라우터의 지원을 받는 네트워크 기반 침입탐지시스템의 구성

*김해식[0] *최경희 **정기현
*아주 대 학 교 정보 및 컴퓨터 공학부, **아주대학교 전자 전기 공학부
raja@cesys.ajou.ac.kr, {khchoi, khchung}@madang.ajou.ac.kr

Constructing Network-based IDS supported by Router

*Hae-Sik Kim[0] *Kyung-Hee Choi **Gi-Hyun Jung
*The Professional Graduate School for Information&Communication Technology, Ajou University,
** Division of Electrical & Electronics Engineering Ajou University

요 약

전통적인 네트워크기반 침입탐지시스템은 네트워크에 흐르는 모든 패킷을 수집하여 이를 가공,분석,보고하는 과정을 거친다. 하지만, 네트워크에서 과도한 트래픽의 발생이나 침입탐지시스템에 대한 의도적인 Dos(Denial of Service) 공격은 침입탐지시스템이 침입으로 간주될 수 있는 패킷을 처리하지 못하도록 함으로써 불법적인 접근을 얻어낼 수 있는 방법이 된다.

본 논문에서는 자체 개발한 내장형 리눅스 기반의 라우터에서 패킷의 필터링 작업을 수행함으로써 일차적으로 내부 네트워크와 네트워크 센서로의 트래픽을 줄이고, 이차적으로 정책기반 라우팅을 이용하여 네트워크 센서에게 직접 라우팅 하도록 함으로써 네트워크센서가 모든 트래픽을 수집하지 않고, 침입을 방지하고자 하는 정책에 기반하여 보내지는 패킷만을 수집,분석 토록 함으로써 네트워크 센서에 집중되는 부하를 최소화하는 시스템의 구성을 제안한다.

1. 서 론

인터넷 망의 확산과 다양한 서비스들이 제공되면서 인터넷을 통한 정보의 습득, 전달은 날로 증가되어 이제 보편적인 통신매체로 자리잡고 있다. 하지만, 인터넷의 사용자가 증가할수록 통신망을 통한 정보의 광범위한 노출은 중요 정보의 유출 가능성을 점점 크게 하고 있다. 초기에는 전문가에 의한 시스템의 침입이 대부분이었던 것에 반해, 현재는 전문 해킹도구의 빠른 유통으로 인하여 비 전문가라 할지라도 간단한 해킹도구를 사용하여 시스템에 막대한 피해를 줄 수 있는 것이 현 실정이다. 특히, 인터넷을 통한 전자상거래나 홈뱅킹 서비스가 해커의 침입을 받는 경우, 그 파급효과는 더 치명적이라 할 수 있다.

이러한 침입을 감지하는 시스템을 침입 탐지 시스템이라 한다.

간단한 침입탐지방법은 시스템의 로그파일을 검사하는 것이었다. 하지만, 해커는 시스템에 접근하여 관리자의 권한을 획득하고, 백도어를 설치한 후, 로그파일에서 해당 내용을 지움으로써 자신의 흔적을 지울 수 있다. 따라서 보다 효과적인 기술들에 대한 연구가 활발히 진행되고 있다. 침입탐지시스템에 대한 객관적인 실험은 아직 이루어지지 않아 그 효율성은 객관적으로 평가되지 못하지만 시스템을 보호하기 위한 하나의 방법으로서 최근의 활발한 연구분야가 되고 있다.

침입탐지 시스템 중 네트워크에 연결된 호스트 전체를 대상으로 네트워크에 흐르는 트래픽을 감시함으로써 침입을 감지하려는 시스템을 네트워크기반 침입탐지시스템이라 한다.

일반적으로 네트워크 기반 침입탐지 시스템은 크게 두 가지로 나누어 질 수 있다.

첫번째는 네트워크 패킷의 내용으로부터 필요한 정보를 축출하고 공격의 패턴을 분석하는 방법이다. NetRanger, Dragon등이 이에 속하는 상업적인 침입탐지시스템이다. 이런 방법은 보통 미리 알려진 공격패턴의 분석에 의한 signature 기반 침입탐지시스템이다.

잘 알려진 공격 패턴은 미리 데이터베이스에 저장되고, 네트워크 센서가 네트워크를 흐르는 모든 패킷을 수신하여 공격 패턴과의 패턴 매칭을 수행함으로써 이루어진다. 이러한 방법은 잘 알려진 공격을 효과적으로 감지할 수 있는 반면, 최신의 signature를 보유하지 않는 한 잘 알려지지 않은 공격에 대처할 수 없으며, 디스크 공간의 부족 등이 이 방법의 단점이다.

두번째는 네트워크의 접근 패턴을 통계적으로 분석하여 비정상적으로 동작하는 패턴인지를 검사하는 방법이다.

이런 방법의 가장 큰 장점은 새로운 공격 방식에 대하여 대처가능하다는 것이며, 단점으로는 장기간에 거쳐 접근의 패턴을 다양화 할 때 거짓탐지 또는 탐지실패가 더 쉽다는 것이다.

우리는 리눅스 기반 내장형 라우터를 개발하였고, 본 논문에서 제안된 침입탐지시스템은 이 라우터와의 연계 방법을 설명하며, 고속/대량의 데이터 망에서 패킷의 손실량을 최소화 하기 위한 침입탐지시스템의 구성을 제시하고자 한다.

2. 침입 유형과 제안된 네트워크의 구성

2.1 침입의 유형

2.1.1 DOS(Denial of Service) Attack

- Smurf Attack

  ICMP 프로토콜을 사용하여 공격하고자 하는 대상의 IP주소를 발신지 주소로 변조하여 로컬 네트워크 전체에 브로드캐스트 주소로 ICMP echo request 패킷을 보내게 된다. 로컬 네트워크 내의 모든 호스트는 echo reply 패킷을 공격대상의 호스트로 보내게 됨으로써 공격대상 호스트는 서비스 거부 상태에 빠지게 된다.

- Land Attack

  공격자는 발신지주소를 목적지주소와 동일하게 변조하여 SYN 패킷을 발송함으로써 공격 대상 시스템은 TCP Loopback이 발생하여 서비스 거부 상태에 빠지게 된다.

- SYN flooding

  일반적인 TCP/IP 네트워크 스택에서 TCP connection은 3-way Handshaking에 의해 이루어 진다.

  공격자는 발신지 주소를 존재하지 않는 호스트의 주소로 변조한 TCP SYN 패킷을 공격 대상 호스트로 보내게 된다. 공격대상 호스트는 SYN/ACK 패킷을 보내고, ACK 패킷을 기다리게 된다. 일반적인 경우, ACK패킷이 도달하거나 RST패킷이 도착하여 호스트는 다음 상태로 진행되지만, 최초 공격자는 존재하지 않는 호스트를 발신지 주소로 변조하여 사용하였으므로 어떠한 패킷도

공격 대상 호스트의 SYN/ACK 패킷에 응답하지 않는다. 이러한 경우, backlog queue에 TCP 연결상태가 저장되며 일정시간 SYN_ACK 패킷에 응답이 없다면 backlog queue에서 삭제된다. 하지만, 만약 이 시간 보다 더 빨리 SYN 패킷이 계속적으로 도착한다면, backlog queue는 overflow상태가 될 것이고, 더 이상의 서비스를 제공할 수 없게 된다.

- DDos(Distributed Denial of Service)

DDos 공격의 경우 공격은 여러 개의 호스트로부터 이루어진다. 공격자는 보안이 취약한 여러 호스트에 대하여 관리 권한을 취득한 후 DDos 소프트웨어를 업로드 한 후 실행시킴으로써 호스트는 공격자의 Dos 공격 명령을 기다리게 된다. 공격자가 DDos 공격 명령을 내림으로써 모든 호스트가 동시에 하나의 공격 대상 호스트에 대하여 Dos공격을 하게 된다. 현재 DDos tool은 급속히 확산되고 있을 뿐만 아니라 그 종류 또한 다양하다. 대표적인 예로 TFN, Trinoo, Stacheldraht, TFN2K 등이 있다

2.1.2 Buffer Overflow

버퍼 오버플로우 공격은 지정된 버퍼의 크기보다 더 많은 데이터를 입력해서 프로그램이 비정상적으로 동작하도록 만드는 방법으로써 공격자는 메모리의 Stack영역내의 적당한 곳에 공격자가 원하는 쉘코드를 집어넣은 다음, 리턴 어드레스를 그 삽입한 코드 부분이 있는 곳으로 바꾸어 줌으로써 해당 코드를 실행하도록 한다. 따라서, 이 후로는 이 쉘을 이용하며 모든 명령을 실행 시킬 수 있게 된다.

2.1.3 포트 스캔

포트 스캔은 실제 네트워크내의 호스트에 피해를 주기위한 방법은 아니다. 이는 공격 전에 공격 대상이 되는 네트워크의 정보를 얻기 위한 수단으로써 사용된다. 포트 스캔은 공격자가 공격 대상 호스트의 TCP/UDP 포트에 접속을 시도함으로써 해당 포트가 서비스를 위해 열려 있는지를 판단 할 수 있다.

2.1.4 네트워크 서비스의 취약점을 이용한 공격

대부분 Web, FTP, TELNET, SMTP 등의 서비스에서 프로토콜 고유의 취약점이나 프로그램의 버그 등을 이용한 다양한 형태의 공격 패턴들이 존재한다.

2.2 네트워크의 구성

대부분의 Signature 기반 침입탐지 시스템은 네트워크에 흐르는 모든 패킷을 스니핑(sniffing)하고, Signature set에 대하여 스트링 매칭을 수행함으로써 침입 여부를 판단한다. 일반적으로 스트링 매칭은 침입탐지시스템의 detection engine에 있어서 높은 CPU resource 를 요구하는 부분이다.

침입탐지시스템은 자신의 주소로 들어오는 패킷만을 처리 하는 것이 아니라 자신이 모니터링하는 네트워크 상의 모든 incoming/outgoing 트래픽을 처리하여야 하므로 새로운 패킷에 의해 버퍼에 채워지는 것 보다 더 빠른 시간 내에 패킷을 처리하지 못한다면 침입탐지시스템은 패킷을 버리게 된다. 하지만, 침입탐지 시스템은 일반적인 방화벽과는 달리 promiscuous mode에서 동작하므로 해당 패킷은 목적 호스트로의 접근이 허용된다.

따라서, 네트워크에서 과도한 트래픽의 발생이나 침입탐지시스템에 대한 의도적인 Dos(Denial of Service) 공격은 침입탐지시스템이 침입으로 간주될 수 있는 패킷을 탐지하지 못하도록 함으로써 불법적인 접근을 얻어낼 수 있는 방법이 된다.

이러한 고속/대량의 네트워크 망에서 패킷의 효율적인 처리를 위해 제안된 침입탐지시스템을 위한 네트워크는 아래[그림 1]과 같이 구성되어질 수 있다.

먼저 가장 큰 침입의 위협을 인터넷을 통한 외부 호스트로 가정하며, 가장 큰 침입의 주대상은 웹, 메일, 텔넷 등의 서비스를 제공하는 내부 네트워크내의 모든 서버로 가정하고자 한다.

제안된 시스템은 외부로부터의 불법 침입에 대한 대응책을 [표 1]와 같이 크게 2가지 단계로 분류한다..

	침입 유형
패킷 필터링	● Source routing 방지 ● Smurf Attack ● Land Attack
네트워크 센서	● SYN flooding ● 포트 스캔 ● DDos ● Buffer Overflow ● 네트워크 서비스 프로토콜의 취약성 또는 버그를 이용한 공격

표 1. 침입 유형별 대응 전략

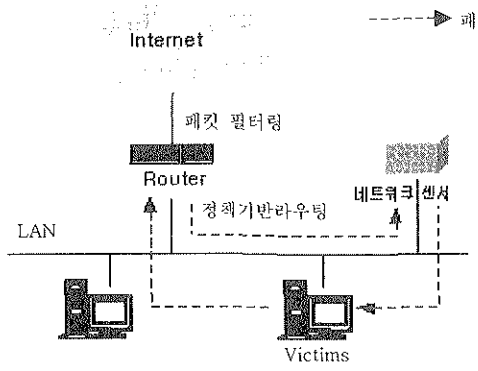

그림 1. 네트워크의 구성

첫번째 단계는 패킷 필터링 기능을 가지는 라우터의 역할이다.

두번째 단계는 네트워크상의 패킷을 수집하고 이를 분석하여 불법침입의 여부를 판단하는 네트워크 센서의 역할이다.

제안된 시스템에서는 Snort[2]를 네트워크 센서로 사용한다.

Snort 에서 네트워크의 트래픽을 수집하기 위해 기반으로 하는 libpcap 라이브러리는 TCPDUMP 에서 사용되는 패킷 필터 기술인 Berkely Packet Filter(BPF)를 제공함으로써 커널로 들어오는 트래픽의 양을 감소시킬 수 있다. 네트워크 센서로 들어오는 트래픽의 양은 일차적으로 라우터의 패킷 필터링을 통하여 걸러지며, 네트워크 센서에서 사용하는 BPF 는 이차적인 패킷 필터링을 통하여 실제 어플리케이션 레벨의 침입 탐지 엔진으로 가는 트래픽의 양을 줄일 수 있다.

라우터에서의 필터링을 통하여 내부 네트워크로의 유입이 허용된 패킷이라 할지라도 모두 침입탐지시스템에서 수집,분석 되는 것이 아니라, 보안정책에 따라 검사되어 질 패킷 만이 네트워크 센서로 직접 라우팅(Direct Routing)된다.

이러한 특징은 다음과 같은 중요한 장점을 가진다.

첫째, 라우터로부터 보내진 패킷만을 수집하고, 이를 분석하는 네트워크 센서는 고속의 네트워크 환경하에서 처리해야 할 패킷의 양을 줄임으로써 패킷 손실을 최소화 할 수 있다.

둘째, 보통의 이더넷 스위치 네트워크 환경에서 NIDS는 직접적으로 NIDS로 목적 IP 주소를 가지는 패킷 만을 수집 가능하다. 따라서, 내부 네트워크 내의 다른 호스트로 보내지는 패킷에 대해서는 실제 NIDS에 노출되지 않는다는 특성이 있다. 따라서, 이러한 환경에서 기존의 NIDS는 스위치의 미러링 포트를 사용하거나 리피터 허브를 사용하여 네트워크를 구성한다.

제안된 시스템에서는 라우터로부터 NIDS로 Direct routing이 이루어 지므로 모든 검사되어져야 할 패킷은 NIDS로 보내지도록 되어 있으므로 이러한 구성이 필요 없다.

3. 구 현

3.1 라우터의 지원

3.1.1 패킷 필터링

우리가 개발한 라우터는 기본적으로 리눅스 커널에게 제공되는 패킷 필터링 기능을 지원한다.

패킷 필터링 기능을 가진 스크리닝 라우터는 단순히 패킷을 어디로 전송할지를 결정할 뿐만 아니라 내부 네트워크로 유입되는 모든 트래픽을 검사하고 필터링 규칙을 적용함으로써 패킷을 전송 해야 할지 말지를 결정한다. 또한 이는 네트워크 상에서 진입구(checkpoint)를 감시하기 때문에 전체 네트워크에 하나의 통합된 보안 솔루션을 제공한다.

내부 네트워크로의 불법 침입에 목적을 둔 패킷은 스크리닝 라우터에 의해서 일차적으로 걸러진다.

패킷 필터링은 네트워크 계층에서 동작하기 때문에 사용자에게 투명성을 제공하며 저 비용의 뛰어난 성능을 제공한다.

패킷 필터링에서 가용정보는 {protocol, source address, source port, destination address, destination port}와 같은 TCP/IP헤더 정보로써 이러한 가용정보를 바탕으로 IP 계층에서 다음의 내용에

대하여 필터링 기능을 수행한다.

- 어드레스 필터링
- 프로토콜 필터링
- 패킷의 무결성 검사

아래[표 2]은 제안된 침입 탐지 방법에서 필터링 규칙의 예를 보여준다.

발신지 주소	발신지 포트	목적지 주소	목적지 포트	처리
*	*	192.168.0.0	23	permit
*	*	192.168.0.0	80	permit
192.168.0.0	*	192.168.0.0	*	drop
*	*	192.168.0.0	513	drop

표 2. 패킷 필터링 규칙의 예

논문에서 제안된 필터링 정책은 기본적으로 허용정책을 사용하고 불필요한 서비스나 일부 알려진 해킹 방법을 막기 위해 일부 주소 및 포트에 대하여 금지정책을 사용하였다.

제안된 침입탐지시스템에서 라우터에서의 패킷 필터링의 적용하는 데는 2가지 장점이 있다.

첫째, 불필요한 패킷의 유입으로 인한 내부 네트워크의 트래픽을 줄인다.

둘째, 내부 네트워크의 불법적 사용자체를 사전에 차단한다.

3.1.2 정책기반 라우팅

정책기반 라우팅은 라우팅을 위한 패킷 정보가 단지 목적지 주소에 의해서만 결정되는 것이 아니라 발신지 주소, 목적지 주소, 포트 번호, TOS에 의해서 결정되어 질 수 있으며, 각 라우팅 정책에 따라 복수개의 라우팅 테이블이 존재한다.

리눅스에서 정책기반 라우팅은 커널 버전 2.1부터 지원되기 시작하였으며, 기존의 목적지 주소 기반의 라우팅 테이블을 routing policy database(RPDB)로 대체하였다.

제안된 침입탐지 시스템에서 정책기반 라우팅은 방화벽을 통과한 패킷 중 탐지하고자 하는 패킷만을 침입탐지를 위한 네트워크 센서로 포워딩 하는 역할을 수행한다.

fwmark는 실제 라우팅 정책에 따라 IP주소 및 포트가 일치하는 패킷에 대하여 해당 라우팅 테이블을 사용하기 위해 라우터에 의해 검사되는 플래그이다.

3.2 네트워크 센서

제안된 시스템에서는 Snort[2]를 네트워크 센서로 사용한다.

Snort는 스니핑(Sniffing) 프로그램으로써 네트워크에 흐르는 모든 패킷을 수집한 뿐 만 아니라, 실시간 트래픽 분석을 통하여 정해진 규칙에 따라 로그 및 경고를 취할 수 있는 경량의 네트워크기반 침입 탐지 시스템이며, 소스는 웹상에 공개되어 있다.

Snort는 Signature 기반 탐지시스템으로써 프로토콜 분석, 패킷의 컨텐트 검색/매칭을 통하여 버퍼 오버플로우, 포트 스캔, CGI 공격, DOS 등 그 밖의 여러 가지 종류의 공격을 탐지하는 데 사용할 수 있다. 또한, 탐지 엔진이 사용하는 플러그인 아키텍처뿐만 아니라 사용자에게 친숙한 탐지규칙을 정의하는 인터페이스를 제공한다.

수집된 패킷의 패턴 매칭을 통하여 침입이 탐지되었을 경우, syslog, 및 SMB(Session Message Block)를 통한 실시간 경계 경보 능력을 갖추고 있다.

Snort는 아래와 같은 4 가지 엔진으로 구성되어 있다.

- 패킷 수집과 디코드 엔진
- 탐지규칙 파싱과 침입탐지검사 엔진
- 로깅 및 경고 엔진
- 플러그 인, 프리프로세서 핸들링 엔진

Snort는 네트워크의 트래픽을 수집하기 위해, libpcap 라이브러리를 기반으로 하며 네트워크 인터페이스가 promiscuous mode에서 동작하도록 한다.

libpcap 라이브러리는 Lawrence Berkeley 연구소에서 개발되어 tcpdump 에서도 사용되는 라이브러리로 사용자 수준에서 시스템에 상관없이 패킷 수집을 용이하도록 하는 라이브러리이다.

시스템과 운영체제에 따라서 패킷 수집을 가능케 하는 각기 다른 인터페이스를 제공하며, TCPDUMP 에서 사용되는 패킷 필터 기술인 BSD Packet Filter(BPF)를 제공함으로써 침입탐지 검사엔진으로 들어오는 들어오는 트래픽의 양을 감소시킬 수 있다.

아래 [그림 2]는 내부구조를 보여준다.

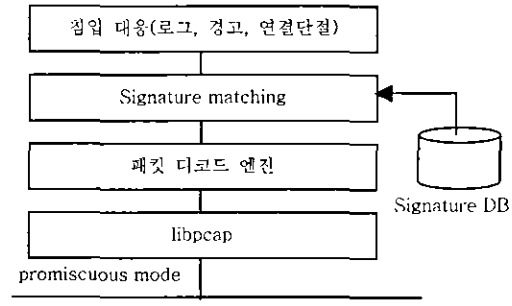

그림 2. 네트워크 센서의 구조

대부분의 침입탐지 시스템은 promiscuous mode 에서 침입탐지 시스템이 속한 단위 세그먼트 내에서 흐르는 모든 트래픽을 수집한다.

제안된 시스템에서는 일차적으로 필터링 기능을 수행하는 라우터로부터 다이렉트 라우팅 되어 보내지는 트래픽 만을 수집하도록 한다.

이것을 위해 아래와 같은 필터링 규칙이 적용된다.

ether src [ether_address]

Snort는 BPF의 필터링 규칙을 적용하기 전에 먼저 라우터로 ARP 메시지를 보냄으로써 라우터의 이더넷 주소를 알아내고 BPF 필터를 위와 같이 적용하도록 수정되었다.

4. 결 론

본 논문에서는 외부로부터의 침입으로부터 내부 네트워크를 보호하기 위한 시스템의 구성에 대하여 기술하였다.

일차적인 방법으로는 불필요한 내부 네트워크의 포트를 막기위해 라우터에서 패킷 필터링을 하였으며, 이차적으로 불법 침입 여부를 탐지코자 하는 패킷을 네트워크 센서로 보냄으로써 분석과정을 거쳐, 보고, 대응 토록 하였다.

네트워크 패킷은 계속 수집과정과 분석과정을 반복하므로 만약 이 과정이 늦어지면 연속해서 들어오는 패킷의 손실이 발생하게 된다.

따라서, 일차적으로 라우터로부터 필터링 된 패킷 만을 받는 네트워크 센서가 처리 해야 할 패킷의 양은 줄어들 수 있다. 또한, 스위칭 환경에서도 별도의 네트워크의 구성없이 적용이 가능하다.

제안된 시스템 구성을 위해서 네트워크 관리자는 내부 네트워크에 보호해야 할 서버의 정보를 알아야 하며, 패킷 필터의 정책과 네트워크 센서의 정책을 일치시켜야만 한다.

5. 참고 문헌

[1]. Bruce Corbridge, Robert Henig, Charles Slater, " Packet Filtering in an IP Router" , LISA, September, 1991

[2]. Martin Roesch, " Snort-Lightweight Intrusion Detection for Networks" , LISA' 99, November, 1999

[3]. Thomas H.Ptacek, Timothy N.Newsham, " Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection", Secure Networks, January 1998.

[4]. Nei KATO, Hiroaki NITOU, Kohei OHTA, Glenn MANSFIELD, Yoshiaki NEMOTO, " A Real-Time Instusion Dectection System(IDS) for Large Scale Networks and Its Evaluations" , IEICE, November, 1999

[5]. Vern Paxson, " Bro:A System for Detection Network Intruders in Real-Time" , USENIX, January, 1998

[6]. Phillip A. Porras, Alfonso Valdes, " Live Traffic Analysis of TCP/IP Gateways" , Networks and Distributed Systems Security Symposium, March, 1998

[7]. Joel Scambray, Stuart McClure, George Kurtz, " Hacking Exposed:Network Security Secrets & Solutions" ,McGraw-Hill, 2001

[8]. PLUS(포항공대 유닉스 보안 연구회), " Security PLUS for UNIX" , 영진출판사, 2000