인터넷 웜 공격으로부터 서버를 보호하기 위한 악성 프로세스 제어 시스템

정회원 김 익 수*

A Malicious Process Control System for Protecting Servers from Internet Worm Attacks

Ik-Su Kim* *Regular Member*

요 약

탐지룰을 기반으로 하는 보안 시스템들은 신종 인터넷 웜에 대응할 수 없다는 문제가 있다. 본 논문에서는 탐지룰을 사용하지 않고 인터넷 웜 공격으로부터 서버를 보호하는 악성 프로세스 및 실행파일 제거 시스템을 제안한다. 제안 시스템은 보호대상서버와 동일한 서비스 프로그램을 구동하면서 인터넷 웜에 의한 멀티캐스팅 공격을 탐지하는 제어서버와 제어서버의 지시에 따라 보호대상서버에 생성된 악성 프로세스와 악성 실행파일을 제거하는 에이전트로 구성된다. 제안 시스템은 탐지룰을 사용하지 않기 때문에 신종 인터넷 웜에 효과적으로 대응할 수 있으며, 기존의 보안 시스템들과 통합될 경우 보안을 더욱 강화할 수 있다.

Key Words : Internet Worm, Intrusion Prevention, Zero-day Attack, Malicious Process, Signature

ABSTRACT

The security systems using signatures cannot protect servers from new types of Internet worms. To protect servers from Internet worms, this paper proposes a system removing malicious processes and executable files without using signatures. The proposed system consists of control servers which offer the same services as those on protected servers, and agents which are installed on the protected servers. When a control server detects multicasting attacks of Internet worm, it sends information about the attacks to an agent. The agent kills malicious processes and removes executable files with this information. Because the proposed system do not use signatures, it can respond to new types of Internet worms effectively. When the proposed system is integrated with legacy security systems, the security of the protected server will be further enhanced.

Ⅰ. 서 론

최근 인터넷 웜을 통한 시스템 불법 침입과 서비스 거부 공격으로 인한 피해 사례가 증가하고 있다. 호스트를 감염시킨 인터넷 웜은 호스트의 관리자 권한을 탈취하여 정보 유출, 삭제, 변조에 의한 피해를 주기도 하며, 대량의 트래픽을 유발함으로써 해당 호스트는 물론 네트워크의 자원을 고갈시켜 인터넷 서비스를 불가능하게 한다. 이러한 인터넷 웜들은 다수의 호스트들을 짧은 시간 내에 감염시키기 위해 악의적인 패킷을 멀티캐스팅 한다.

지금까지 다양한 인터넷 웜 공격을 실시간으로 탐지하기 위한 많은 연구들이 진행되어 왔다[1-5]. 침입 탐지 방법은 탐지 기술에 따라 크게 오용 탐지와 이상 탐지로 분류되는데, 오용 탐지 기법의 단점은 알려지지 않은 공격을 탐지할 수 없으며, 이상 탐지 기법은

* 숭실대학교 컴퓨터학부(skycolor@ss.ssu.ac.kr)
논문번호 : KICS2009-09-403, 접수일자 : 2009년 9월 11일, 최종논문접수일자 : 2010년 2월 4일

이전에 발생하지 않았던 정상적인 사건을 침입으로 간주한다는 문제가 있다. 특히, 이상 탐지 기법은 오용 탐지 기법에 비해 시스템 구현이 매우 어렵고 False Positive 비율이 높기 때문에 오용 탐지 기법을 보완하는 하이브리드 탐지 방법이 사용되고 있다. 이러한 침입탐지 및 차단 시스템의 성능은 무엇보다도 얼마나 많은 탐지룰을 포함하고 있는가와 새로운 공격에 대한 탐지룰이 얼마나 빠르게 갱신되는가에 달려있다. 새로운 탐지룰 생성을 위해서는 새로운 공격에 대한 정보가 그 만큼 신속하게 수집되고 분석되어야 한다. 이러한 이유로 공격 정보를 신속하고 효과적으로 수집하기 위한 허니팟 관련 연구가 진행되어 왔다[6-12].

허니팟은 악의적인 공격을 일부러 허용하는 컴퓨터 자원으로써 침입이 발생했을 때 그에 대한 정보를 수집하는데 목적이 있다. 하지만 새로운 공격에 대한 탐지룰을 생성하기 위해서는 허니팟에 의해 수집된 정보를 관리자가 적절히 가공해야 침입 탐지 및 차단 시스템에 의해 사용될 수 있기 때문에 새로운 공격에 실시간으로 대응하는 데에는 한계가 있다.

이에 본 논문에서는 알려진 공격에 대한 탐지룰 없이 신종 인터넷 웜 공격으로부터 네트워크상에 존재하는 서버를 보호하는 악성 프로세스 및 실행파일 제거 시스템을 제안한다. 제안 시스템은 인터넷 웜 공격을 탐지하여 공격 정보를 보호 대상 서버에게 전달하는 제어 서버와 제어 서버로부터 수신한 정보를 이용하여 보호 대상 서버에서 생성되는 악성 프로세스와 악성 실행파일을 제거하는 에이전트로 구성된다. 제어 서버는 보호 대상 서버와 동일한 서비스 프로그램들이 실행되는 서버 환경으로써 외부로부터 유입되는 악의적인 패킷에 의해 프로세스가 생성되면 이를 식별하여 공격 근원지 IP 주소와 포트번호를 에이전트에게 전송한다. 에이전트는 이 정보를 기반으로 인터넷 웜에 의해 생성된 보호 대상 서버 상의 악성 프로세스와 실행파일을 제거한다. 제안 시스템은 알려진 공격에 대한 탐지룰을 사용하여 공격을 차단하는 방법이 아니기 때문에 신종 인터넷 웜에 효과적으로 대응할 수 있어 기존의 보안 시스템들과 통합될 경우 보호대상서버의 안전성을 높일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 이해를 돕기 위해 인터넷 웜과 침입탐지 기법, 그리고 허니팟과 관련된 연구들에 대해 기술하며, 3장에서는 악성 프로세스 및 실행파일 제거 시스템에 관해 기술한다. 그리고 4장에서는 제안 시스템을 평가하며, 마지막으로 5장에서는 본 논문의 결론을 맺는다.

Ⅱ. 관련 연구

2.1 인터넷 웜

2003년 인터넷 대란을 일으켰던 slammer웜은 포트 번호 1434를 사용하는 MS-SQL 서버를 공격 대상으로 하는 웜으로, 서버가 종료될 때까지 임의의 목적지 IP 주소를 가지는 패킷을 보내기 때문에 감염 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 네트워크 서비스를 불가능하게 한다. 아울러 2008년에는 MS08-067 취약점을 악용하는 인터넷 웜이 등장하였는데 감염된 사용자 PC에는 인터넷 접속 장애가 발생하며, 공격자는 원격에서 감염된 PC를 통제하는 것이 가능하다. 이 인터넷 웜의 공격은 네트워크 내에서 보안상 취약한 서버들을 찾기 위해 TCP 445번 포트로 스캔 공격을 시작하며, 취약한 서버를 발견한 경우 취약점 공격 코드를 전송한다.

그림 1과 2는 OpenSSL의 보안 취약점을 악용하는 Linux Slapper 웜[13]과 MS08-067 취약점[14]을 악용하는 웜이 보안상 취약한 호스트를 찾는 과정에서 생성되는 패킷 헤더 정보를 요약한 것이다.

두 인터넷 웜이 생성하는 패킷들의 공통점을 살펴

S_IP	D_IP	S_Port	D_Port
.75	1	37778	80
.75	2	37779	80
.75	3	37780	80
.75	8	37785	80

그림 1. OpenSSL 취약점 공격을 위한 스캔 패킷 헤더
Fig. 1. Packet headers for OpenSSL vulnerability scan

S_IP	D_IP	S_Port	D_Port
.75		3745	445
.75		3752	445
.75	7	3741	445
.75	3	3744	445

그림 2. MS08-067 취약점 공격을 위한 스캔 패킷 헤더
Fig. 2. Packet headers for MS08-067 vulnerability scan

보면, 네트워크 내의 많은 호스트들을 공격하기 위해 패킷의 목적지 주소가 계속적으로 변하는 것을 알 수 있으며, 공격 대상이 되는 서비스 프로그램이 구동되고 있는지 확인하기 위해 고정된 포트번호로 패킷이 전달되는 것을 알 수 있다. 인터넷 웜은 스스로 동작해 다른 호스트에 복제되며 전파 속도가 매우 빨라 인터넷 전체에 큰 영향을 줄 수 있기 때문에 신종 인터넷 웜에 의한 피해를 최소화하기 위해서는 인터넷 웜 공격을 신속하게 식별하고 차단하는 것이 매우 중요하다.

2.2 침입 탐지 및 차단 시스템

공격자는 서버에 침입할 때 서버 프로그램에 존재하는 보안 취약점을 악용한다. 취약점을 악용하여 공격에 성공하게 되면 인터넷 웜을 제작하여 빠른 시간 내에 다수의 서버에 침입하기도 한다. 이러한 공격을 탐지하기 위해서는 시스템에 남겨진 로그 파일을 분석할 수 있지만, 관리자가 주기적으로 확인해야 한다는 문제가 있기 때문에 로그 파일 분석을 통한 실시간 침입 탐지는 많은 어려움이 따른다. 이러한 문제를 해결하기 위해 지금까지 침입 탐지에 관한 많은 연구들이 진행되어 왔다[15-17].

침입 탐지 방법은 탐지 기술에 따라 크게 오용 탐지(misuse detection)와 이상 탐지(anomaly detection) 방법으로 구분된다. 오용 탐지 방법은 정해진 모델과 일치하는 행위를 탐지하는 방법을 의미하며, 이상 탐지 방법은 정해진 모델을 벗어나는 비정상적인 행위를 탐지하는 방법을 의미한다. 오용 탐지 방법은 알려진 공격 유형에 관한 정보를 수집 및 분석하여 침입을 탐지하기 위한 룰을 생성하고 공격 패턴과 일치하는 행위가 발생할 경우 이를 침입으로 간주한다. 이상 탐지 방법은 일반적인 시스템 사용 패턴에 벗어나는 행위를 발견했을 때 이를 침입으로 간주한다. 즉, 정상적이고 평균적인 상태를 기준으로 하여, 이에 상대적으로 급격한 변화를 일으키거나 일어날 확률이 낮은 사건이 발생할 경우 이를 침입으로 간주하는 방식이다. 예를 들면, 사용자의 로그인과 로그아웃 시간, 사용자의 프로세서, 메모리 및 디스크 사용률이 기존의 평균적인 상태와 급격히 다를 경우를 비정상적인 사건으로 간주한다.

또한, 침입 탐지 방법은 탐지 시스템의 설치 지점에 따라 네트워크 기반과 호스트 기반 침입탐지 시스템으로 구분된다. 네트워크 기반 침입 탐지 시스템은 네트워크에 지나가는 모든 트래픽을 감시하여 침입을 탐지하는 반면, 호스트 기반 침입 탐지 시스템은 특정 호스트에 설치되어 서버 내부에서 발생하는 사건들을 감시하고 침입을 탐지한다.

Snort[18]는 오픈 소스 기반의 네트워크 침입탐지 및 차단 시스템으로서 미리 정의된 탐지룰을 통해 공격자의 불법행위를 탐지하는 오용 탐지 기법을 이용하기 때문에 알려진 공격에 대한 탐지율이 매우 높다. 또한, 비정상적인 트래픽을 식별하여 공격을 탐지하는 이상 탐지 기법을 통해 스캔공격이나 인터넷 웜들을 탐지하며, 최근에는 IPtables를 통한 침입 차단 기능을 함께 제공하기 때문에 네트워크 보안시스템으로써 유용하게 사용되고 있다. 그림 3은 Snort가 리눅스 쉘 코드 공격과 Linux Slapper 웜 공격을 탐지하기 위해 사용하는 탐지룰의 일부를 나타낸다.

공격 탐지룰을 살펴보면, 임의의 외부 네트워크로부터 서버가 존재하는 네트워크의 특정 포트로 들어오는 패킷에 “90 90 90 e8 c0 ff ff ff|/bin/sh”라는 데이터가 포함되어 있을 경우, 이를 리눅스 쉘 코드 공격으로 간주하며 “SHELLCODE Linux shellcode”라는 경고 메시지를 외부 도록 한다. 네트워크 기반의 침입탐지 시스템에 의해 사용되는 탐지룰은 네트워크를 통해 수집된 패킷 공격 정보하며 반으로 외부되기 때문에 새로운 유형의 공격에 민첩하게 대응하기 위해서는 효과적인 정보 수집 방법이 요구된다.

[3]에서는 멀티캐스트 기반의 인터넷 웜 공격을 탐지하고 탐지룰을 생성하는 시스템을 제안하였다. 이 연구에서는 인터넷 웜의 특성을 이용하여, 짧은 시간 내에 다수의 유입되는 패킷들의 목적지 IP 주소가 모두 다르고 목적지 포트 번호가 모두 같을 경우 인터넷 웜 공격으로 판단한다. 만일 이러한 공격 패턴이 발생하였으나 탐지 시스템 내에 정확한 탐지룰이 존재하지 않으면 일정 시간동안 수집된 패킷 정보들로부터 프로토콜, 포트번호, 데이터를 추출하여 탐지룰을 생성한다. 이 시스템은 신종 웜에 대한 정확한 탐지룰이 생성되기 전에 임시방편적으로 사용될 수 있는 탐지룰을 생성한다는 장점이 있다. 하지만 탐지룰을 생성할 때에 공격 식별에 반드시 필요한 패킷 데이터뿐만 아니라 공격 패킷들에 포함되어 있는 전체 데이터를 단순히 일정 길이만큼 추출하여 생성하기 때문에 탐지룰의 크기가 지나치게 커질 수 있는 단점이 있다.

[1]에서는 변종 인터넷 웜을 탐지하기 위한 시그너처 생성 엔진을 제안하였다. 대부분의 탐지 시스템에

```
$EXTERNAL_NET any -> $HOME_NET $SHELLCODE_PORTS
(msg:"SHELLCODE Linux shellcode"; content:"|90 90 90 e8 c0 ff ff ff|/bin/sh";)
```

그림 3. 침입 탐지를 위한 Snort 탐지룰
Fig. 3. A signature of Snort for intrusion detection

서 정확한 패턴 매칭을 이용하기 위해 LCS(Longest Common Substring) 알고리즘을 사용하는 반면, [1]에서는 연속되지 않은 동일한 패턴을 찾기 위해 LCSeq(Longest Common Subsequence) 알고리즘을 사용하여 변종 웜에 대한 탐지룰을 생성한다. 하지만, LCSeq 알고리즘은 변종 웜을 탐지할 수 있다는 장점은 있으나 LCS 알고리즘에 비해 오탐율이 높아지는 현상이 발생한다.

[4]에서는 기존의 웜 탐지 시스템들이 네트워크의 모든 패킷들을 검사해야 한다는 문제점 제시하고 이를 보완하기 위한 탐지 방법을 제시하였다. 이 연구에서는 네트워크 트래픽량, 트래픽량의 미분값, 트래픽량의 평균 미분값, 트래픽량의 평균 미분값과 평균 트래픽량의 곱에 대한 변이 값을 분석하여 웜 공격을 탐지한다. 이 방법은 모든 패킷을 검사하는 방식과 비교할 때, 많은 CPU와 메모리 사용량이 요구되지 않아 웜 공격을 탐지하기 위한 시간을 감소시킬 수 있다는 장점이 있다.

[5]에서는 탐지룰 없이 웜의 자기 복제 성질을 이용하여 웜을 탐지하는 방법을 제안하였다. 이 연구에서는 웜 공격부터 자기복제 단계까지를 Host Search, File Access, Networking, Memory Injection/infection으로 구분하였다. 제안된 탐지 기법에서는 자기 복제 성질을 이용하기 때문에 실질적인 웜 탐지는 4번째 단계에서 복제 활동이 이루어져야 가능하다. 따라서 웜 공격을 탐지하여 차단할지라도 복제 이전에 시스템 상에 발생하는 공격 행위에 대해서는 아무런 대응도 할 수 없다.

요약하면, 기존의 연구들 대부분은 신종 웜에 대한 탐지 및 탐지룰 생성에 관한 연구들이며 웜에 의한 감염으로 인해 발생하는 문제를 다루고 있지 않다. 또한 호스트 상에서 발생하는 현상들을 통해 웜을 탐지하는 방법들도 웜의 자기 복제 성질을 이용하기 때문에 자기복제 이전에 발생하는 피해를 막을 수 없다는 문제가 있다.

2.3 허니팟

허니팟은 악의적인 공격을 고의적으로 허용하기 위해 구축된 시스템으로서, 침입이 발생했을 때 그에 대한 정보를 수집하는데 목적이 있다. 허니팟은 서비스를 목적으로 제공되는 자원이 아니기 때문에 허니팟으로의 접근은 공격 행위로 간주된다. 따라서 허니팟에서 수집된 정보들은 많은 양이 아니지만, 공격에 사용된 방법과 공격 도구 분석, 취약한 서비스 식별 및 분석에 사용될 수 있어 상당한 가치가 있다.

허니팟은 특정 네트워크 내의 서버를 보호하기 위한 목적으로도 사용된다. 서버와 유사한 환경을 허니팟에 구축함으로써 공격자에 의한 서버 공격 가능성을 감소시키고, 허니팟에서 공격 행위가 이루어지는 동안 서버를 안전하게 구성할 수 있는 시간을 확보할 수 있다.

이러한 허니팟을 이용하여 공격 정보를 효과적으로 수집하기 위한 시스템들이 제안되었는데, [12]에서는 서버에서 사용되고 있지 않은 포트를 유인포트의 목적으로 열어두고 이 포트로 유입되는 모든 패킷을 공격 패킷으로 간주하여 허니팟으로 리다이렉트함으로써 공격자의 정보를 수집한다. 그리고 [10]에서는 사용하지 않는 IP 주소에 접근하는 행위를 공격으로 간주하여 허니팜으로 리다이렉트하는 시스템을 제안하였다. 이 시스템은 네트워크마다 허니팟을 구축할 때 생기는 비용과 관리상의 어려움을 해결하기 위해 각 네트워크에는 단지 프록시만을 설치한 후, 프록시가 사용하지 않는 IP 주소로의 접근을 탐지하여 허니팜으로 리다이렉트하는 통합 시스템이다.

현재 국내외 보안 관련 기관에서는 허니넷을 통해 새로운 유형의 웜, 바이러스 정보를 수집하고 있으며, 이 정보는 침입 탐지 및 차단 시스템에 사용되는 탐지룰을 생성하기 위해 사용될 수 있는 유용한 정보이다. 하지만 탐지룰 생성을 위한 목적으로 허니팟에 의해 수집된 정보를 이용하더라도 분석 및 가공 절차를 거쳐야 하기 때문에 새로운 유형의 공격에 실시간으로 대응하는 데에는 한계가 있다.

Ⅲ. 악성 프로세스 및 실행파일 제거 시스템

본 장에서는 인터넷 웜에 의한 감염으로 발생하는 서버 상의 피해를 줄이기 위한 인터넷 웜 프로세스 및 실행파일 제거 시스템에 관해 기술한다. 본 논문의 연구범위는 인터넷 웜에 의해 감염될 경우 해당 웜에 의해 생성된 악성 프로세스와 악성 실행파일을 제거하여 서버를 보호하는 것으로 제한하며, 인터넷 웜에 감염된 외부 서버로부터 발생하는 DoS 공격에 대한 방어는 연구 범위에서 제외하기로 한다. 아울러 본 연구의 결과물은 완전한 보안 솔루션은 아니며 기존 보안 시스템의 단점을 보완하기 위한 목적으로 제안되었다.

3.1 시스템 구축을 위한 인터넷 웜 분석

인터넷 웜에 의한 공격 절차는 크게 세 부분으로 이루어진다. 첫 번째는 취약한 호스트를 감염시키기 위한 행위이며, 두 번째는 감염된 호스트나 외부 호스

트를 공격하는 행위, 세 번째는 전파를 목적으로 외부 호스트를 감염시키는 행위이다.

앞서 기술했듯이 인터넷 웜은 공격자가 단기간에 많은 호스트를 공격하기 위해 개발된 자동화 프로그램이기 때문에 미리 등록된 IP 주소로 악의적인 패킷을 멀티캐스팅한다는 특징을 갖는다. 인터넷 웜은 호스트를 감염시키기 위해 특정 네트워크상에 존재하는 여러 호스트들의 서비스 프로그램 종류와 버전 정보를 조사하기 위한 패킷을 전송한다. 이 때 인터넷 웜이 공격 조건(서비스 프로그램 종류, 버전 정보)에 만족하는 대상을 찾으면 이들에게 공격 패킷을 전송한다.

인터넷 웜의 감염 과정은 먼저 대상 호스트의 취약점을 악용하여 세션을 수립하고 전파 목적으로 자기 복제본을 전송한다. 전송된 복제본의 실행을 위해서는 공격 호스트가 직접 실행 명령을 전송하기도 하지만, 관리자에 의해 악성 프로세스가 종료되거나 감염된 호스트가 재부팅 되더라도 자동으로 실행되도록 하기 위해 crontab에 스크립트를 생성한다. 복제복이 실행되면 악성 프로세스가 생성되며 감염된 호스트를 공격하거나 외부 호스트로 전파를 위한 공격이 시작된다.

인터넷 웜 공격으로부터 안전한 서버 환경을 구축하기 위해서는 취약점이 없는 서비스 프로그램 개발이 요구되지만 이는 거의 불가능하기 때문에 대부분의 서비스 프로그램들은 내재된 취약점이 존재한다. 따라서 취약한 서비스 프로그램에 대한 보안패치와 보안시스템들에 의해 사용되는 탐지룰이 신속히 개발되어야 한다. 하지만 불행히도 이러한 조치는 취약점 공격이 선행된 이후에 이루어지는 경우가 많다.

3.2 시스템 배치

인터넷 웜이 짧은 시간 내에 특정 네트워크의 많은 호스트를 감염시키기 위해 네트워크에 전체적으로 악의적인 패킷을 전송한다는 점과 허니팟이 보호 대상 서버와 유사한 환경을 구축한다는 점에 착안하여 그림 4와 같이 제안 시스템을 구축한다. 사용자가 특정 서버에 서비스 요청을 할 때에는 서버의 IP 주소나 도메인명을 명시하기 때문에 해당 서버에게 서비스 요청 패킷이 전송된다. 하지만 제어 서버는 서비스를 목적으로 구축된 서버가 아니기 때문에 사용자의 서비스 요청 패킷은 제어 서버로 전달되지 않는다. 하지만 외부로부터 인터넷 웜 공격이 시작되면 웜의 특성 상 공격 패킷이 멀티캐스팅되기 때문에 보호 대상 서버들은 물론 네트워크에 존재하는 제어 서버에게도 전달된다.

악성 프로세스 및 실행파일 제거 시스템은 제어 서버와 에이전트로 구성된다. 에이전트는 보호 대상 서버에 설치되며, 제어 서버의 명령에 따라 악성 프로세스와 실행파일을 제거하는 역할을 한다. 제어 서버에는 보호 대상 서버와 동일한 서비스 프로그램이 구동되며, 서비스 프로그램의 구동에 필요한 시스템 및 설정 파일만이 저장되어 있다.

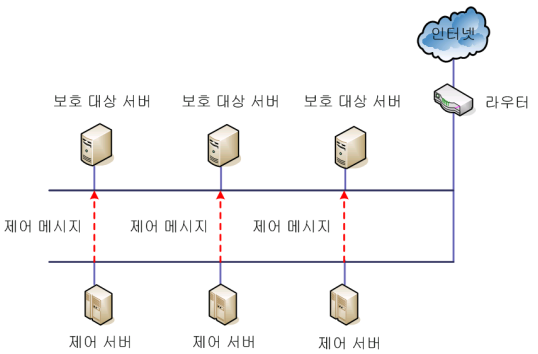

그림 4. 악성 프로세스 및 실행파일 제거 시스템 배치도
Fig. 4. Deployment of a system for removing malicious processes and executable files

보호 대상 서버가 인터넷 웜 공격을 받게 되면 서버 상에 악성 프로세스가 생성되며, 동일한 서비스 프로그램이 구동되고 있는 제어 서버 역시 악성 프로세스가 생성된다. 이 때 제어 서버는 악성 프로세스와 관련된 정보를 에이전트에게 전송하며, 에이전트는 수신된 정보를 통해 악성 프로세스와 실행파일을 탐색하고 제거하는 역할을 한다.

3.3 제어서버와 에이전트

제어 서버에서는 외부로부터 접근이 발생했을 때 이를 식별하고 악성 프로세스의 생성을 탐지한다. 인터넷 웜에 의한 공격이 성공되면 서비스 프로그램과 관련된 프로세스로부터 새로운 프로세스가 생성되는데, 그림 5는 인터넷 웜이 서버에서 실행중인 ssh 서비스 프로그램을 공격하여 악성 프로세스가 생성된 상태를 나타낸다.

인터넷 웜에 의한 공격이 발생하면 네트워크 내에 존재하는 많은 호스트에게 공격 패킷이 전달되기 때문에 동일한 취약점을 갖는 호스트에서도 이러한 악성 프로세스가 생성된다. 이후 이 프로세스로부터 생성되는 모든 악성 프로세스들은 동일한 SID(Session ID)를 부여받기 때문에 인터넷 웜에 의해 생성된 모든 프로세스들을 제거하기 위해서는 같은 SID를 갖는 프로세스들을 탐색하여 제거하면 된다.

그림 6은 보호 대상 서버 상에 생성되는 악성 프로세스와 실행파일을 제거하기 위한 시스템 구조이다.

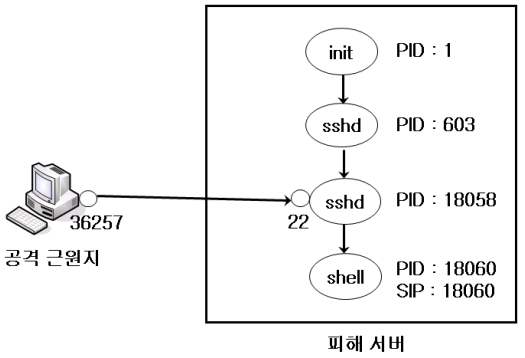

그림 5. 피해 서버상에서의 악성 프로세스 생성
Fig. 5. Creating malicious processes on a victim server

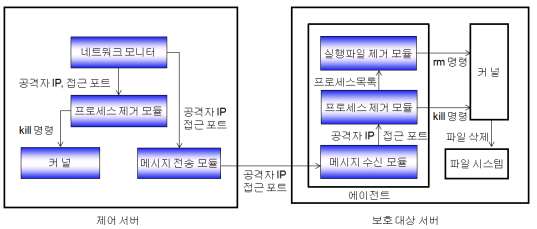

그림 6. 악성 프로세스 및 실행파일 제거 시스템 구조
Fig. 6. Architecture of a system for removing malicious processes and executable files

1) 네트워크 모니터 : 인터넷 웜에 의한 서비스 포트로의 접근을 실시간 감시한다. 제어 서버는 보호 대상 서버와 동일한 포트를 오픈하고 있으며, 네트워크 모니터는 이 포트로의 접근이 발생하면 공격 근원지 IP 주소와 공격 대상이 되는 서비스 포트번호를 프로세스 제거 모듈과 메시지 전송 모듈에게 전달한다.
2) 악성 프로세스 제거 모듈 : 제어 서버와 에이전트에 함께 존재하는 모듈로서, 공격자 IP 주소와 공격 대상이 되는 서비스 포트에 관한 정보를 이용하여 악성 프로세스들 중에서 최상위 프로세스의 SID를 추적한다. 이후, 추적된 SID를 갖는 모든 악성 프로세스들의 PID를 추출하고 커널에게 "kill -9 PID" 명령을 보내어 이들을 제거한다.
3) 악성 프로그램 제거 모듈 : 악성 프로세스를 제거하더라도 인터넷 웜에 의해 설치된 악성 프로그램이 존재하면 Crontab에 의해 악성 프로세스가 재생성된다. 악성 프로그램 제거 모듈은 에이전트에 존재하는 모듈로서, 차후에 악성 프로세스가 재생성되는 것을 막기 위해 현재 실행 중인 악성 프로세스들의 정보를 이용하여 악성 실행파일을 찾은 후 이를 제거한다.
4) 메시지 전송 모듈 : 제어 서버상의 네트워크 모니터로부터 수신한 공격자 IP 주소와 접근 포트 번호를 에이전트에게 전달하여 보호대상서버에 생성된 악성 프로세스 및 실행파일을 찾도록 한다.
5) 메시지 수신 모듈 : 에이전트에 존재하는 모듈로서, 제어 서버로부터 수신한 공격자 IP 주소와 접근 포트 번호를 프로세스 제거 모듈에게 전달한다.

3.4 네트워크 모니터

네트워크 모니터는 libpcap 라이브러리를 이용하여 구현되며, 그림 7은 네트워크 모니터의 동작 코드를 나타낸다.

net_monitor()는 외부로부터 수신되는 패킷을 캡쳐하기 위해 초기화 작업들을 수행한다. 초기화 과정에서 호출되는 pcap_compile() 함수는 네트워크상에 존재하는 모든 패킷들 중에서 특정 패킷만을 감시하기 위해 룰을 컴파일한다. 룰은 pcap_compile() 함수의 두 번째 인자인 rule에 명시되어 있는데, 보호 대상 서버의 서비스 프로그램과 바인딩되어 있는 포트 목록들이 포함된다. 따라서 네트워크 모니터는 전체 트래

```
net_monitor()
{
    ...
    // pcap_open_live()가 사용하게 될 적당한 네트워크 디바이스의 포인터 리턴
    dev = pcap_lookupdev(buf);
    // 네트워크 상의 패킷들을 보기 위한 패킷 캡쳐 디스크립터를 리턴
    ds = pcap_open_live(dev, PROMISCUOUS, 1024, buf);
    // 네트워크 디바이스와 관련된 네트워크 번호와 마스크를 결정하기 위해 사용
    pcap_lookupnet(dev, &net, &mask, buf);
    pcap_compile(ds, &cd, rule, 0, netmask); // 필터 룰을 컴파일하기 위해 사용
    pcap_setfilter(ds, &cd); // 필터를 지정
    pcap_loop(ds, -1, extract_info, 0); // 패킷을 캡쳐할 때마다 monitor 함수를 호출
}
extract_info()
{
    ...
    source_IP = inet_ntoa(ipheader->saddr); // 패킷에서 공격자 IP 주소 추출
    ...
    dest_Port = ntohs(tcpheader->dest); // 패킷에서 서비스 포트번호 추출
    send_info(source_IP, dest_Port); // 프로세스 제거 모듈로 IP주소와 포트번호 전송
}
```

그림 7. 네트워크 모니터에 대한 주요 소스코드
Fig. 7. Source code for network monitor

픽을 감시하는 것이 아니라 보호 대상 서버의 서비스 포트번호로 전송되는 패킷만을 감시한다. pcap_loop() 함수의 세 번째 인자인 extract_info는 패킷이 수신되었을 때 호출될 함수의 포인터이며, extract_info() 함수에서는 패킷에서 공격자 IP 주소와 공격에 사용되는 포트번호를 추출하여 프로세스 제거 모듈에게 전달하는 역할을 한다.

3.5 악성 프로세스 및 프로그램 제거 모듈

인터넷 웜에 감염된 서버 데이터의 훼손과 감염 서버로부터 발생하는 외부로의 공격을 막기 위해서는 서버에 생성된 악성 프로세스와 악성 실행파일을 제거해야 한다. 악성 프로세스를 제거할 때에는 악성 실행파일을 찾기 위한 정보를 미리 저장하여 악성 실행파일 검색에 사용한다.

네트워크 모니터로부터 공격 근원지 IP 주소와 접근 포트 번호를 수신한 프로세스 제거 모듈은 그림 8과 같이 “netstat -anp” 명령을 통해 인터넷 웜과 네트워크 세션을 맺은 서비스 프로그램의 프로세스 ID(18058)를 검색한다.

이후 프로세스 제거 모듈은 그림 9의 (a)와 같이 “ps -axj” 명령을 통해 프로세스 ID가 18058인 프로세스로부터 생성된 자식 프로세스를 탐색하고, 이 프로세스의 SID(18060)을 추출한다. 여기에서 PPID는 부모 프로세스 ID를, SID는 세션 ID를 나타낸다.

마지막으로 프로세스 종료 모듈은 인터넷 웜 공격으로 인해 생성된 모든 프로세스들을 제거하기 위해 SID가 18060인 프로세스들을 탐색한 후, kill 명령을 이용하여 프로세스 종료 시그널을 커널에게 전달한다.

```
Local Address          Foreign Address        PID/Program name
xxx.xxx.xxx.40:22      xxx.xxx.xxx.234:3757   18058/sshd
```

그림 8. netstat 명령을 이용한 프로세스 ID 검색
Fig. 8. Searching process IDs with netstat

```
 PPID    PID    PGID    SID    COMMAND
  603  18058     603    603    /usr/sbin/sshd
18058  18060   18060  18060    /bin/sh
18068  18101   18101  18060    ./mal
```

(a) ps 명령을 이용한 악성 프로세스의 SID 검색

```
COMMAND     PID     NAME
mal       18101     /tmp/mal
mal       18101     /lib/ld-2.7.so
mal       18101     /lib/libc-2.7.so
```

(b) lsof 명령을 이용한 악성 실행파일 검색

그림 9. ps와 lsof를 이용한 악성 프로세스 및 실행파일 검색
Fig. 9. Searching malicious processes and executable files with ps and lsof

인터넷 웜에 의해 생성된 악성 프로세스를 제거한 이후에는 시스템 재부팅 시 악성 프로세스가 재생성되는 것을 막기 위해서 악성 프로세스의 실행파일을 제거한다. 제어 서버의 경우 외부로부터의 접근이 발생할 때 즉시 악성 프로세스를 탐색하여 제거하기 때문에 인터넷 웜에 의한 프로그램 설치가 불가능하다. 하지만 제어 서버가 공격을 탐지하여 보호 서버에게 공격 정보를 생성하고 전송하는 지연 시간 동안 제어 서버에는 악성 프로그램이 설치될 수 있다. 그림 9에서 (b)는 “lsof | grep mal” 명령을 이용하여 mal 악성 프로세스와 관련된 파일의 목록을 검색한 결과이다. NAME 필드의 /tmp/mal은 mal 프로세스의 실행파일 경로이며, /lib/ld-2.7.so와 /lib/libc-2.7.so는 mal 프로그램에 의해 사용되는 동적 라이브러리들의 경로들이다. 이 정보를 통해 보호 서버에 탑재된 에이전트는 악성 프로세스 이름과 동일한 /tmp 디렉토리의 mal 실행파일을 삭제하기 위한 rm 명령을 커널에게 보낸다.

Ⅳ. 시스템 평가

제안 시스템을 평가하기 위한 실험은 그림 10과 같이 OPNET Modeler를 이용한 시뮬레이션 환경에서 수행되었다. OPNET Modeler는 네트워크 모델링 및 시뮬레이션 도구로서 가상 네트워크 구축이 가능하며, 보안 정책을 위한 목적으로 매우 유용하게 사용될 수 있다. 본 실험에서는 네트워크 기반 침입탐지시스템이 구축된 환경과 제안 시스템 및 네트워크 기반 침입탐지시스템이 구축된 환경에서 신종 웜 탐지 능력을 평가한다.

테스트를 위해 구현된 인터넷 웜은 Linux.Slapper 웜과 유사하게 동작한다. Linux.Slapper 웜의 경우에는 C클래스의 162개 고정 IP 주소로 공격을 시도하는

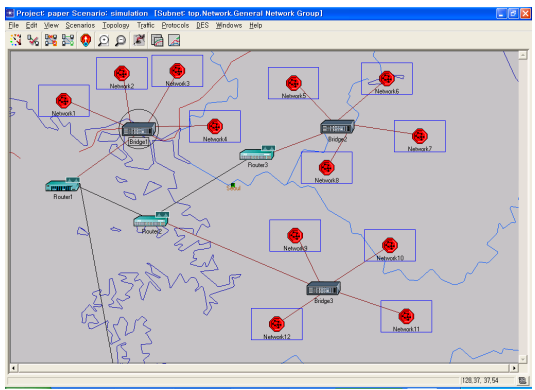

그림 10. 제안 시스템 테스트 환경
Fig. 10. Environment for the proposed system testing

데, 테스트 용 인터넷 웜은 162개의 랜덤한 IP 주소로 공격을 시도하도록 구현하였다. Network1부터 6까지는 제안 시스템과 네트워크 기반 침입탐지시스템이 배치된 네트워크이며, 보호 대상 서버와 제어 서버, 그리고 252대의 클라이언트들로 구성되었다. 그리고 Network7부터 12까지는 제안 시스템이 배치되지 않은 것을 제외하면 Network1과 동일하다.

총 500회의 웜 공격이 발생했을 때 Network1부터 Network6까지 제안 시스템과 네트워크 기반 침입탐지시스템의 웜 공격 탐지 수는 그림 11과 같이 평균적으로 433회였으며, Network7부터 Network12까지 네트워크 기반 침입탐지시스템의 웜 공격 탐지 수는 330회였다. 만일 동일한 환경에서 신종 웜에 의한 500회의 공격이 발생하고, 네트워크 기반 침입탐지시스템 대신 자기복제 성질을 이용하여 신종 웜을 탐지하는 시스템이 배치된 상황이라면, 제안 시스템은 네트워크 상의 신종 웜 탐지 능력을 약 31% 향상시키는 결과를 가져온다.

표 1은 기존 시스템들과 제안 시스템의 장단점을 비교 분석한 결과이다. Snort는 탐지룰을 기반으로 공격을 탐지하고 차단하기 때문에 알려진 인터넷 웜에는 적절히 대응하지만 새로운 유형의 공격에는 대응할 수 없다. 이러한 탐지룰 기반의 침입탐지시스템을 보완하고자 연구 개발된 [1]은 신종 웜 탐지가 불가능하며, [3]은 지나치게 큰 탐지룰을 생성하여 패턴 매칭에 소요되는 시간이 증가한다. 또한 이들은 웜 공격 차단에 대한 기능이 부재하다. [5]는 웜의 자기복제 행위를 감시하여 웜을 탐지하기 때문에 별도의 탐지룰이 요구되지 않는다는 장점이 있다. 이는 신종 웜을 탐지하여 차단할 수 있는 좋은 방법이지만 웜에 의한 자기복제 행위가 이루어지기 전에 발생하는 서버상의 악성 행위를 차단할 수 없다는 문제가 있다. 반면, 제안 시스템은 신종 웜 공격이 발생하면 이를 식별하고

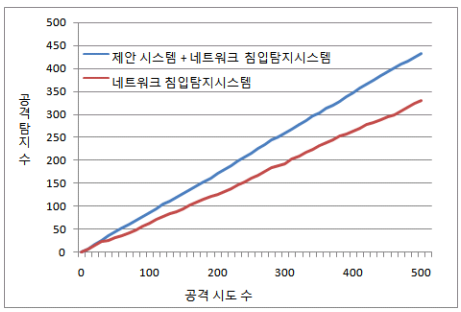

그림 11. 웜 공격 시도에 따른 탐지 횟수
Fig. 11. The number of worm detection for the number of worm attack

표 1. 기존 시스템과 제안 시스템 비교
Table 1. Comparison of legacy system and the proposed system

대응 방법	장점	단점
Snort [18]	알려진 웜 탐지 및 차단 가능	신종 인터넷 웜 탐지 불가능
[1]	변종 웜 탐지 가능	탐지오판율 증가 신종 웜 탐지 불가능 웜 공격 차단 기능 없음
[3]	신종 웜 탐지 가능	탐지룰 크기 증가에 따른 성능 저하 웜 공격 차단 기능 없음
[5]	신종 웜 탐지 가능	웜의 자기복제 이전에 발생하는 악성 행위 차단 불가
제안 시스템	신종 웜 탐지 가능 웜에 의한 악성 프로세스 및 실행파일 제거 가능 제어 서버로의 웜 공격 발생 시, 보호 대상 서버가 공격 받지 않아도 취약점 발견 가능	제어 서버로의 공격이 없으면 탐지 및 차단이 불가능

악성 프로세스와 실행파일을 제거하며, 공격 발생 시 제어 서버에 악성 프로세스가 생성될 때 보호 대상 서버에서 실행중인 서버 프로그램의 새로운 취약점을 바로 발견할 수 있는 장점이 있다. 하지만 제어 서버로의 공격이 발생하지 않을 경우에는 침입 탐지 및 차단이 불가능한 단점이 있다.

V. 결 론

본 논문에서는 인터넷 웜 공격으로부터 네트워크상에 존재하는 서버를 보호하는 악성 프로세스 및 실행파일 제거 시스템을 제안하였다. 제안된 시스템은 탐지룰을 사용하지 않고 웜 공격을 탐지하여 서버를 보호하기 때문에 새로운 공격에 대한 탐지룰이 부재한 환경에서도 신종 인터넷 웜에 대응할 수 있다. 실험에서도 확인했듯이 제안 시스템은 기존의 레거시 침입탐지시스템과 사용될 경우 웜에 의한 공격 탐지율을 증가시킬 수 있다. 하지만 웜에 의한 공격이 제어 서버에 이루어지지 않거나 공격 근원지로부터의 패킷이 라우팅에 의한 문제로 제어 서버에 상당 시간동안 지연되어 도착할 경우에는 늦은 대응으로 인한 피해가 유발될 수 있다.

앞서 살펴본바와 같이 허니팟의 공격 수집 능력을 향상시키기 위해 사용하지 않는 IP 주소로의 접근을

공격 행위로 간주하여 허니팟으로 포워딩하는 여러 연구가 진행되고 있다. 제안 시스템 역시 패킷 포워딩 기법을 이용할 경우, 제어 서버의 IP 주소가 인터넷 웜 공격 대상에 포함되지 않을지라도 공격 패킷을 수신할 가능성을 높일 수 있다. 그리고 제어 서버로 전달되는 공격 패킷이 보호 대상 서버에 비해 현저히 늦게 도착하더라도 사용하지 않는 IP 주소로 전달되는 공격 패킷을 받음으로써 신속히 대응할 수 있는 가능성을 높일 수 있다. 이에 본 연구의 향후 과제는 인터넷 웜 공격에 더욱 효과적으로 대응하기 위해 패킷 포워딩 기법을 접목한 시스템을 구축하는 것이다.

참 고 문 헌

〔1〕 고준상, 김봉한, 이재광, “LCSeq를 이용한 변형 웜 시그너쳐 생성 엔진 구현”, *한국콘텐츠학회논문지*, 제 7권, 제 11호, pp.94-101, 2007.

〔2〕 오진태, 김대원, 김익균, 장종수, 전용희, “고속 정적 분석 방법을 이용한 폴리모픽 웜 탐지”, *한국정보보호학회논문지*, 제19권, 제4호, pp.29-39, 2009.

〔3〕 김익수, 조혁, 김명호, “스캔 기반의 인터넷 웜 공격 탐지 및 탐지룰 생성 시스템 설계 및 구현”, *한국정보처리학회논문지*, 제 12-C권, 제2호, pp.191-200, 2005.

〔4〕 강신헌, 김재현, “네트워크 트래픽 특성 분석을 통한 스캐닝 웜 탐지 기법”, *한국정보과학회논문지*, 제35권, 제6호, pp.474-481, 2008.

〔5〕 황유동, 박동규, 유승엽, 임황빈, 장종수, 오진태, “자기 복제 성질을 이용한 웜 탐지 기법에 대한 연구”, *한국통신학회논문지*, 제34권, 제6호, pp.169-178, 2009.

〔6〕 L. Spitzer, *Honeypots: Tracking Hackers*, Addison-Wesley, 2002.

〔7〕 L. Spitzer, Honeypot Farms, http://www.securityfocus.com/infocus/1720, 2003.

〔8〕 Know Your Enemy: Honeynets, http://old.honeynet.org/papers/honeynet/, 2005.

〔9〕 C. Hoepers, K. Steding-Jessen, L. Cordeiro, and M. Chaves, “A National Early Warning Capability Based on a Network of Distributed Honeypots,” *17th Annual FIRST Conference on Computer Security Incident Handling*, 2005.

〔10〕 X. Jiang, D. Xu, and Y. Wang, “Collapsar: A VM-based Honeyfarm and Reverse Honeyfarm Architecture for Network Attack Capture and Detention,” *Journal of Parallel and Distributed Computing*, Vol.66, No.9, pp.1165-1180, 2006.

〔11〕 H. Artail, H. Safa, M. Sraj, I. Kuwatly, and Z. Al-Masri, “A Hybrid Honeypot Framework for Improving Intrusion Detection Systems in Protecting Organizational Networks,” *Computers & Security*, Vol.25, No.4, pp.274-288, 2006.

〔12〕 김익수, 김명호, “사용되지 않는 포트를 이용하여 해커를 허니팟으로 리다이렉트하는 시스템 설계 및 구현”, *한국정보보호학회논문지*, 제16권, 제5호, pp.15-24, 2006.

〔13〕 http://www.cert.org/advisories/CA-2002-27.html

〔14〕 MS08-067 악성코드 분석, 한국정보보호진흥원, 2008.

〔15〕 U. Lindqvist and P. Porras, “Detecting Computer and Network Misuse through the Production-based Expert System Toolset(P-BEST),” *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pp.146-161, 1999.

〔16〕 S. Stolfo, S. Hershkop, L. Bui, R. Ferster, and K. Wang, “Anomaly Detection in Computer Security and an Application to File System Accesses,” *ISMIS 2005*, pp.14-28, 2005.

〔17〕 C. Michael and A. Ghosh, “Simple, State-based Approaches to Program-based Anomaly Detection,” *ACM Transactions on Information and System Security*, Vol.5, Issue 3, pp.203-237, 2002.

〔18〕 M. Roesch, “Snort - Lightweight Intrusion Detection for Networks,” *Proc. of LISA ’99: 13th Systems Administration Conference*, Nov. 1999.

김 익 수 (Ik-Su Kim) 정회원

2000년 2월 숭실대학교 컴퓨터학부

2002년 2월 숭실대학교 컴퓨터학과 석사

2008년 2월 숭실대학교 컴퓨터학과 박사

2009년 9월~현재 숭실대학교 컴퓨터학부 조교수

<관심분야> 시스템 보안, 네트워크 보안, 모바일 보안, 시스템 소프트웨어