스캔 기반의 인터넷 웜 공격 탐지 및 탐지룰 생성 시스템 설계 및 구현

김 익 수[†] · 조 혁[††] · 김 명 호[†††]

요 약

컴퓨터와 인터넷의 발달로 컴퓨터 사용자들은 유용한 정보를 쉽게 얻을 수 있게 되었다. 그러나 동시에, 시스템 불법 침입과 서비스 거부 공격에 의한 피해는 심각한 수준에 이르렀다. 특히 인터넷 웜을 통한 서비스 거부 공격은 컴퓨터와 네트워크 서비스를 무력화 시킬 수 있기 때문에 이에 대한 대처방안이 시급하다 할 수 있다. 지금까지 많은 침입 탐지 시스템들이 탐지룰을 기반으로 공격을 탐지해왔지만, 새롭게 등장하는 인터넷 웜을 탐지하는데 한계가 있다. 본 논문에서는 인터넷 웜이 여러 호스트들을 감염시키기 위해 네트워크 스캔 작업을 한다는 점에 착안하여, 스캔 기반의 인터넷 웜 공격을 효과적으로 탐지하기 위한 침입 탐지 및 탐지룰 생성 시스템을 제안한다. 제안된 시스템은 탐지룰을 기반으로 인터넷 웜 공격을 탐지하며, 탐지룰에 존재하지 않는 인터넷 웜에 의한 트래픽이 유입될 경우, 수집된 트래픽 정보를 통해 새로운 탐지룰을 생성하기 때문에 신종 인터넷 웜에 신속히 대응할 수 있다. 그리고 필요할 때만 패킷 데이터를 수집하기 때문에 시스템 부하와 디스크 사용량을 줄일 수 있다.

Design and Implementation of a System to Detect Intrusion and Generate Detection Rule against Scan-based Internet Worms

Ik-Su Kim[†] · Hyuk Jo[††] · Myung Ho Kim[†††]

ABSTRACT

The brilliant achievements in computers and the internet technology make it easy for users to get useful information. But at the same time, the damages caused by intrusions and denial of service attacks are getting more worse. Specially because denial of service attacks by internet worm incapacitate computers and networks, we should draw up a disposal plan against it. So far many rule-based intrusion detection systems have been developed, but these have the limits of these ability to detect new internet worms. In this paper, we propose a system to detect intrusion and generate detection rule against scan-based internet worm, paying attention to the fact that internet worms scan network to infect hosts. The system detects internet worms using detection rule. And if it detects traffic causing by a new scan-based internet worm, it generates new detection rule using traffic information that is gathered. Therefore it can response to new internet worms early. Because the system gathers packet payload, when it is being necessary only, it can reduce system's overhead and disk space that is required.

키워드 : 침입 탐지(Intrusion Detection), 인터넷 웜(Internet Worm), 포트 스캔(Port Scan)

1. 서 론

최근 컴퓨터와 인터넷의 발달은 정보통신 사업 발달에 중요한 촉매제 역할을 하고 있다. 개개인은 원하는 정보를 손쉽게 얻을 수 있게 되었으며, 쇼핑, 은행, 주식거래와 같은 업무도 인터넷을 통해 가능하게 되었다. 그리고 사업자들은 기업의 이익을 위해 인터넷 사업에 많은 투자를 하고 있는 것이 현실이다. 이에 반해 컴퓨터와 네트워크의 취약점을 악용한 공격자들의 불법적인 침입이 커다란 문제가 되고 있다. 특히, 최근 들어서는 인터넷 웜을 통한 시스템 불법 침입과 서비스 거부 공격이 심각한 수준에 이르렀다. 인터넷 웜은 서비스 거부 공격을 통해 피해 호스트의 컴퓨팅 자원과 네트워크 자원을 고갈시킴으로써 서비스를 무력화 시킬 수 있으며, 빠른 전파력을 지니고 있기 때문에 이에 대한 대처 방안이 시급하다[1]. 실례로 2003년 1월 25일에 발생한 MS-SQL 서버 웜 slammer에 의한 인터넷 대란은 현재 인터넷 서비스의 현실을 여실히 보여주는 사건이었다. MS-SQL 서버 웜 slammer는 취약성을 가지는 MS-SQL 서버에 침투하여 대량의 트래픽을 유발하고, 같은 취약성을 가지는 다른 호스트에 전파된다. 결과적으로 네트워크의 서비스는

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.
† 준 회 원 : 숭실대학교 대학원 컴퓨터학과 박사과정
†† 준 회 원 : 숭실대학교 대학원 컴퓨터학과 석사과정
††† 종신회원 : 숭실대학교 컴퓨터학부 부교수
논문접수 : 2004년 8월 24일, 심사완료 : 2005년 3월 10일

물론, DNS 서버의 부하를 증가시켜 인터넷 전반에 걸친 접속 지연과 서비스 불가 상태를 야기한다[2]. 2002년 9월 13일 처음 발견된 Linux slapper 웜은 OpenSSL 버퍼 오버플로우 취약점을 통해 취약 시스템의 nobody 권한의 쉘을 획득하여, 악성 프로그램을 설치하고 다른 컴퓨터에 서비스 거부 공격을 한다[3]. 이와 같이 공격자와 인터넷 웜은 특정 호스트가 지니고 있는 취약점을 악용하여 공격하기 때문에 취약점 정보 수집을 위한 스캔 공격이 선행된다. 특히 인터넷상에는 많은 보안 취약점 스캐너가 공개되어 있어 스캔 공격에 의한 트래픽을 조기에 탐지하는 것이 매우 중요하다. 이에 본 논문에서는 연구 범위를 스캔 기반의 공격 탐지에 초점을 두기로 한다.

공격자의 불법 행위를 신속히 탐지하고 대응하기 위해 많은 침입 탐지 시스템들이 개발되어 왔다. 특히 RTSD(Real Tme Scan Detector)는 국내 정보통신망에 대한 스캔 공격을 탐지하고 적극 대응하고자 한국정보보호진흥원에서 개발된 실시간 스캔 공격 탐지 도구로서, 스캔 공격을 탐지하면 관리자에게 이메일을 통해 통보해준다. 그러나 RTSD는 알고리즘의 단순성으로 인해 최근 문제가 되고 있는 신종 인터넷 웜을 탐지할 수 없다. 오픈 소스인 Snort는 포트 스캔 탐지는 물론, 많은 탐지룰을 보유하여 버퍼 오버플로우, CGI 공격 등의 다양한 공격을 탐지할 수 있다. 침입 탐지 기능 외에 네트워크 스니핑 모드와 패킷 로깅 모드를 지원하기 때문에 네트워크 트래픽을 분석하기 위한 도구로 활용할 수 있다. 하지만 Snort와 같이 룰 기반의 침입 탐지 시스템들은 패킷의 헤더 및 데이터를 탐지룰과 비교하기 때문에 대규모 네트워크 환경에서는 상당한 부하를 가지게 된다. 그리고 탐지룰을 기반으로 공격자의 불법 행위를 탐지하기 때문에 탐지룰을 벗어나는 새로운 인터넷 웜을 탐지하는데 한계가 있다. 따라서 이러한 보안 시스템들의 문제점을 개선하여 나날이 새롭게 등장하는 인터넷 웜을 탐지하기 위한 보안 시스템 개발이 필수적이라 할 수 있다.

본 논문에서는 대부분의 인터넷 웜이 네트워크상의 호스트들을 감염시키기 위해 여러 호스트로 스캔 작업을 한다는 점에 착안하여, 스캔 기반의 인터넷 웜 공격을 효과적으로 탐지하기 위한 침입 탐지 및 탐지룰 생성 시스템을 제안한다. 제안된 침입 탐지 시스템은 네트워크로부터 유입되는 패킷 정보를 탐지룰과 비교 분석하여 스캔 기반의 인터넷 웜을 탐지하고, 탐지룰에 존재하지 않는 스캔 기반의 공격을 탐지할 경우 패킷의 데이터 부분을 탐지룰 생성 시스템에게 전달한다. 탐지룰 생성 시스템은 패킷의 헤더만을 수집하며, 침입 탐지 시스템에 의해 전달받은 패킷의 데이터 부분과 수집된 패킷 헤더를 사용하여 새로운 스캔 기반의 신종 인터넷 웜을 탐지하기 위한 탐지룰을 자동으로 생성한다. 이를 통해, 기존의 룰기반 침입 탐지 시스템이 탐지하지 못하는 신종 인터넷 웜에 의한 공격을 조기에 탐지하여 신속히 대응할 수 있다. 그리고 평소에는 패킷의 헤더만을 수집하고, 의심스런 트래픽을 탐지할 경우에만 패킷의 데이터 부분을 수집하기 때문에 시스템에 의한 부하를 크게 감소시킬 수 있다. 탐지룰을 생성하기 위해 수집된 정보는 세션을 기반으로 한 누적된 헤더 정보이기 때문에, Snort의 패킷 로깅 모드에 의해 생성된 로그 파일에 비해 매우 작은 디스크 용량만을 필요로 한다.

본 논문의 구성은 다음과 같다. 2장에서는 스캔 기반의 공격도구와 이에 대응하기 위해 개발된 보안 시스템에 관해 살펴본다. 3장에서는 스캔 기반의 신종 인터넷 웜 탐지 및 탐지룰 생성을 위한 시스템에 대해 기술한다. 그리고 4장에서는 구현된 시스템에 대한 테스트 및 평가를 하고, 5장에서는 결론을 맺는다.

2. 관련 연구

이 장에서는 보안 취약점 스캐너 및 인터넷 웜과 이를 탐지하기 위해 개발된 보안 시스템에 대해 살펴보도록 한다. 공격자들은 스캔을 통해 공격하고자 하는 호스트의 서비스 여부와 취약점 정보를 수집하며, 수집된 정보를 악용하여 특정 권한을 획득한다. 특히, 최근에는 인터넷 웜을 통한 불법 침입과 서비스 거부 공격 방법이 등장하여 보안상 커다란 문제가 되고 있다.

2.1 보안 취약점 스캐너

보안 취약점 스캐너는 네트워크상에 존재하는 호스트들의 서비스 제공 여부와 보안상의 취약점을 점검하기 위해 개발되었다. 관리자는 스캐너를 통해 수집된 정보를 이용하여 해당 호스트의 취약점을 패치하거나 불필요한 서비스들을 제거함으로써 호스트의 보안성을 강화할 수 있다. 그러나 이러한 스캐너들은 시스템 공격자들에 의해 쉽게 사용될 수 있어서 오히려 인터넷 보안에 커다란 문제가 되고 있다.

Sscan은 johann sebastian bach가 개발한 보안 취약점 스캐닝 도구로 1999년 1월에 버전 0.1이 발표되었다. Sscan은 발표된 지 얼마 되지 않아서 많은 공개 사이트에서 소개되었으며 미국 CERT 팀에서도 그 위험성을 경고하였다. Sscan은 포트 스캔 기능은 물론, 네트워크 보안 취약점 점검 기능이 매우 강력해서 많은 수의 보안 취약점들을 점검할 수 있다. 또한 유닉스 시스템뿐만 아니라 윈도우즈 시스템에 많은 위협을 주는 백오리피스 진단도 가능하다. 그리고 자기 복제가 가능한 스크립트 파일을 통해 보안 취약점에 대한 적극적인 공격 까지 자동으로 이루어질 수 있어 관리자들의 주의가 필요하다[4].

Nessus는 Renaud Deraison에 의해 개발되기 시작하여 1998년 4월 첫 번째 버전이 발표되었으며, 2004년 8월 현재 2.0.8a 버전이 발표된 상태이다. Nessus는 공격자에 의해 이용될 수 있는 네트워크 취약점을 발견하여 공격자로부터 시스템을 어떻게 보호해야 하는지를 알려준다. Nessus는 클라이언트 서버 방식으로 운영되는데, 클라이언트는 윈도우 시스템이나 GTK가 설치되어 있는 유닉스 시스템에서 실행되어 편리한 사용자 인터페이스를 제공한다. 또한 클라이언트와 서버 사이를 오고가는 정보는 시스템의 취약점 정보를

포함하기 때문에 클라이언트 서버간의 통신에 암호화 기법을 사용한다[5].

Nmap은 네트워크 보안을 위한 스캐너로서, 시스템 관리자는 Nmap을 통해 네트워크에 어떤 호스트가 존재하고, 그들이 어떠한 서비스를 제공하며, 운영체제가 무엇인지를 점검할 수 있다[6]. 특히 오픈 소스이며 이식성이 좋아 대부분의 운영체제를 지원하고, 지속적인 업데이트를 통해서 다양한 스캔 방법을 제공한다. 그리고 Nessus의 모든 포트 스캔 방법을 포함하고 있을 뿐만 아니라, 여러 가지 옵션을 통해 강력한 스캔을 할 수 있는 장점이 있다. Nmap은 TCP 프로토콜을 이용한 Connect, SYN, FIN, NULL, XMAS 스캔은 물론, 다양한 프로토콜을 이용한 스캔 방법을 지원한다. TCP Connect Scan은 호스트와의 완전한 연결을 통해서 열린 포트를 검색하기 때문에, 정확하지만 해당 시스템에 로그를 남기는 단점이 있다. 그러나 Half-open 스캔이라 불리는 TCP SYN scan은 three-way handshake가 이루어지기 전에 RST 패킷을 보내 연결을 끊어버리기 때문에 해당 호스트에 로그를 남기지 않는다. 그리고 기존의 침입 탐지 시스템이 포트 스캔을 탐지하기 위해 SYN 패킷만을 감시한다는 점을 이용한 FIN, NULL, XMAS 스캔은 SYN을 제외한 플래그들을 설정하여 스캔함으로써 침입 탐지 시스템을 우회할 수 있다. 이러한 스캔 방법들은 대상 호스트가 BSD 계열의 시스템일 경우에만 적용 가능하며, 최근 개발된 침입 탐지 시스템은 패킷내의 모든 플래그를 감시함으로써 스캔 공격을 탐지할 수 있다.

Protocol	S_IP	D_IP	S_Port	D_Port	Length	Flag	Time
TCP	xxx.xxx.64.31	xxx.xxx.23.79	4270	1499	60	S	04.04.08.12.26.27
TCP	xxx.xxx.64.31	xxx.xxx.23.79	4271	280	60	S	04.04.08.12.26.27
TCP	xxx.xxx.64.31	xxx.xxx.23.79	4272	513	60	S	04.04.08.12.26.27
TCP	xxx.xxx.64.31	xxx.xxx.23.79	4273	397	60	S	04.04.08.12.26.27
			:	:			
TCP	xxx.xxx.64.31	xxx.xxx.23.79	4285	397	60	S	04.04.08.12.26.28

(a) TCP Connect Scan

TCP	xxx.xxx.64.31	xxx.xxx.23.79	47075	614	40	S	04.04.08.12.36.17
TCP	xxx.xxx.64.31	xxx.xxx.23.79	47075	16059	40	S	04.04.08.12.36.17
TCP	xxx.xxx.64.31	xxx.xxx.23.79	47075	1127	40	S	04.04.08.12.36.17
TCP	xxx.xxx.64.31	xxx.xxx.23.79	47075	794	40	S	04.04.08.12.36.17
			:	:			
TCP	xxx.xxx.64.31	xxx.xxx.23.79	47075	61	40	S	04.04.08.12.36.18

(b) TCP SYN Scan

TCP	xxx.xxx.64.31	xxx.xxx.23.79	34282	869	40	F	04.04.08.12.40.23
TCP	xxx.xxx.64.31	xxx.xxx.23.79	34282	896	40	F	04.04.08.12.40.23
TCP	xxx.xxx.64.31	xxx.xxx.23.79	34282	141	40	F	04.04.08.12.40.23
TCP	xxx.xxx.64.31	xxx.xxx.23.79	34282	448	40	F	04.04.08.12.40.23
			:	:			
TCP	xxx.xxx.64.31	xxx.xxx.23.79	34282	7100	40	F	04.04.08.12.40.24

(c) TCP FIN Scan

TCP	xxx.xxx.64.31	xxx.xxx.23.79	62011	1381	40		04.04.08.12.48.52
TCP	xxx.xxx.64.31	xxx.xxx.23.79	62011	542	40		04.04.08.12.48.52
TCP	xxx.xxx.64.31	xxx.xxx.23.79	62011	807	40		04.04.08.12.48.52
TCP	xxx.xxx.64.31	xxx.xxx.23.79	62011	808	40		04.04.08.12.48.52
			:	:			
TCP	xxx.xxx.64.31	xxx.xxx.23.79	62011	1459	40		04.04.08.12.48.53

(d) TCP NULL Scan

TCP	xxx.xxx.64.31	xxx.xxx.23.79	63234	366	40	UPF	04.04.08.12.50.32
TCP	xxx.xxx.64.31	xxx.xxx.23.79	63234	977	40	UPF	04.04.08.12.50.32
TCP	xxx.xxx.64.31	xxx.xxx.23.79	63234	148	40	UPF	04.04.08.12.50.32
TCP	xxx.xxx.64.31	xxx.xxx.23.79	63234	303	40	UPF	04.04.08.12.50.32
			:	:			
TCP	xxx.xxx.64.31	xxx.xxx.23.79	63234	1399	40	UPF	04.04.08.12.50.33

(e) TCP XMAS Scan

UDP	xxx.xxx.64.31	xxx.xxx.23.79	49468	2027	8		04.04.08.12.56.33
UDP	xxx.xxx.64.31	xxx.xxx.23.79	49468	2045	8		04.04.08.12.56.33
UDP	xxx.xxx.64.31	xxx.xxx.23.79	49468	1388	8		04.04.08.12.56.33
UDP	xxx.xxx.64.31	xxx.xxx.23.79	49468	896	8		04.04.08.12.56.33
			:	:			
UDP	xxx.xxx.64.31	xxx.xxx.23.79	49468	1443	8		04.04.08.12.56.34

(f) UDP Scan

(그림 1) Nmap을 이용한 포트 스캔에 의해 유입된 패킷 헤더 정보

(그림 1)은 Nmap이 지원하는 공격 방법들을 통해 포트 스캔을 한 결과이며, 포트 스캔에 의해 생성되는 패킷으로부터 포트 스캔의 특징을 살펴보기 위해 패킷 헤더에 포함되는 프로토콜, IP 주소, 포트 번호, 패킷 길이, 플래그를 나타내고 있다. 모든 포트 스캔 방법들이 짧은 시간에 하나의 목적지 IP 주소로 다수의 동일한 패킷을 전달하며, 목적지 포트 번호가 계속적으로 변하고 공격 방법에 따라 플래그가 다르게 설정되어 있다. 이러한 스캔을 통해 유입되는 패킷의 수는 공격 호스트와 대상 호스트의 위치에 따라 수개에서 수천 개까지 다양하다.

2.2 인터넷 웜

최근 들어 호스트의 취약점을 이용해 잠입하는 인터넷 웜에 의한 피해가 속출하고 있다. 인터넷 웜은 피해 호스트에 잠입한 이후 다른 취약 호스트를 감염시키기 위해 네트워크 스캔을 하며, 대량의 트래픽을 생성함으로써 특정 호스트나 네트워크 서비스를 불가능하게 한다.

Linux slapper 웜은 2002년 9월 13일 처음 발견된 리눅스 웜으로 아파치 웹서버를 대상으로 OpenSSL의 보안상 취약점을 이용하여 전파된다. 이 웜은 취약점을 가지는 웹서버를 찾기 위해 임의의 IP 주소로 GET 요청을 한다. 이때 웹서버의 응답 메시지가 "Apache" 문자열을 포함할 경우, SSL 서버 포트인 443번에 접속하여 SSL의 취약점을 이용한 공격을 시도한다. 웜에 감염된 시스템은 백도어가 남게 되며 분산 서비스 거부 공격을 위한 용도로 사용된다.

2003년 1월 25일 CERTCC-KR에서 긴급 경보를 내린 MS-SQL 서버 웜 slammer 공격은 일반적인 웜과 같이 파일형태로 저장되어 감염되는 것이 아니라 2001년 7월에 발견되었던 Code Red와 같이 메모리상에 상주하는 악성 코드이다. 포트 번호 1434를 사용하는 MS-SQL 서버를 공격 대상으로 하며 서버가 종료될 때까지 임의의 목적지 IP 주소를 가지는 패킷을 보내기 때문에, 감염된 시스템은 패킷 전송에 따른 시스템 부하가 증가하게 된다. 또한 웜에 감염된 서버가 늘어나면서 수 시간 내에 네트워크의 대부분을 UDP 공격 패킷이 차지하게 되고, 임의의 목적지 IP 주소를 가지는 패킷으로 인해 다른 네트워크에서 되돌아오는 ICMP Unreachable 패킷도 많아져 라우터의 부하가 증가된다. 이는 결과적으로 전체 네트워크가 느려지는 현상을 초래한다.

Protocol	S_IP	D_IP	S_Port	D_Port	Length	Time	Ascii Data
TCP	xxx.xxx.23.75	xxx.xxx.23.61	37778	80	70	04.04.01.16.34.33	GET / HTTP/1.1
TCP	xxx.xxx.23.75	xxx.xxx.23.62	37779	80	70	04.04.01.16.34.33	GET / HTTP/1.1
TCP	xxx.xxx.23.75	xxx.xxx.23.63	37780	80	70	04.04.01.16.34.33	GET / HTTP/1.1
				:			
				:			
TCP	xxx.xxx.23.75	xxx.xxx.23.68	37785	80	70	04.04.01.16.34.34	GET / HTTP/1.1

(a) Linux Slapper Worm

Protocol	S_IP	D_IP	Type	Code	Length	Time	Hexa_Data
ICMP	xxx.xxx.45.24	xxx.xxx.23.34	8	0	92	04.04.02.14.18.16	aaaaaaaaaaaaaaaa...
ICMP	xxx.xxx.45.24	xxx.xxx.23.38	8	0	92	04.04.02.14.18.16	aaaaaaaaaaaaaaaa...
ICMP	xxx.xxx.45.24	xxx.xxx.23.46	8	0	92	04.04.02.14.18.16	aaaaaaaaaaaaaaaa...
				:			
				:			
ICMP	xxx.xxx.45.24	xxx.xxx.23.51	8	0	92	04.04.02.14.18.17	aaaaaaaaaaaaaaaa...

(b) Welchia Worm

(그림 2) 인터넷 웜에 의해 유입된 패킷 정보

(그림 2)는 Linux Slapper 웜과 Welchia 웜에 의해 유입된 패킷 정보를 나타낸다. Nmap에 의한 포트 스캐닝과는 다르게, Linux Slapper 웜에 의한 스캔은 여러 호스트에 행해지며 TCP 패킷을 사용하여 같은 포트로 특정 데이터를 전송한다. Welchia 웜에 의한 스캔은 ICMP 프로토콜을 사용하여 특정 데이터를 전송한다. 이는 여러 호스트에 특정 애플리케이션이 실행되고 있는지 검색하거나 특정 서버의 취약점 정보를 수집 및 공격하기 위한 과정이다. 인터넷 웜은 포트 스캔과 달리 스스로 동작해 다른 호스트에 복제되며 전파 속도가 매우 빨라 인터넷 전체에 큰 영향을 줄 수 있다. 그러므로 인터넷 웜의 식별과 탐지는 새로운 웜에 의한 피해를 최소화하는 데에 있어 매우 중요하다 할 수 있다.

2.3 실시간 침입 탐지 시스템

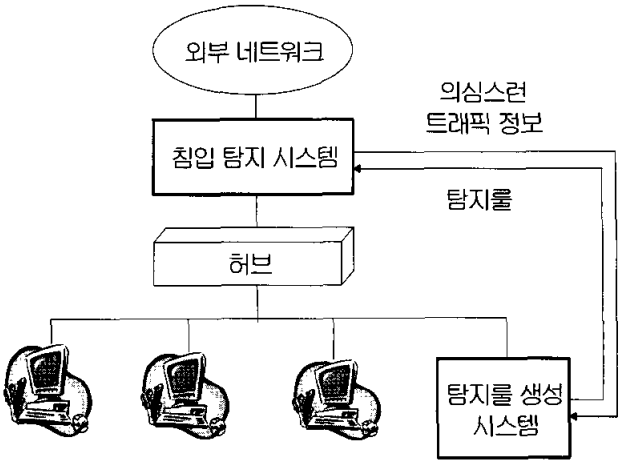
RTSD(Real Time Scan Detector)는 국내 정보통신망에 대한 스캔 공격을 탐지하고 적극 대응하고자 한국 정보보호진흥원에서 개발한 실시간 스캔 공격 탐지 도구이다[7]. RTSD의 스캔 공격 탐지 방법은 한 호스트에서 일정 시간 간격으로 일정한 수의 연결 요청이 있을 경우 취약점 검색 공격이라 간주한다. 기존의 스캔 도구들은 시스템의 서비스 여부를 판단하기 위해 TCP SYN 패킷만을 전송했지만 최근 스캔 도구들은 침입 탐지 시스템을 우회하기 위해 TCP, UDP, ICMP 프로토콜을 통한 다양한 방법으로 스캔 공격을 하고 있다. 이에 기존의 SYN 패킷만을 감시하던 RTSD 역시, 최근에는 여러 프로토콜을 통한 공격을 탐지한다. 하지만 RTSD는 알고리즘의 단순성으로 인해 탐지 기능이 스캔 공격에 국한된다는 단점이 있다.

Snort는 Martin Roesch에 의해 개발된 소프트웨어 기반의 실시간 네트워크 침입 탐지 시스템이다[8]. 실시간 트래픽 분석과 IP 네트워크상에서의 패킷 로깅이 가능하여 프로토콜 분석, 내용 검색, 매칭을 수행할 수 있으며 버퍼 오버플로우, Stealth 포트 스캔, CGI 공격, OS 확인 시도 등의 다양한 공격과 스캔을 탐지할 수 있다. Snort는 Chain Header와 Chain Options라 불리는 이차원의 linked list 안에 탐지룰을 가지고 있다. 탐지 처리를 빠르게 하기 위해 모든 Chain Option을 거치는 것이 아니라 공통의 IP와 Port를 가지는 공격에 대한 Chain Option이 일련의 linked list를 이루게 되므로 탐지 처리가 빠르게 수행될 수 있다. Snort는 이러한 탐지 처리 방식으로 빠르게 수행되며 침입에 대한 로그 역시 IP 주소별로 관리하기 때문에 로그분석에서도 효율적이다. 하지만 탐지룰을 기반으로 공격자의 불법 행위를 탐지하기 때문에 최근 문제가 되고 있는 신종 인터넷 웜을 탐지하는데 한계가 있다.

3. 스캐닝 기반의 인터넷 웜 공격 탐지 및 탐지룰 생성 시스템 구현

앞서 기술했듯이 시스템 공격자와 인터넷 웜에 의한 피해는 이미 심각한 수준에 이르렀다. 이러한 공격들은 호스트의 서비스 여부와 취약점 정보를 수집하기 위한 스캔 작업이 선행되기 때문에 이를 조기에 탐지하는 것은 보안상 매우 중요하다. 일반적으로 침입 탐지 시스템은 패킷 헤더와 데이터를 모두 감시하기 때문에 상당한 부하를 가져온다. 이는 분석되어야 할 패킷이 버려지는 결과를 가져오며, 나아가 침입 탐지시스템의 탐지 능력을 저하시킨다. 그리고 알려진 공격에 대한 탐지룰을 기반으로 공격자의 불법 행위를 탐지하기 때문에, 최근 문제가 되고 있는 신종 인터넷 웜을 탐지하는데 한계가 있다. 따라서 대량의 트래픽에서도 효율적이며, 새로운 인터넷 웜을 탐지할 수 있는 보안 시스템 구축이 필요하다.

이 장에서는 스캔 기반의 인터넷 웜 공격을 효과적으로 탐지하고 새로운 탐지룰을 생성하는 시스템을 구현한다. 구현 시스템은 평소에 패킷 헤더만을 수집하며, 의심스런 트래픽을 감지하였을 때만 패킷 데이터를 수집하기 때문에 대량의 트래픽이 유입되는 환경에서 효과적으로 작동한다. 또한 의심스런 트래픽이 유입될 경우, 해당 트래픽 정보를 통해 탐지룰을 생성할 수 있어 신종 인터넷 웜을 탐지하는데 효과적이다. 탐지룰을 생성하기 위해 수집된 정보들은 세션 기반으로 누적된 헤더 정보이기 때문에 매우 작은 디스크 용량만을 필요로 한다.



(그림 3) 침입 탐지 및 탐지룰 생성 시스템의 구성도

(그림 3)은 탐지룰 생성 시스템과 침입 탐지 시스템의 구성도이다. 대량의 트래픽이 유입되면 네트워크상의 트래픽을 감시하거나 수집하는 작업은 해당 호스트에 상당한 부하를 가져오기 때문에, 각각의 시스템을 서로 다른 호스트에서 운영함으로써 네트워크로부터 유입되는 트래픽 처리 능력을 향상시킬 수 있다. 침입 탐지 시스템은 기본적으로 포트 스캔을 탐지할 수 있으며, 스캔 공격 탐지 알고리즘과 탐지룰을 통해 인터넷 웜을 탐지한다. 탐지룰에 존재하지 않는 의심스런 트래픽을 탐지할 경우 해당 트래픽 정보를 탐지룰 생성 시스템에 전달한다. 탐지룰 생성 시스템은 주기적으로 생성한 로그 정보와 침입 탐지 시스템으로부터 전달받은 의심스런 트래픽 정보를 사용하여 새로운 탐지룰을 생성한다.

3.1 시간 흐름에 따른 스캔 기반의 공격 및 인터넷 서비스 트래픽 변화

최근 문제가 되고 있는 스캔 기반의 인터넷 웜을 탐지하기 위해서는 스캔 기반의 공격 유형에 따른 트래픽 분석이 매우 중요하다. 특히 침입 탐지 시스템의 탐지 오류율을 감소시키기 위해서는 공격에 의한 트래픽은 물론 정상적인 인터넷 서비스에 의한 트래픽의 비교 분석이 필요하다.

〈표 1〉 스캔 기반의 공격과 인터넷 서비스에 의해 발생하는 트래픽의 변화

트래픽 분석 대상	프로토콜	목적지 IP 주소	목적지 포트 번호	목적지 IP 주소 및 포트 번호가 연속적으로 변경된 패킷 수(초당)
Nmap	TCP/UDP	F	V	약 5개 이상
Welchia Worm	ICMP	V		약 5개 이상
Slapper Worm	TCP	V	F	약 5개 이상
SSH	TCP	F	F	0
HTTP	TCP	F or V	F	약 3개 이하
FTP	TCP	F	F	0
TELNET	TCP	F	F	0

<표 1>은 하나의 호스트로부터 스캔 기반 공격 및 인터넷 서비스 접속을 했을 때 발생하는 트래픽이 시간 흐름에 따라 어떻게 변하는지를 나타낸다. <표 1>에서 표기한 'F'는 시간 흐름에 따라 IP 주소와 포트 번호가 고정된다는 것을 의미하며 'V'는 IP 주소와 포트 번호가 변한다는 것을 의미한다. 2장에서 살펴본 바와 같이 Nmap을 통한 포트 스캔으로 생성되는 트래픽은 목적지 IP 주소가 동일하며, 시간 흐름에 따라 목적지 포트 번호가 연속적으로 변화하는 특성을 띄고 있다. 반면 Slapper 웜에 의한 스캔은 목적지 포트 번호가 동일하고, 시간 흐름에 따라 목적지 IP 주소가 연속적으로 변화한다. ICMP 프로토콜을 이용한 Welchia 웜 역시 시간 흐름에 따라 목적지 IP 주소가 연속적으로 변화한다. 그러나 정상적인 SSH, FTP, TELNET 서비스에 의한 트래픽은 스캔 기반의 공격과는 달리, 일단 호스트간의 세션이 연결되면 목적지 IP 주소나 목적지 포트 번호의 변화가 없다. 특히 HTTP 서비스의 경우에는 일반적으로 목적지 IP 주소가 고정되지만, 여러 서버를 통해 서비스를 제공할 경우 목적지 IP 주소가 변하기도 한다. 하지만 목적지 IP 주소의 변화는 연속적이지 않으며, 변화 회수 또한 작기 때문에 스캔 기반의 공격과 구별될 수 있다. 요약하면, 패킷의 목적지 IP 주소 및 포트 번호가 연속적으로 변경되는 수는 스캔 기반의 공격을 판단하기 위한 Threshold 값으로 사용될 수 있으며, 패킷 크기, 패킷 데이터, 프로토콜, 포트 번호에 대한 정보는 인터넷 웜에 대한 탐지룰을 생성하기 위해 사용될 수 있다.

SSH, FTP, TELNET과 같은 정상적인 트래픽은 세션이 연결되면 이를 통해 지속적인 통신이 이루어진다. 탐지룰 생성 시스템은 지속적으로 패킷 헤더 정보를 수집해야 하므로 상당한 디스크 용량을 필요로 한다. 따라서 정상적인 트래픽이 세션 기반으로 지속적인 통신을 한다는 점에 착안하여, 일정시간 동안 세션별로 패킷 정보를 누적하여 저장하면 중복된 정보에 따른 디스크 공간의 낭비를 줄일 수 있다.

3.2 실시간 침입 탐지 시스템

본 논문에서는 트래픽을 비정상적인 트래픽, 의심스런 트래픽, 정상적인 트래픽으로 구분한다. 비정상적인 트래픽은 침입 탐지 시스템이 보유하고 있는 탐지룰과 일치하는 대량의 패킷들을 의미하고, 의심스런 트래픽은 탐지룰에 존재하지는 않지만 스캔 공격에 의해 발생하는 패턴과 유사한 대량의 패킷들을 의미한다. 정상적인 트래픽은 비정상적인 트래픽과 의심스런 트래픽에 속하지 않는 일반적인 트래픽을 의미한다.

포트 스캐닝은 특정 호스트가 제공하는 서비스들을 검색하기 위해 여러 포트로 스캐닝을 한다. 그리고 인터넷 웜은 여러 호스트에 대해 특정 서비스의 취약점을 공격하기 위해 하나의 포트로 스캐닝을 한다. 즉, 포트 스캐닝 도구에 의한 공격으로 발생하는 트래픽은 근원지/목적지 IP 주소가 동일하며, 짧은 시간에 목적지 포트 번호가 계속적으로 변화하는 특성을 가진다. 그리고 웜에 의한 스캔은 다른 호스트로의 전파를 위해 목적지 IP를 계속적으로 변경하면서 하나의 특정 포트를 스캔하게 된다. 특히 이러한 공격 도구는 동일한 패킷을 다수 생성하기 때문에 침입 탐지 시스템은 이와 같은 특징을 이용하여 공격자의 불법 행위를 탐지할 수 있다. 이러한 불법적인 트래픽은 다음과 같은 방법을 통해 탐지할 수 있다.

- 포트 스캐닝

한 호스트로부터 짧은 시간 내에 다수의 유입되는 TCP/UDP 패킷들의 목적지 IP 주소가 모두 같고 목적지 포트 번호가 모두 다를 경우 포트 스캐닝으로 판단

- 인터넷 웜

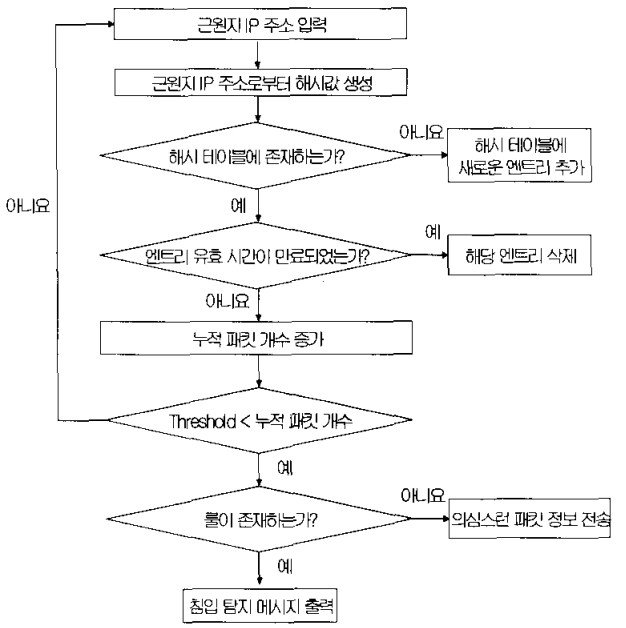
한 호스트로부터 짧은 시간 내에 다수의 유입되는 TCP/UDP 패킷들의 목적지 IP 주소가 모두 다르고 목적지 포트

번호가 모두 같을 경우 인터넷 웜으로 판단

한 호스트로부터 짧은 시간 내에 다수의 유입되는 ICMP 패킷들의 목적지 IP 주소가 모두 다를 경우 인터넷 웜으로 판단

앞서 기술했듯이 정상적인 SSH, FTP, TELNET 서비스에 의한 트래픽은 일단 호스트간의 세션이 연결되면, 목적지 IP 주소나 목적지 포트 번호의 변화가 없기 때문에 침입 탐지 시스템은 정상적인 트래픽으로 판단한다. 또한 HTTP 서비스의 경우에도 목적지 IP 주소의 변화가 일시적이고 변화 회수가 작기 때문에 정상적인 트래픽으로 판단한다.

침입 탐지 시스템은 포트 스캐닝과 인터넷 웜에 의한 트래픽을 탐지하기 위해 호스트로부터 유입되는 패킷들의 헤더 정보를 IP 주소별로 일정 시간 동안 저장해야 한다. 이들 정보를 저장하기 위해서는 해시 테이블을 사용하며, 해시 값을 생성하기 위한 입력 값으로 근원지 IP 주소를 사용한다. 해시 테이블의 각 엔트리는 근원지/목적지 IP 주소, 근원지/목적지 포트 번호, 패킷의 길이, 엔트리의 유효 시간, 누적 패킷의 개수를 나타내는 멤버 변수를 가진다. 엔트리의 유효 시간은 불법적인 트래픽을 탐지하기 위해 패킷 헤더 정보가 저장되는 기간을 나타내고, 누적 패킷의 개수는 다수의 유입되는 패킷 수를 저장하기 위해 사용된다.



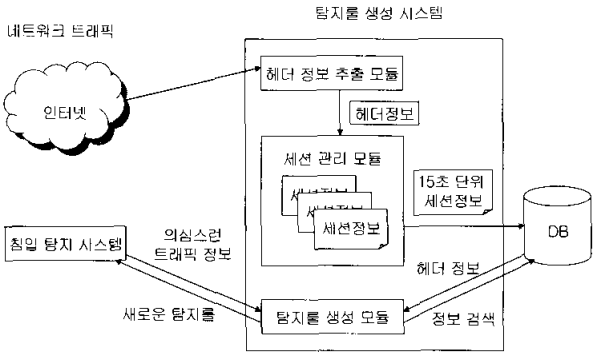
(그림 4) 침입 탐지 시스템 동작 순서도

(그림 4)는 침입 탐지 시스템의 동작 순서도를 나타낸다. (그림 4)에서 볼 수 있듯이 침입탐지 시스템은 처음 유입된 패킷 헤더 정보를 저장하기 위해 근원지 IP 주소로 해시 값을 생성하여 새로운 엔트리를 생성한다. 이후에 유입되는 패킷들은 해시 테이블에 존재하고 엔트리 유효 시간이 만료되지 않았을 경우, 누적 패킷의 개수를 나타내는 변수를 증가시킨다. 그리고 누적 패킷의 개수가 Threshold 값을 넘게 되면 의심스런 패킷들로 간주한다. 그러나 엔트리의 유효 시간이 만료될 경우에는 정상적인 패킷으로 간주하고 리스트에서 제거된다.

침입 탐지 시스템은 탐지룰을 포함하는데 이는 탐지룰 생성 시스템에 의해 생성되는 공격 패턴이다. 침입 탐지 시스템이 의심스런 패킷을 탐지하면 탐지룰을 검색하여 해당 공격에 대한 탐지 메시지를 출력한다. 그러나 일치하는 탐지룰이 존재하지 않을 경우에는 해당 패킷의 헤더 및 데이터 정보와 탐지 시간을 탐지룰 생성 시스템에게 전달한다. 침입 탐지 시스템은 의심스런 패킷을 탐지했을 경우에만 패킷의 데이터를 캡쳐하기 때문에 시스템의 부하를 줄일 수 있다. 탐지룰 생성 시스템에 전달되는 헤더 정보는 근원지 IP 주소와 스캔 기반의 공격을 탐지하기 위해 설정된 Threshold 값, 목적지 IP 주소 및 포트 번호의 변화 여부에 관한 정보가 포함된다.

3.3 탐지룰 생성 시스템

탐지룰 생성 시스템은 침입 탐지 시스템에 의해 의심되는 트래픽이 탐지되었을 때 전달되는 정보를 사용하여 새로운 공격에 대한 탐지룰을 생성한다.



(그림 5) 탐지룰 생성 시스템의 내부 구조

(그림 5)는 탐지룰 생성 시스템의 내부 구조를 나타내며, 네트워크로부터 유입된 패킷의 헤더를 추출하기 위한 헤더 정보 추출 모듈과 헤더 정보를 효율적으로 저장하기 위한 세션 관리 모듈, 알려지지 않은 새로운 인터넷 웜에 대한 탐지룰을 생성하는 탐지룰 생성 모듈로 구성된다.

3.3.1 헤더 정보 추출 모듈

헤더 정보 추출 모듈은 Libpcap 라이브러리를 사용하여 네트워크로부터 유입된 패킷을 수집한다. 이 모듈은 수집된 패킷으로부터 프로토콜, 근원지/목적지 IP 주소, 근원지/목적지 포트 번호, 패킷 길이 정보만을 추출하여 세션 관리 모듈에게 전달한다. Libpcap 라이브러리를 사용하면 사용자 수준에서 시스템에 상관없이 쉽게 패킷 수집을 할 수 있고, 시스템과 운영체제에 따라 패킷 수집을 가능하도록 각기 다른 인터페이스를 제공하기 때문에 다른 시스템으로의 이식성이 좋다.

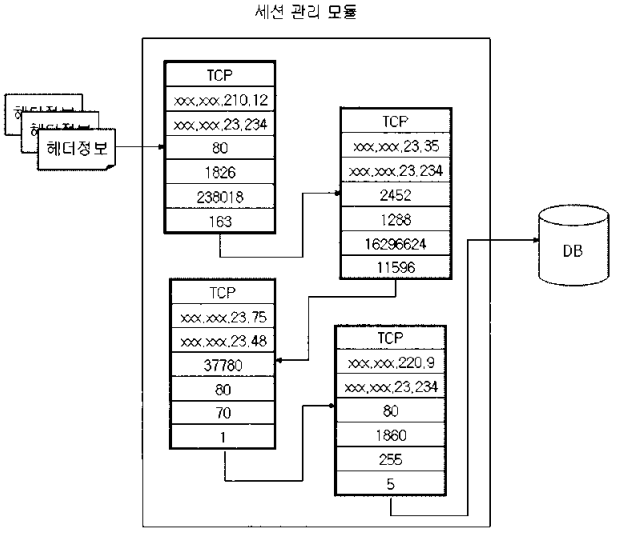
ICMP(근원지 IP 주소)(목적지 IP 주소)(패킷 길이)
TCP(근원지 IP 주소)(목적지 IP 주소)(근원지 포트 번호)(목적지 포트 번호)(패킷 길이)
UDP(근원지 IP 주소)(목적지 IP 주소)(근원지 포트 번호)(목적지 포트 번호)(패킷 길이)

(그림 6) 헤더 정보 추출 모듈을 통해 생성되는 헤더 정보 포맷

(그림 6)은 헤더 정보 추출 모듈을 통해 생성되는 헤더 정보 포맷을 나타낸다. 네트워크로부터 유입된 대부분의 패킷은 데이터 부분을 포함하고 있기 때문에 전체 패킷을 저장할 경우 짧은 시간 내에 제한된 디스크 용량을 초과할 수 있다. 탐지룰 생성을 위해서는 패킷 데이터가 필요하지만 이는 앞서 기술한 침입 탐지 시스템이 공격을 탐지했을 경우에 탐지룰 생성 모듈에 전달되기 때문에 헤더 정보 추출 모듈에서는 단지 패킷 헤더만을 수집함으로써 디스크 용량 초과 문제를 해결할 수 있다.

3.3.2 세션 관리 모듈

헤더 정보 추출 모듈로부터 전달된 정보는 네트워크로부터 유입된 모든 패킷들의 헤더 정보들이다. 비록 패킷 데이터는 제거된 상태지만 저장된 상당량의 정보는 새로운 탐지룰을 생성하기 위해 사용되기 때문에 시스템 부하를 최소화하기 위해 헤더 정보의 효율적인 관리가 필요하다. 이에 세션 관리 모듈은 불필요하게 중복된 정보들을 제거함으로써 디스크 사용량은 물론, 탐지룰을 생성하기 위한 작업량을 크게 감소시킨다.



(그림 7) 세션 관리 모듈에 의한 헤더 정보 관리

(그림 7)은 헤더 정보 추출 모듈로부터 전달된 헤더 정보가 세션 관리 모듈에 의해 연결 리스트로 관리되는 모습을 나타낸다. 각각의 노드는 프로토콜, 근원지 IP 주소 및 포트 번호 기반으로 생성된 세션 정보 엔트리이며, 각각의 엔트리들은 메모리상에서 15초간 유지되면서 누적된 헤더 정보를 유지한다. 시간이 만료될 경우 누적된 정보를 가지는 엔트리들이 데이터베이스에 저장되기 때문에, 한번 세션이 연결된 이후 지속적인 통신이 이루어지는 TELNET, SSH, FTP와 같은 트래픽 정보의 양을 크게 줄일 수 있다.

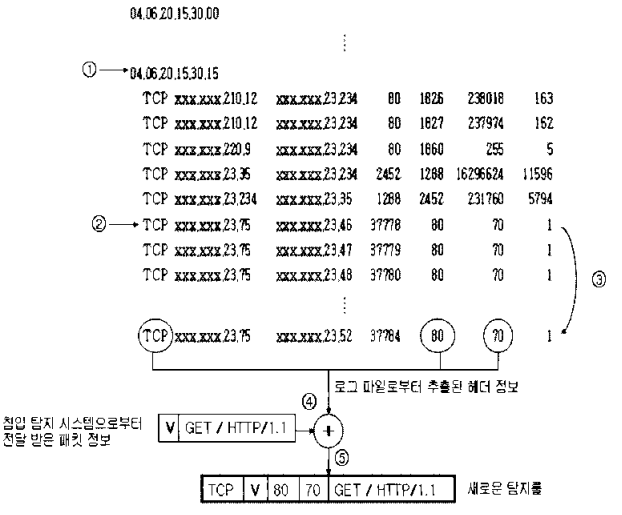
15초 단위 시간 정보
ICMP(근원지 IP 주소)(목적지 IP 주소)(누적 패킷 길이)(누적 패킷 수)
TCP(근원지 IP 주소)(목적지 IP 주소)(근원지 포트 번호)(목적지 포트 번호)(누적 패킷 길이)(누적 패킷 수)
UDP(근원지 IP 주소)(목적지 IP 주소)(근원지 포트 번호)(목적지 포트 번호)(누적 패킷 길이)(누적 패킷 수)

(그림 8) 세션 관리 모듈을 통해 생성되는 헤더 정보 포맷

(그림 8)은 세션 관리 모듈을 통해 생성되는 헤더 정보에 대한 포맷이다. 15초 단위 시간 정보는 로그 기록 시간을 나타내며, 이후의 로그들은 헤더 정보를 나타낸다. 헤더 정보의 첫 번째 필드는 해당 패킷의 프로토콜을 나타내며 TCP, UDP 패킷의 경우에는 근원지/목적지 IP 주소와 근원지/목적지 포트 번호, 15초간 누적된 패킷들의 총 길이, 15초간 누적된 패킷 수를 나타낸다. ICMP 패킷의 경우에는 근원지/목적지 포트 번호가 없다는 것을 제외하고 TCP/UDP 패킷과 동일한 정보를 가진다. 또한 세션 관리 모듈은 헤더 정보를 저장하기 위한 파일을 하루 단위로 생성한다.

3.3.3 탐지룰 생성 모듈

탐지룰 생성 모듈은 신종 인터넷 웜에 대한 탐지룰을 생성하기 위한 모듈로서 침입 탐지 시스템에 의해 알려지지 않은 스캔 기반의 공격이 탐지될 경우 동작한다. 이때 침입 탐지 시스템은 공격에 의해 유입된 패킷 정보를 탐지룰 생성 모듈에 전달하며, 탐지룰 생성 모듈은 전달된 정보와 세션 관리 모듈로부터 생성된 로그 정보를 이용하여 새로운 탐지룰을 생성한다.



(그림 9) 탐지룰 생성 모듈에 의한 탐지룰 생성 절차

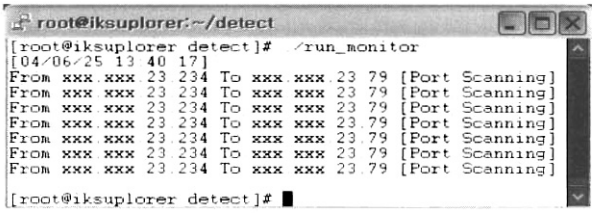
(그림 9)는 탐지룰 생성 모듈이 탐지룰을 생성하는 절차를 나타내며, 탐지룰을 생성하기 위한 알고리즘은 다음과 같다.

① 침입 탐지 시스템으로부터 전달받은 탐지 시간 정보를 이용하여 로그파일에서 15초 단위로 저장된 로그 그룹을 찾는다.
② 로그 그룹에서 침입 탐지 시스템으로부터 전달받은 근원지 IP 주소와 일치하는 행들을 찾는다.
③ 목적지 IP 주소 및 포트 번호의 변화 회수와 침입 탐지 시스템으로부터 전달받은 Threshold 값을 비교한다.
④ Threshold 값을 초과한 행으로부터 프로토콜과 목적지 포트 번호, 패킷 길이를 추출한다.
⑤ 추출된 정보와 침입 탐지 시스템으로부터 전달받은 패킷 데이터 및 목적지 IP 주소의 변화 여부에 관한 정보를 사용하여 새로운 탐지룰을 생성한다.

탐지룰 생성 모듈에 의해 생성된 새로운 탐지룰은 실시간으로 침입 탐지 시스템에 전달되어 신종 인터넷 웜을 탐지하기 위해 사용된다.

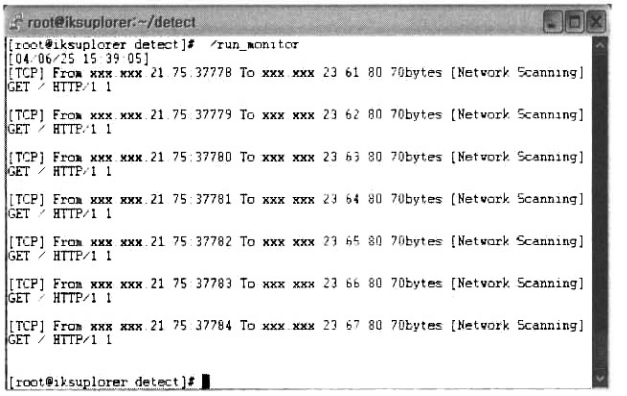
4. 실 험

구현 시스템은 리눅스 서버에서 운영되며, 실험을 위한 스캔 기반의 공격 도구로는 Linux slapper 웜과 Nmap을 사용하였다. 실험에서는 스캔 기반의 공격을 감지하기 위한 Threshold 값을 7로 설정하였으며 엔트리 유효 시간은 1초로 설정하였다. Threshold 값과 엔트리 유효 시간에 대한 설정값은 침입 탐지 시스템의 탐지율을 높이기 위해서 네트워크 환경에 따라 관리자가 적당한 값으로 설정할 필요가 있다. 그리고 탐지룰 생성 시스템에 의해 수집된 로그파일의 디스크 사용량과 Snort에 의해 수집된 로그파일의 디스크 사용량을 비교함으로써 제안 시스템의 효율성을 평가한다.

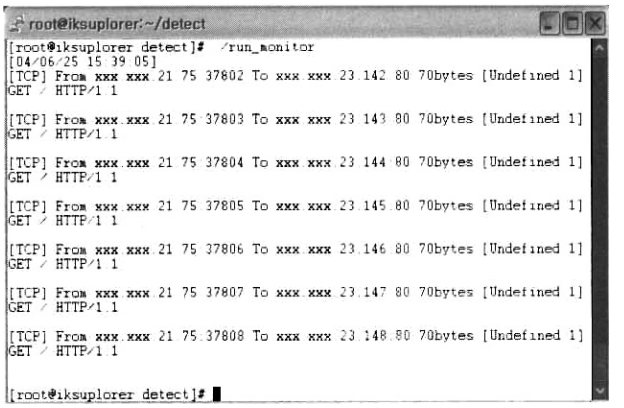


(그림 10) 침입 탐지 시스템의 포트 스캐닝 탐지

(그림 10)은 침입 탐지 시스템이 Nmap에 의한 포트 스캐닝을 탐지한 결과이다. 앞서 기술했듯이 침입 탐지 시스템은 한 호스트로부터 짧은 시간 내에 다수의 유입되는 패킷들의 목적지 IP 주소가 모두 같고 목적지 포트 번호가 모두 다를 경우 포트 스캐닝으로 판단하기 때문에 탐지룰 파일의 사용 없이 탐지할 수 있다.

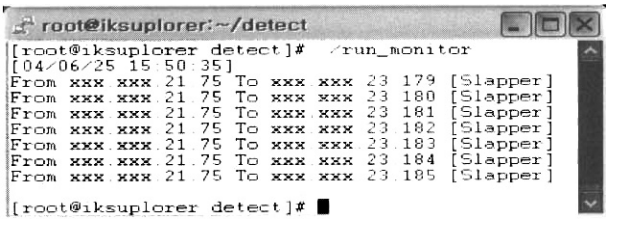


(그림 11) slapper 웜에 대한 탐지룰이 존재하지 않을 때의 탐지 결과



(그림 12) 탐지룰 생성 시스템에 의해 탐지룰이 추가된 이후의 slapper 웜 탐지

(그림 11)과 (그림 12)는 slapper 웜에 대한 탐지룰 생성 과정을 나타낸다. 일반적으로 인터넷 웜은 여러 호스트를 감염시키기 위해 호스트의 서비스 여부 및 취약점 정보 수집을 위한 스캐닝을 수행한다. 침입 탐지 시스템은 초기에 slapper 웜에 대한 탐지룰 정보가 없기 때문에 네트워크에 대한 스캔 탐지 결과로 "Network Scanning" 메시지를 보여준다. 이 단계에서는 현재 공격중인 웜을 식별할 수는 없지만, 시스템 관리자는 신속하게 외부로부터 시스템이 공격을 받고 있다는 것을 확인할 수 있다. 탐지룰 생성 시스템에 의해 탐지룰이 추가된 이후에는 "Undefined #" 메시지가 탐지룰에 추가되며, (그림 13)과 같이 관리자가 새롭게 생성된 탐지룰에 "Slapper"라는 이름을 지정하면 침입 탐지 시스템은 해당 공격을 구체적으로 명시한다.



(그림 13) 공격 이름을 지정한 후의 slapper 웜 탐지

이와 같이 구현 시스템은 slapper 웜에 의한 스캔 기반의 의심스런 트래픽을 탐지하고, 이 때 수집된 패킷의 헤더와 데이터를 추출하여 slapper 웜을 탐지하기 위한 탐지룰을 자동으로 생성할 수 있었다. 따라서 구현 시스템은 기존의 룰 기반 침입 탐지 시스템이 탐지하지 못하는 스캔 기반의 신종 인터넷 웜 공격을 조기에 발견하여 탐지룰 생성함으로써 이들 공격에 신속히 대응할 수 있다.

〈표 2〉 시간별 유입된 트래픽 양과 탐지룰 생성 시스템 로그 파일 크기

구간 (30분 단위)	트래픽 양 (Mbyte)	탐지룰 생성 시스템 로그 파일 크기 (Mbyte)
1	65	0.395
2	11	0.439
3	24	0.343
4	5	0.367
5	18	5

<표 2>는 30분 단위로 유입된 트래픽 양과 탐지룰 생성 시스템이 생성하는 로그 파일의 크기를 나타낸다. 1구간은 파일 전송을 통해 트래픽이 증가한 구간을 나타내며, 2-4구간은 일반적인 트래픽 구간을 나타낸다. 그리고 5구간은 포트 스캐닝에 의한 트래픽이 포함된 구간이다. 포트 스캐닝에 의한 트래픽이 포함된 구간에서는 로그 파일의 크기가 트래픽 양에 비해 72.3%의 감소율을 보인 반면, 정상적인 트래픽 구간인 1-4구간에서 생성된 로그 파일 크기는 약 90% 이상의 감소율을 보이고 있다. 특히 탐지룰 생성 시스템은 세션 기반으로 일정시간 동안 누적된 정보를 기록하기 때문에, 1구간에서와 같이 많은 양의 데이터가 전송되어도 정상적인 트래픽일 경우에는 로그 파일의 크기에 크게 영향을 미치지 않는다.

탐지룰 생성 시스템의 로그 파일은 트래픽의 헤더 정보를 기록하기 때문에 Snort의 패킷 로그 모드에 의해 생성된 파일과 유사하다. <표 3>은 탐지룰 생성 시스템과 Snort에 의해 생성된 로그 파일의 크기를 나타낸다.

〈표 3〉 탐지룰 생성 시스템과 Snort의 시간별 로그 파일 크기

구간 (30분 단위)	Snort 로그 파일 크기 (Mbyte)	탐지룰 생성 시스템 로그 파일 크기(Mbyte)
1	10.067	0.131
2	157.540	3.928
3	155.475	0.135

1구간에서 생성된 로그 파일은 일반적인 트래픽에 의해 생성된 로그 파일로서 Snort 로그 파일의 총 크기에 비해 탐지룰 생성 시스템의 로그 파일은 98.7%의 감소율을 보이고 있다. 2구간은 포트 스캐닝에 의한 트래픽이 포함된 구간이며, Snort의 경우 로그 파일의 총 크기가 매우 큰 것을 알 수 있다. Snort는 각 세션에 대해 로그 파일을 생성하기 때문에 한 호스트의 전체 포트 스캔은 65536개의 로그 파일을 생성하게 된다. 실험에서 포트 스캔을 한 결과 snort는 각 포트에 대해 4096byte의 로그 파일을 생성하는 것을 확인했으며, 따라서 전체 포트에 대해 스캔할 경우 약 268Mbyte의 디스크 공간을 차지하게 된다. 3구간은 FTP를 통해 파일 전송이 이루어진 구간으로 탐지룰 생성 시스템의 경우 같은 세션의 트래픽을 15초 단위로 누적하여 기록하기 때문에 디스크 사용량이 작은 반면, Snort 로그 파일은 패킷당 로그를 남기기 때문에 디스크 사용량이 매우 큰 것을 알 수 있다. 요약하면, 구현 시스템을 통해 생성된 로그파일은 네트워크로부터 유입된 트래픽 양과 Snort에 의해 생성된 로그 파일에 비해 매우 작은 크기이기 때문에, 탐지룰 생성 시스템이 새로운 탐지룰을 생성하기 위해 처리해야 할 작업량이 크게 감소한다는 것을 의미한다.

5. 결 론

컴퓨터와 인터넷의 발달에 따라 최근 대두되고 있는 개인 정보 유출과 서비스 거부 공격에 의한 피해는 이미 심각한 수준에 이르렀다. 특히 인터넷 웜을 통한 서비스 거부 공격은 대상 호스트의 권한 획득과 관계없이 피해 호스트의 컴퓨팅 자원과 네트워크 자원을 소모함으로써 서비스를 무력화 시킬 수 있기 때문에 이에 대한 대처 방안이 시급하다. 그리고 시스템 공격 방법은 날이 갈수록 다양해져서 기존의 네트워크 보안 시스템은 이러한 공격에 즉각적인 대응을 할 수 없는 실정이다.

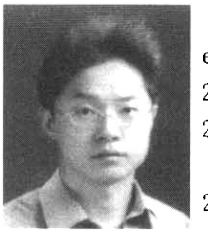
이에 본 논문에서는 스캔 기반의 인터넷 웜 공격을 효과적으로 탐지하기 위한 침입 탐지 및 탐지룰 생성 시스템을 제안하였다. 인터넷 웜이 여러 호스트를 감염시키기 위해 네트워크상의 취약 시스템을 검색한다는 점에 착안하여, 침입 탐지 시스템은 스캔 공격을 탐지하면 탐지룰에 따라 해당 공격을 알려준다. 그러나 일치하는 탐지룰이 존재하지 않을 경우에는 해당 패킷의 헤더 및 데이터 정보와 탐지 시간을 탐지룰 생성 시스템에게 전달하여 새로운 탐지룰을 생성한다. 그리고 침입 탐지 시스템의 부하를 최소화하기 위해 평소에는 패킷의 헤더만을 수집하며, 의심스런 트래픽을 탐지할 경우에만 패킷의 데이터 부분을 수집하도록 하였다. 또한, 탐지룰 생성 시스템은 디스크 사용량을 줄이기 위해서 같은 세션의 트래픽을 15초 단위로 누적하여 기록했기 때문에, 각 패킷마다 로그를 남기는 Snort와 비교하여 약 97% 이상의 감소율을 보였다. 이러한 감소율은 탐지룰 생성 시스템이 새로운 탐지룰을 생성하기 위해 처리해야 할 작업량이 크게 감소한다는 것을 의미한다.

참 고 문 헌

[1] 한국정보보호진흥원, "2004년 04월 해킹바이러스 통계 및 분석 월보", 2004.

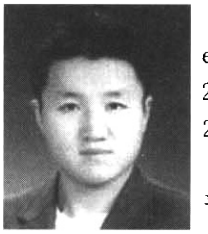
[2] 전완근, 류성철, 김승철, "MS-SQL 서버 웜-슬래머 공격 테스트 및 사고 대응", 2003.

[3] http://www.cert.org/advisories/CA-2002-27.html
[4] 정현철, "Sscan 분석 보고서", 1999.
[5] http://www.nessus.org
[6] 전 숙, "Nmap 네트워크 점검 도구 및 보안 스캐너"
[7] 이현우, 이상엽, 정현철, 정윤종, 임채호, "Analysis of Large Scale Network Vulnerability Scan Attacks and Implementation of the Scan-Detection tool", 1999.
[8] MartinRoesch, "Snort-Lightweight Intrusion Detection for Networks," Proc. of LISA '99: 13th Systems Administration Conference, Nov., 1999.
[9] 정현철, 변대용, "트래픽 분석을 통한 서비스 거부 공격 추적", 2003.
[10] 박현미, 오은숙, 이동련, "IP 네트워크 scanning 기법", 2002.
[11] Fyodor, "The Art of Port Scanning," Phrack Magazine Volume 7 Issue 51, 1997.
[12] Dfir Arkin, "ICMP Usage in Scanning," 2001.
[13] Joseph Reves, Sonia Panchen, "Traffic Monitoring with Packet-Based Sampling for Defense against Security Threats"
[14] P. Phaal, S. Panchen, N. McKee, "InMon Corporations's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks," InMon Corp, Sep., 2001.
[15] Guo Xiaobing, Qian Depei, Liu Min, Zhang Ran, Xu Bin, "Detection and Protection against Network Scanning: IEDP," Proc. of the 2001 IEEE International Conference on Computer Networks and Mobile Computing, pp.487-493, Oct., 2001.
[16] http://securityresponse.symantec.com/avcenter/venc/data/w32.welchia.worm.html
[17] http://securityresponse.symantec.com/avcenter/venc/data/linux.slapper.worm.html
[18] R. Russell and A. Machie, "Code Red Worm," Tech. Rep, Incident Analysis, SecurityFocus, Aug. 2001.
[19] A. Machie, J. Roculan, R. Russell, and M. V. Velzen, "Nimda Worm Analysis," Tech. Rep, Incident Analysis, SecurityFocus, Sept., 2001.
[20] 홍석범, "시스템 및 네트워크 모니터링을 통한 보안 강화", 오늘과 내일 넷센터.
[21] Stephen Northcutt, Judy Novak, 'Network Intrusion Detection An Analyst's Handbook', 2nd Ed., New Riders, 2000.
[22] http://www.securitymap.net
[23] http://packetstormsecurity.org



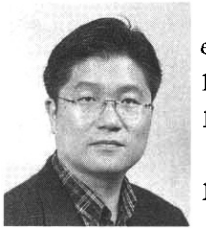
김 익 수

e-mail : skycolor@ss.ssu.ac.kr
2000년 숭실대학교 컴퓨터학부(학사)
2002년 숭실대학교 대학원 컴퓨터학과 (공학석사)
2004년 숭실대학교 대학원 컴퓨터학과 박사과정
관심분야 : 리눅스 커널, 컴퓨터 보안, 정보보호



조 혁

e-mail : laphael7@ss.ssu.ac.kr
2002년 평택대학교 정보통계학과(학사)
2002년~현재 숭실대학교 대학원 컴퓨터학과 석사과정
관심분야 : 네트워크 보안, 분산처리, 데이터 마이닝



김 명 호

e-mail : kmh@comp.ssu.ac.kr
1989년 숭실대학교 전자계산학과(학사)
1991년 포항공과대학교 전자계산학과(공학석사)
1995년 포항공과대학교 전자계산학과(공학박사)
1995년 한국전자통신연구소 선임연구원
1998년~1999년 University of Tennessee 전자계산학과 교환교수
1995년~현재 숭실대학교 컴퓨터학부 부교수
관심분야 : 병렬/분산처리, 컴퓨터 보안, BI, 클러스터링, 리눅스