포트 스캐닝 탐지와 로그 분석을 이용한 해킹 탐지 시스템의 설계

○
이 성 희* · 김 상 욱* · 김 성 조*
* 중앙대학교 컴퓨터공학과

A Design of Hacking Detection System using Port Scanning Detection and Analysis of System Log Files

*Sung Hee Lee, *Sang Wook Kim, *Sung Jo Kim
* Dept. of Computer Science & Engineering, Chung-Ang University

요 약

네트워크 호스트에 대한 해킹의 기본 단계로서 포트 스캐닝이 사용된다. 최근에는 TCP의 보안 취약성과 구현 상의 버그를 이용한 스텔스 스캐닝 기법이 소개되었다. 본 논문에서는 스텔스 스캐닝의 원리에 대해 살펴보고, 자동화된 침입 탐지 및 보고 시스템의 구현을 위한 포트 스캐닝 탐지 기법으로서 스캐닝 성향 계수를 제안한다. 그리고 에이전트 기반 해킹 탐지 시스템의 모델을 제시함으로써 포트 스캐닝 탐지 기능과 로그 분석 기능이 상호 유기적으로 사용 가능함을 보인다.

1. 서론

미 국방부가 정보 입수를 목적으로 개발한 인터넷이 일반인에게 보급되면서 정보 통신의 흐름이 완전히 바뀌었다. 과거 전화와 팩스 등으로 이루어진 정보 통신은 인터넷이라는 새로운 통신 혁명을 맞으면서 급격하게 변화하고 있다. 실제로 96년 말 4천만명이었던 인터넷 사용자가 현재는 약 1억명으로 급증하였고 2000년에는 5억명 이상으로 늘어날 것으로 전망되고 있다. 이는 인터넷이 정보 통신의 대명사로 군림할 날이 머지 않았음을 예고하여 주는 대목이다. 그러나 인터넷이 급성장함에 따라 인터넷을 통한 해킹의 위험성은 날로 커져가고 있으며 그 방법 또한 다양해지고 있다.

해킹을 위한 기본 단계로는 주로 포트 스캐닝이 사용되며 최근에는 TCP 프로토콜의 보안 취약성을 이용한 스텔스 스캐닝(Stealth scanning)방법이 소개되고 있다[1, 2]. 포트 스캐닝이란 공격 대상 호스트의 주요 네트워크 서비스 포트들에 대하여 어떤 서비스가 제공되고 있는지 알아내는 작업이다. 포트 스캐닝을 탐지하기 위한 방법으로 네트워크 모니터링 도구의 사용이 필요하며, 대표적인 네트워크 모니터링 도구로 tcpdump를 들 수 있다. 그리고 시스템 정보를 얻기 위한 외부 침입자의 다양한 시도는 시스템 로그 파일의 분석을 통하여 탐지할 수 있다. 따라서 본 논문에서는 포트 스캐닝에 대한 탐지 기법으로서 스캐닝 성향 계수를 소개하고, 효율적인 로그 검색 기법을 제시한 후, 스캐닝 탐지와 로그 검색을 이용한 해킹 탐지 시스템을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 TCP 프로토콜의 연결과 해제 과정에서의 보안 취약성을 이용한 스텔스 스캐닝과 '스캐닝 성향 계수'를 이용한 스캐닝 탐지 기법을 소개한다. 3장에서는 스캐닝 탐지 및 로그 검색 기능을 이용한 에이전트 기반 해킹 탐지 시스템에 대해 제안한다. 4장에서는 결론 및 향후 연구 과제에 대해 서술한다.

2. 스텔스 스캐닝과 탐지 기법

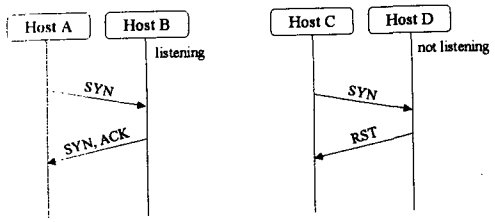
2.1 스텔스 스캐닝

TCP 패킷 헤더에는 패킷의 특성을 나타내는 컨트롤 플래그(CF) 필드가 존재하는데, 각 플래그가 나타내는 의미는 다음 <표 1>과 같다.

<표 1> TCP 헤더의 플래그 값의 의미

플래그	의 미
URG	Urgent 필드의 값이 의미 있음
ACK	전송 확인 번호가 유효함
PSH	현재 세그먼트의 데이터가 즉시 응용계층에 전달되어야 함
RST	연결의 재설정
SYN	연결 설정 요구이며 전송 확인이 있어야 함
FIN	단방향 연결 끊기

현재 알려진 스텔스 스캐닝 방법에는 세 가지가 있다. 첫 번째 방법은 Half-open 스캐닝이다. Half-open 스캐닝은 TCP 의 접속 확인 방법인 3자간 핸드쉐이킹(three-way handshaking)의 마지막 ACK를 보내지 않음으로써 정상적인 소켓 연결을 회피하는 방법이 사용된다. Half-open 스캐닝의 패킷 순서는 (그림 1)과 같다.



(a) 소켓이 열려 있을 때 (b) 소켓이 열리지 않았을 때

(그림 1) Half-open 스캐닝의 패킷 순서

다른 두 가지 방법은 TCP 프로토콜 구현의 버그를 이용한다. 두 번째 방법은 Send-FIN 스캐닝이다. 이 방법은 목표 호스트의 특정 포트에 FIN을 보내 돌아오는 패킷이 RST인지 검사하는 것이다. 만약 돌아오는 패킷에 RST가 세팅되어 있지 않다면 그 포트는 서비스를 제공하지 않는다는 판단이 가능하다. Send-FIN이 이용하는 버그는 대다수 시스템의 TCP 프로토콜 구현에서 발견되는 문제이다.

마지막 방법은 Send-ACK 스캐닝인데, 이 방법은 Linux와 FreeBSD TCP의 버그를 이용한다. Send-ACK 스캐닝은 ACK를 보내고 RST를 기다린 후, TCP헤더의 Window와 IP 헤더의 TTL을 검사한다. 패킷을 보낼 때의 TTL보다 받은 패킷의 TTL이 작다면 포트가 listen하고 있다고 판단한다. 이것은 Linux에서만 나타나는 버그이다. FreeBSD에는 되돌려주는 패킷 헤더의 Window를 0보다 크게 세팅하는 버그가 있다[2].

2.2 스텔스 스캐닝 탐지 기법

스텔스 스캐닝을 탐지하기 위해서 다음과 같은 방법을 사용한다. 앞에서 설명한 Half-open 스캐닝은 SYN 플래그가 세팅된 패킷을 전송하고, Send-FIN, Send-ACK 스캐닝의 경우에도 각각 FIN 플래그와 ACK 플래그가 세팅된 패킷을 전송하므로 TCP 헤더의 SYN, FIN, ACK를 모니터한다. 또한 일반적인 포트 스캐닝 프로그램에서도 TCP 연결을 위해서 3자간 핸드쉐이킹을 위한 SYN 패킷과 ACK 패킷을 전송하므로 같은 방법을 사용할 수 있다. 여기에 사용되는 네트워크 모니터링 도구로는 tcpdump를 사용한다. tcpdump를 이용함으로써 얻을 수 있는 가장 큰 장점은 소켓을 통해 들어오는 패킷에 대한 정보를 원하는 대로 필터링해서 출력해볼 수 있다는 것이며, 이것은 tcpdump 실행 시 파라미터의 지정을 통해 가능하다[3]. 포트 스캐닝 탐지를 위한 tcpdump의 파라미터는 다음과 같다.

```
tcpdump 'tcp[13] & 19 != 0'
```

이것은 TCP 헤더의 14번째 바이트를 19(13h)와 AND 연산을 시켜서, 즉 헤더의 SYN, FIN. ACK 중에 하나 만이라도 세팅이 되어 있으면 그 패킷 정보를 출력하는 것이다.

이 경우 동일 호스트에서 수십 초 이내에 다수의 포트에 대한 접근이 시도되었다면 스캐닝 행위로 간주할 수 있다. 이러한 행위에 대하여 침입 탐지 시스템은 관리자에게 보고할 기준으로서 본 논문에서는 호스트 h에 대한 스캐닝 성향 계수를 제시한다. 스캐닝 성향 계수는 시간별 포트 접속 수를 나타내는 것으로서 스캐닝 성향 계수가 높을수록 포트 스캐닝 공격을 당하고 있을 확률이 높아진다. 스캐닝 성향 계수 $D_c[h]$는 다음과 같다.

$$D_c[h] = \frac{p}{s_2 - s_1}$$

h: 상대편 *host*
p: 연결포트수
s_1: 첫 포트 접속 시간
s_2: 마지막 포트 접속 시간

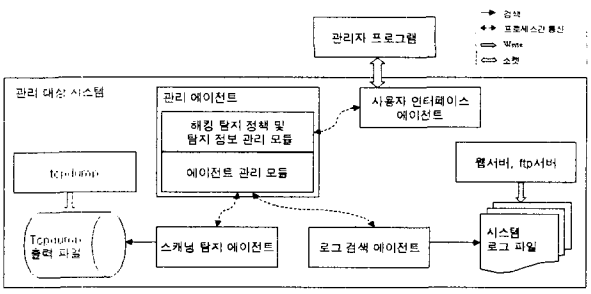
$D_c[h]$ 계산 후 스캐닝 탐지 시스템은 관리자가 정한 $D_c[h]$의 임계치와 비교하여 h의 스캐닝 여부를 자동적으로 판단한다. 그러나 만약 침입자가 수십 초에 걸쳐 스캔을 시도하면 (s_2-s_1)의 값이 커지므로 $D_c[h]$ 값은 낮아진다. 따라서 이런 경우에는 $D_c[h]$ 값을 배제하고, 관리자가 지정해준 포트 수 이상의 포트에 접속한 호스트를 침입자로 간주한다.

3. 해킹 탐지 시스템의 설계

3.1 해킹 탐지 시스템의 구성

해킹 탐지 시스템은 에이전트 기반 시스템으로 디자인된다. 해킹 탐지 시스템은 (그림 2)와 같이 해킹에 대응하기 위한 각각의 침입 탐지 기능과 역추적 기능들은 기능별로 개별적인 에이전트로 존재하며, 에이전트를 관리하기 위한 관리 에이전트와 사용자 인터페이스 에이전트가 별도로 존재한다. 각각의 에이전트는 개별적인 프로세스로 실행된다. 이와 같이 에이전트를 기반으로 해킹 탐지 시스템을 구성하는 이유는 다음과 같다.

- 해킹에 대응하기 위한 시스템의 확장이 유연성을 가진다. 새로운 에이전트가 추가는 에이전트 관리 모듈의 수정만으로 가능하며, 따라서 해킹 탐지 시스템 전체에 미치는 영향이 작아진다.
- 시스템의 불필요한 부하를 줄인다. 불필요한 에이전트는 실행시키지 않음으로써 시스템의 부하를 줄일 수 있다.
- 관리 에이전트와 사용자 인터페이스 에이전트를 제외한 에이전트의 실행이 중단되더라도 전체 시스템은 중단되지 않는다.



(그림 2) 해킹 탐지 시스템 모델

3.2 관리 에이전트의 구성

관리 에이전트는 사용자 인터페이스 에이전트로부터 해킹 탐지 정책에 관한 정보를 전달받아 스캐닝 탐지 에이전트와 로그 검색 에이전트의 동작을 제어하고, 스캐닝 탐지 에이전트와 로그 검색 에이전트로부터 해킹 탐지 결과를 전달받아 사용자 인터페이스 에이전트에 전달한다. 이를 위해 관리 에이전트는 해킹 방지 정책 및 탐지 정보 관리 모듈과 에이전트 관리 모듈로 구성된다.

해킹 탐지 정책은 에이전트의 상호 유기적인 탐지 활동을 위한 것으로, 예를 들어 스캐닝이 탐지된 경우에는 시스템 해킹의 가능성이 높으므로 강제로 로그 검색의 인터벌을 줄이고, 시스템 관리자에게 이를 통보하는 과정으로 설명될 수 있다.

3.2 사용자 인터페이스 에이전트

사용자 인터페이스 에이전트는 시스템 관리자로부터 해킹 탐지 정책에 대한 정보를 전달받아 관리 에이전트에 전달하고, 관리 에이전트로부터 해킹 탐지 결과를 전달받아 시스템 관리자에게 전달하는 역할을 수행한다.

사용자 인터페이스 에이전트를 관리 에이전트로부터 분리한 이유는 사용자 인터페이스를 다양하게 구성하기 위함이다. 즉 시스템 관리자는 사용자 인터페이스 에이전트의 구현에 따라 웹 기반 관리, 전용 관리 프로그램에 의한 해킹 탐지가 가능하다.

사용자 인터페이스 에이전트는 해킹 탐지 시스템의 시스템 외부의 네트워크와 직접 연결되는 유일한 통로이기 때문에 해킹의 공격 대상이 될 가능성이 존재한다. 따라서 사용자 인터페이스 에이전트의 통신 프로토콜 설계 및 관리자 인증에는 세심한 주의가 따라야 한다.

3.3 스캐닝 탐지 에이전트

2장에서 설명한 바와 같이 tcpdump의 출력을 이용하여 스캐닝 탐지 계수를 측정하여 스캐닝 탐지 계수가 높은 호스트가 발견되면 관리 에이전트에게 보고한다.

3.3 로그 검색 에이전트

로그 분석을 통해 역추적가능한 시스템 해킹 방법은 다음 <표 2>와 같다[4].

<표 2> 로그를 이용해 역추적 가능한 해킹 방법

로그 파일	파악 가능한 해킹의 종류
utmp, wtmp	허용되지 않는 호스트로부터의 접속
http 로그	phf.cgi 등으로 시스템 명령어 수행
ftp 로그	/etc/passwd, /etc/shadow 등 중요 파일 전송
messages	불법적인 재시동 및 서비스 중단

로그 검색 에이전트는 역추적 개념으로 <표 2>의 로그 파일 http 로그와 ftp 로그 파일을 검색하여 해커의 침입으로 간주되는 cgi의 실행 및 파일의 전송을 감시한다. 일반적으로 http 로그 파일과 ftp 로그 파일은 시간에 따라 지속적으로 사이즈가 커질 뿐 아니라 검색에 따른 시간이 소요되며, 텍스트 파일의 형태이기 때문에 파싱을 위한 부하도 가중되기 때문에 끊임없이 로그 파일을 감시하는 것은 시스템에 부하만 가중시킬 뿐이다. 따라서 해킹 탐지 정책에 따라 관리 에이전트가 일정한 시간 간격으로 호출한다.

5. 결론

네트워크에서 해킹의 시발점이라고 할 수 있는 포트 스캐닝을 탐지해낼 수 있다면 관리자는 자신의 시스템이 해킹의 목표물이 되고 있음을 쉽게 인지할 수 있고, 그에 따른 대비책을 세울 수 있다.

본 논문에서는 네트워크 포트 스캐닝에 대한 탐지 기법으로서 스캐닝 성향 계수를 제안하였고, 효율적인 시스템 로그 검색 방법을 제안하였다. 그리고 에이전트 기반의 해킹 탐지 시스템의 모델을 제시함으로써 탐지 정책에 따라 스캐닝 탐지와 로그 검색 기능의 유기적인 결합이 가능함을 보였다. 향후 사용자 인터페이스의 보안 문제와 탐지 정책에 대하여 연구가 이루어져야 할 것이다.

참고 문헌

[1] 임채호, 해킹과 보안의 新조류를 파악하라, 마이크로소프트웨어, 1999.5

[2] U. Maimon, TCP Port Stealth Scanning, http://munkora.cs.mu.oz.au/~fbugd/spoofing/portscan.html

[3] G. Maguire Jr., HTML Version of Tcpdump Manual Page, http://www.it.kth.se/edu/gru/Internet/tcpdump.html

[4] KISA, 정보시스템 침해사고 방지기술 개발에 관한 연구, 1999.1