⑤단계에서 파일 권한을 조절하는 행위를 할 경우가 있다. 하지만, 그림 5와 같은 코드에서와 같이 평균 적합도 이상의 값을 가지는 패턴들은 이런 의존성 문제에 결부되지 않음을 알 수 있다.

그림 5와 같이 생성된 침입패턴의 가용성을 검증하기 위해 평균 적합도 보다 높은 패턴 700개를 무작위로 추출하여 LinSTAT을 이용하여 탐지유무를 실험하였으며 표 2에서 보는 바와 같이, 생성된 패턴의 약 42%가 탐지되었다는 것은 이들이 LinSTAT에서 정의한 탐지 규칙과 동일한 패턴이라는 것을 알 수 있다. 나머지 탐지되지 못한 58%의 패턴들은 LinSTAT에서 가지고 있지 않은 새로운 침입패턴이다.

표 2. 생성된 침입패턴의 탐지율

입력된 패턴의 수	탐지된 패턴의 수	탐지율(%)
700	297	42.428

5. 결론 및 향후 연구

전산화로 인해 보안에 대한 관심이 급증하고 있는 가운데 다양한 침입탐지시스템이 개발 및 연구되고 있다. 또한, 이러한 침입탐지시스템의 성능평가를 위한 연구도 활발히 진행되고 있다. 하지만 성능평가를 위해 알려진 침입을 사용하기 때문에 새롭고 다양한 침입패턴이 필요하다. 본 논문에서는 대화형 유전자 알고리즘을 이용하여 새롭고 다양한 패턴을 생성하였으며, 가용성을 검증하기 위해 LinSTAT을 이용하여 실험하였다.

실험 결과, 생성과정에서 초기에는 의존성 문제에 부딪혀 낮은 점수를 획득하였으나 세대가 지날수록 의존성 문제에 부딪히지 않았다. 생성된 패턴을 분석 한 결과 각 단계별로 의존성 문제에 부딪히지 않은 행위끼리 잘 조합되어 생성됨을 알 수 있었다. 평균 적합도 이상의 값을 가진 패턴들을 분석한 결과 의존성 문제에 부딪히지 않았다. 또한 생성된 패턴들을 LinSTAT을 이용하여 검증한 결과 약 생성된 패턴의 약 58% 침입으로 작동됨을 알 수 있었다.

향후, 지금까지 정의한 각 단계별 행동이 모든 침입유형을 대표한다고 하기 어려우므로 좀 더 구체적인 행위들을 조사 및 추가적인 실험이 필요하다. 또한 각 단계에서 반드시 한 행동만이 선택되지 않고 두개 이상의 행동이 선택되거나 혹은 한 단계를 건너뛸 가능성이 있으므로 이런 예외적인 경우의 침입패턴 역시 감안할 필요가 있다.

감사의 글

본 연구는 대학 IT 연구센터 육성/지원 사업의 연구 결과로 수행되었음

참고문헌

[1] J. Nicholas, et al, "A methodology for testing intrusion detection systems," *IEEE Trans. on Software Engineering*, vol. 22, no. 10, pp. 719-729, October 1996.

[2] M. John, "The 1998 Lincoln laboratory IDS evaluation a critique," RAID 2000, pp. 145-161, October 2000.

[3] G. Vigna, S. T. Eckmann and R. A. Kemmerer, 'The STAT tool suite," *In Proc. of IEEE Information Survivability Workshop*, Boston, MA, October 2000.

[4] DARPA *Intrusion Detection Evaluation*, http://www.ll.mit.edu/IST/ideval

[5] N. Puketza, M. Chung, R. A. Ollson and B. Mukherjee, "A software platform for testing intrusion detection systems," *IEEE Software*, vol. 14, no. 5, pp. 43-51, September 1997.

[6] R. Bace, "An introduction to intrusion detection & assessment", *Technical Report*, ICSA INC., 2001.

[7] Don Libes, *Exploring Expect*, O'REILLY, 1996.