ISSN 1598-0170 (Print)
ISSN 2287-1136 (Online)
http://www.jksii.or.kr

악성 프로세스 제어 시스템의 성능 향상을 위한 보안 프레임워크

Security Framework for Improving the Performance of the Malicious Process Control System

김 익 수[1] 최 종 명[2*]
Iksu Kim Jongmyung Choi

요 약

지금까지 인터넷 웜에 대응하기 위한 다양한 연구가 진행되어 왔다. 대부분의 인터넷 웜 탐지 및 차단 시스템은 탐지룰을 이용하여 인터넷 웜 공격에 대응하지만 새로운 인터넷 웜에 대응할 수 없는 문제가 있다. 이에 인터넷 웜의 멀티캐스트 특징을 이용한 악성 프로세스 제어 시스템이 제안되었다. 하지만 이 시스템은 서비스를 제공해야 할 서버의 수가 많을수록 시스템 구축비용이 증가하고 부분적 공격 유형의 인터넷 웜 공격 탐지 확률이 낮다. 본 논문에서는 악성 프로세스 제어 시스템의 구축비용을 절감하고, 부분적 공격 유형의 인터넷 웜 공격 탐지 확률을 높일 수 있는 보안 프레임워크를 제안한다. 제안된 보안 프레임워크에서는 가상머신을 이용하여 제어서버 구축비용을 줄이며, 사용되지 않는 여분의 IP 주소를 동적으로 인터넷 웜 공격 탐지에 이용함으로써 부분적 공격 유형의 인터넷 웜 공격 탐지 확률을 증가시킬 수 있다. 결국 제안된 보안 프레임워크는 비교적 낮은 비용으로 새로운 유형의 다양한 인터넷 웜에 효과적으로 대응할 수 있다.

주제어 : 인터넷 웜, 침입 탐지, 침입 차단, 악성 프로세스

ABSTRACT

Until now, there have been various studies against Internet worms. Most of intrusion detection and prevention systems against Internet worms use detection rules, but these systems cannot respond to new Internet worms. For this reason, a malicious process control system which uses the fact that Internet worms multicast malicious packets was proposed. However, the greater the number of servers to be protected increases the cost of the malicious process control system, and the probability of detecting Internet worms attacking only some predetermined IP addresses is low. This paper presents a security framework that can reduce the cost of the malicious process control system and increase the probability of detecting Internet worms attacking only some predetermined IP addresses. In the proposed security framework, virtual machines are used to reduce the cost of control servers and unused IP addresses are used to increase the probability of detecting Internet worms attacking only some predetermined IP addresses. Therefore the proposed security framework can effectively respond to a variety of new Internet worms at lower cost.

☞ keyword : Internet Worm, Intrusion Detection, Intrusion Prevention, Malicious Process

1. 서 론

1988년 Morris 인터넷 웜 이후 지금까지 다양한 유형의 인터넷 웜들이 등장해왔다. 인터넷 웜에 의해 발생하는 공격은 대부분 보안상 취약한 프로그램에서 기인한다. 인터넷 웜 공격에 대응하기 위해 많은 보안 시스템들이 개발되어 왔지만 이들은 대부분 알려진 공격 정보를 기반으로 생성된 탐지룰을 이용하여 공격을 탐지하고 차단하기 때문에 공격에 대한 정보를 신속하고 효과적으로 수집할 수 있어야 한다.

현재 공격 정보를 신속하고 효과적으로 수집하기 위해 사용되는 대표적인 보안 시스템으로 허니팟을 들 수 있다 [1]. 허니팟은 의도적으로 공격을 받기 위해 구축된 서버로써 서비스 제공 목적으로 구축된 자원이 아니기 때문에 허니팟으로의 접근은 공격으로 간주될 수 있다. 그러므로 허니팟 관리자는 외부로부터 유입되는 패킷과 허니팟에서 발생하는 프로세스의 활동을 감시하여 공격 정보를 수집할 수 있다. 하지만 공격 정보로부터 탐지룰을 실시간으로 추출해 내는 데는 한계가 있기 때문에 새로운 인터넷 웜에 즉각적으로 대응하는 데에는 어려움이

1 School of Computer Science & Engineering, Soongsil University, Seoul 156-743, Korea
2 Department of Computer Engineering, Mokpo National University, Jeonnam 534-729, Korea
* Corresponding author (jmchoi@mokpo.ac.kr)
[Received 7 November 2012, Reviewed 12 November 2012, Accepted 18 Febryary 2013]

있다. 이에 감염된 서버에서 동작하는 악성 프로세스의 활동을 감시하여 공격을 탐지하는 시스템들이 제안되어 왔지만 자기복제라는 특정 행동에 대해서만 탐지가 가능하다는 문제를 가지고 있다 [2-3].

근본적으로 인터넷 웜 공격으로부터 피해를 막기 위해서는 안전한 프로그램 개발이 요구된다. 안전한 프로그램 개발을 위해 프로그램의 소스코드를 컴파일 할 때 취약점을 탐지하는 정적 분석 도구와 프로그램 실행 시 취약점을 탐지하는 동적 분석 도구들이 개발되어 왔다. 하지만 정적 분석 도구는 단순히 위험한 코드만을 탐지해주기 때문에 안전한 프로그램으로 수정을 하는 것은 전적으로 프로그래머의 능력에 달려 있다. 또한 동적 분석 도구 역시 이를 우회하는 공격들이 등장하여 공격을 완벽히 차단할 수는 없다. 이에 인터넷 웜의 멀티캐스트 특징을 이용한 악성 프로세스 제어 시스템이 제안되었는데, 이 시스템은 취약점이 내재된 프로그램에 공격이 발생했을 때 이를 탐지하고 악성 프로세스 및 프로그램을 삭제하는 능력을 갖추고 있다 [4]. 하지만 이 시스템의 단점은 서비스를 제공해야 할 서버의 수가 증가할수록 악성 프로세스 제어 시스템 구축에 필요한 비용이 크게 증가하고, 전체 IP 주소로 악성 패킷을 전송하는 인터넷 웜에 적절히 동작하지만 IP 주소를 부분적으로 선별하여 공격을 수행하는 인터넷 웜에 대해서는 성능 저하를 보인다.

본 논문에서는 [4]에서 제안된 시스템을 확장하여, 구축비용을 절감하고 IP 주소를 부분적으로 선별하여 공격을 수행하는 인터넷 웜에 더 효과적으로 대응할 수 있는 보안 프레임워크를 제안한다. 제안된 보안 프레임워크에서는 가상머신을 이용하여 다수의 제어서버를 하나의 서버에 통합함으로써 하드웨어 구축비용을 절감할 수 있는 효과를 얻을 수 있다. 그리고 IP 주소를 부분적으로 선별하여 공격을 수행하는 인터넷 웜 공격을 더 효과적으로 탐지하기 위해서 제어 서버에 DHCP 서버를 구축한 후, 네트워크에서 미사용 중인 IP 주소를 인터넷 웜 공격 탐지에 이용함으로써 [4]에서 제안한 시스템 성능을 향상시킬 수 있다. 결과적으로 제안된 보안 프레임워크는 탐지률 없이 낮은 비용으로 새로운 유형의 다양한 인터넷 웜에 효과적으로 대응할 수 있기 때문에 기존의 통합보안관리시스템(Enterprise Security Management)에 적용될 경우 네트워크의 보안성을 크게 향상시킬 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 인터넷 웜과 기존의 보안 시스템들을 소개하고, 3장에서는 제안하는 보안 프레임워크를 기술한다. 그리고 4장에서는 기존 시스템들과 비교 평가하며, 5장에서 결론을 맺는다.

2. 관련 연구

2.1 인터넷 웜

서버 상에서 구동되는 서버 프로그램들은 성능상의 이유로 C/C++ 프로그래밍 언어로 구현되는 경우가 많다. 하지만 C/C++ 프로그래밍 언어에는 보안 취약점을 유발하는 함수들이 존재한다. 대표적인 보안 취약 함수로는 strcpy, strcat, gets, sprintf 등이 있으며, 이 함수들이 포함된 프로그램은 공격자가 버퍼의 경계를 초과하는 값을 입력하여 버퍼 오버플로우를 발생시킴으로써 공격에 성공할 수 있다. 이러한 버퍼 오버플로우 공격을 방지하기 위해 소스코드를 분석하여 버퍼 오버플로우 취약점이 존재하는지의 여부를 판단해주는 도구들도 개발되어왔다 [5-8]. 이러한 도구들은 취약점이 내재된 코드를 사전에 탐지해준다는 장점이 있지만, 안전한 프로그램으로의 자동 변환 기능을 제공하지 않기 때문에 개발자는 안전한 프로그램 개발을 위한 보안 지식을 갖추어야 한다. 이에 프로그램 실행 시 취약한 프로그램에서 발생하는 공격을 차단하는 연구도 진행되어 왔지만 [9], 이를 우회하여 공격을 수행하는 방법들이 등장해왔다 [10].

해커들은 서버 프로그램에서 새로운 취약점을 발견하면 짧은 시간 내에 다수의 서버에 침입하기 위해 인터넷 웜을 개발한다. 인터넷 웜은 네트워크 공격 범위에 따라 네트워크 클래스 A를 공격하는 유형, 클래스 B를 공격하는 유형, 클래스 C를 공격하는 유형으로 구분할 수 있다. IP 주소는 xxx.xxx.xxx.xxx 와 같이 총 4바이트로 구성되는데, 클래스 A를 공격하는 유형은 IP 주소의 1바이트를 상수로 지정하고 나머지 3바이트를 바꾸어가며 서버를 공격하는 형태이다. 마찬가지로 클래스 B와 클래스 C를 공격하는 유형은 각각 IP 주소의 2바이트, 3바이트를 상수로 지정하고 나머지 2바이트와 1바이트를 바꾸어가며 서버를 공격한다. 또한, 클래스 내의 공격 대상 IP 주소를 선택하는 방법에 따라 순차적 공격 유형, 부분적 공격 유형, 임의적 공격 유형의 인터넷 웜으로 구분할 수 있다. 예를 들어, 클래스 C를 공격하는 유형 중에서 순차적 공격 유형에 해당하는 인터넷 웜은 xxx.xxx.xxx.1부터 xxx.xxx.xxx.254에 해당하는 IP 주소로 악성 패킷을 전송한다. 부분적 공격 유형은 인터넷 웜 코드에 미리 설정된 IP 주소만을 공격한다. 마지막으로 임의적 공격 유형은 xxx.xxx.xxx.1부터 xxx.xxx.xxx.254 사이의 IP 주소를 임의로 선택하면서 악성 패킷을 전송한다.

2.2 허니팟

허니팟은 공격자의 정보를 수집하기 위해 사용되는 보안 시스템으로, 설치되는 대상 컴퓨터의 종류에 따라 Client-side 허니팟과 Server-side 허니팟으로 구분될 수 있다. Client-side 허니팟은 클라이언트 컴퓨터에서 실행되면서 인터넷 상의 서버들에 자동으로 접속을 시도한다. 서버 접속 후, 클라이언트 컴퓨터의 레지스트리, 파일 시스템, 프로세스 구조의 허가되지 않은 변화가 발생하는지를 관찰하면서 악의적인 서버들을 식별하고 바이러스나 인터넷 웜과 같은 악성 프로그램의 정보를 수집한다 [11-12]. 최근 Client-side 허니팟을 이용하여 악의적인 서버들로부터 악성 콘텐츠나 악성 파일에 대한 정보를 효과적으로 수집하기 위한 연구가 활발히 진행되고 있다 [13-15].

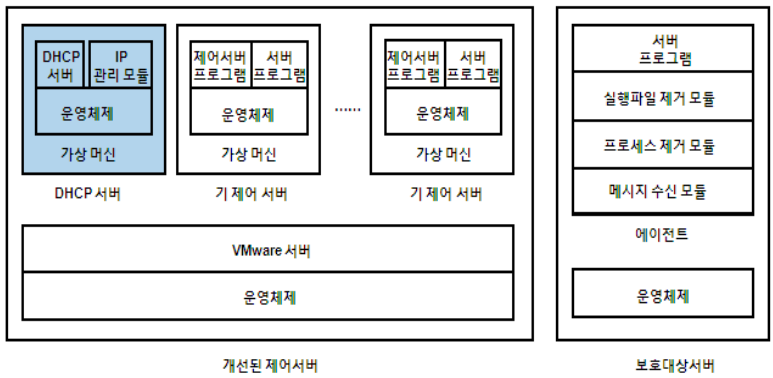
Server-side 허니팟은 서버에서 실행되며, 공격자의 침입을 기다리면서 공격 정보를 수집한다. 즉, Client-side 허니팟은 직접 서버를 접속하면서 공격정보를 수집하는 Active 허니팟인 반면, Server-side 허니팟은 해커의 공격을 기다리면서 공격정보를 수집하는 Passive 허니팟이다. Server-side 허니팟은 특정 서버 프로그램의 취약점이나 운영체제의 취약점 정보들을 수집하기 위해 고의로 구축된 보안 시스템으로, 사용자에게 서비스를 제공하기 위해 구축된 서버가 아니기 때문에 허니팟으로의 모든 접근을 공격으로 간주한다. 허니팟을 통해 관리자는 공격 근원지를 신속히 알아낼 수 있으며, 인터넷 웜에 의해 유입된 공격 패킷이나 허니팟 내의 프로세스 감시를 통해 알려지지 않은 새로운 공격에 대한 정보 수집이 가능하다.

2.3 침입탐지 및 차단시스템

침입 탐지 기법은 이상 탐지 기법과 오용 탐지 기법으로 구분된다. 이상 탐지 기법을 이용한 침입 탐지 시스템은 정상적인 시스템 및 네트워크 사용 패턴을 벗어나는 행위가 발생하는 상황을 공격으로 간주한다. 이상 탐지 기법은 새로운 공격에 대해서도 탐지가 가능하지만 탐지 오판율이 높다는 단점이 있다. [16, 17]는 이상 탐지 기법을 적용한 침입탐지시스템으로써 해커와 인터넷 웜이 보안상 취약한 서버를 찾기 위해 네트워크에 존재하는 서버들로 포트 스캔을 시도한다는 점을 이용하여 침입 탐지를 한다. 포트 스캔은 공격하고자 하는 서버 프로그램이 실행 중인 서버를 찾는 행위인데, 이를 확인하기 위해서 네트워크에 존재하는 다수의 컴퓨터로 동일 패킷을 전송한다. 하나의 컴퓨터가 짧은 시간 내에 다수의 컴퓨터로 패킷을 전송한다는 것은 정상행위로 볼 수 없기 때문에 이를 침입 행위로 간주하는 것이다. 오용 탐지 기법을 이용한 침입 탐지 시스템은 공격 정보를 수집하고 이를 가공하여 생성된 탐지룰을 사용한다. 침입에 대한 판별은 네트워크 트래픽이나 프로세스의 행동을 미리 작성된 탐지룰에 비교하여 패턴이 일치하면 이를 공격으로 간주한다. 하지만 탐지룰을 벗어나는 새로운 공격은 탐지할 수 없는 문제가 있다. Snort는 이상탐지와 오용 탐지 기법을 모두 적용한 대표적인 침입 탐지 시스템으로 침입 차단 기능도 함께 제공한다 [18].

이외에도 시스템 침입에 대해 신속한 탐지 기능을 제공하기 위해 디코이 파일을 이용한 침입 탐지 기법이 제안되었다 [19-21]. [19]에서 제안된 트랩 기반의 방어 메커니즘은 보호되어야 할 컴퓨터상에 자동으로 디코이 문서를 생성하고 저장한다. 디코이 문서는 공격자를 속이기 위한 파일이며, 이 문서는 거짓의 기밀 내용을 포함하고 있다. 만일 공격자가 이 거짓 문서를 열람하거나 외부로 유출하려는 행동을 탐지할 경우, stealthy beacon이라는 센서가 이를 탐지하여 특정 서버로 신호를 보내주어 침입 사실을 알려준다. 하지만 공격자가 침입에 성공한 이후, 디코이 문서를 접근하지 않은 상태에서 발생하는 불법행위에 대해서는 탐지가 불가능하다는 문제가 있다.

[4]에서는 인터넷 웜 공격에 대응하기 위한 악성 프로세스 제어 시스템을 제안하였다. 이 시스템 역시 인터넷 웜이 짧은 시간에 많은 서버를 공격하기 위해 악성 패킷을 멀티캐스트 하는 특징을 이용하였다. 이 시스템의 구성요소인 제어서버는 서비스를 목적으로 배치되는 서버가 아니라 인터넷 웜에 의해 생성된 악성 패킷을 탐지하고 공격 근원지 IP 주소와 공격 대상이 되는 서비스 포트 번호를 보호대상 서버에게 알리는 역할을 한다. 보호대상 서버에 설치된 에이전트는 제어서버로부터 전달받은 IP 주소 및 서비스 포트번호 정보를 토대로 netstat, ps, lsof와 같은 시스템 명령을 통해 악성 프로세스의 프로세스 ID와 실행파일의 경로를 검색한다. 이후 에이전트는 검색된 정보를 통해 kill과 rm명령으로 악성 프로세스와 실행파일을 제거하여 보호대상 서버를 보호한다. 하지만, 2.1절에서 살펴본바와 같이 부분적 공격 유형의 인터넷 웜은 특정 IP 주소를 가진 서버만을 공격하기 때문에, 공격 대상에 제어서버의 IP 주소가 포함되지 않을 경우 보호 대상 서버를 보호할 수 없는 문제가 있다. 그리고 시스템 구축에 소요되는 비용은 보호 대상 서버가 증가할수록 제어서버 구축비용이 크게 증가한다는 문제가 있다.



(그림 1) 제안된 보안 프레임워크
(Figure 1) Proposed security framework

최근에는 해커나 인터넷 웜 공격 정보만을 수집하기 위해 허니팟을 이용하는 것이 아니라 공격에 적극적으로 대응하기 위해 허니팟을 이용하는 연구들이 진행되고 있다 [22-23]. [22]에서는 소규모 네트워크에서 발생하는 인터넷 웜 공격을 탐지하기 위해 허니팟을 이용하며, 허니팟에서 수집한 정보를 통해 인터넷 웜에 실시간으로 대응할 수 있는 프레임워크를 제시하였다. 이 프레임워크에서는 네트워크상에 존재하는 클라이언트 컴퓨터에 에이전트들이 설치되는데, 이 에이전트들은 인터넷 웜을 유인하기 위한 디코이 포트를 생성한다. 인터넷 웜이 디코이 포트에 접근하게 되면 의심스러운 행위로 판단되어 허니팟으로 모든 트래픽이 포워딩되기 때문에 인터넷 웜에 의한 악성 행위들은 허니팟에서 모니터링이 가능하다. 만일 인터넷 웜이 공격에 성공하게 되면 허니팟은 보호 대상 서버에 설치된 에이전트에게 공격 정보를 전달하며, 에이전트는 수신된 정보를 통해 인터넷 웜을 제거하게 된다. 하지만 인터넷 웜을 유인하기 위해 모든 클라이언트 컴퓨터에 에이전트들이 설치되어야 한다는 부담이 따르게 된다.

3. 인터넷 웜 공격 대응 보안 프레임워크

본 장에서는 2.3절에서 소개한 악성 프로세스 제어 시스템 구축비용을 절감하고 성능을 향상시키기 위한 보안 프레임워크에 관해 기술한다. 본 논문에서는 임의적 공격 유형과 부분적 공격 유형의 인터넷 웜 공격에 있어 인터넷 웜의 공격 대상을 예측할 수 없기 때문에 다음과 같은 가정을 한다. 첫째, 임의적 공격 유형의 인터넷 웜은 공격 대상 IP 주소를 생성할 때에 모든 IP 주소가 동일한 확률로 생성된다고 가정한다. 즉, C 클래스 네트워크를 공격하는 임의적 공격 유형의 인터넷 웜이 254번의 서버 공격을 시도할 때에 xxx.xxx.xxx.1부터 xxx.xxx.xxx.254 사이의 모든 IP 주소들을 한 번씩 공격한다고 가정한다. 둘째, 모든 부분적 공격 유형의 인터넷 웜들에 미리 설정된 IP 주소들은 동일한 확률로 코드에 출현한다고 가정한다. 예를 들면, 부분적 공격 유형의 인터넷 웜 전체 집합을 S라하고 이 집합이 a, b, c 원소로 구성된다고 하자. 이 때, 인터넷 웜 a, b, c에 의해 공격이 수행되면 xxx.xxx.xxx.1부터 xxx.xxx.xxx.254 사이의 모든 IP 주소들은 동일한 횟수로 공격을 받는다고 가정한다. 마지막으로 본 논문에서는 보호대상 서버가 클래스 C 네트워크에 존재한다고 가정하여 보안 프레임워크가 기술된다.

3.1 기존연구의 문제점

2.3절에서 소개한 악성 프로세스 제어 시스템에 관한 논문에서 제안된 시스템 배치도에 따라 네트워크가 구성될 경우에는 인터넷 웜 공격 탐지 성능과 시스템 구축비용 상의 문제가 있다. 먼저 구축비용 관점에서 고려해보면, 제어서버의 수는 보호 대상 서버의 수와 동일해야 하기 때문에 보호 대상 서버 수가 증가하면 제어서버의 수도 증가해야 한다. 따라서 보호 대상 서버의 수가 증가할 때 마다 제어서버 구축에 필요한 비용이 크게 증가할 수 있으며, 물리적으로 분산되어 있는 제어 서버들을 개별적으로 관리하는데 어려움이 발생한다.

악성 프로세스 제어 시스템의 성능을 고려해보면, 인터넷 웜 공격이 발생했을 때의 탐지 능력은 제어서버의

수에 비례한다. 기 제안된 시스템 배치도에서는 각각의 제어서버에 하나의 IP 주소가 할당되며, 제어서버의 수는 보호대상 서버의 수와 같다. 순차적 공격 유형의 인터넷 웜 공격이 시작되면, 네트워크에 존재하는 모든 IP 주소로 악성 패킷이 전송되기 때문에 제어서버는 이를 탐지할 수 있다. 그리고 임의적 공격 유형의 인터넷 웜 공격이 발생하는 경우, 임의의 IP 주소 선택이 동등한 확률로 이루어진다고 가정하여 모든 IP 주소로 공격이 이루어지기 때문에 제어서버는 공격을 탐지할 수 있다. 반면, 부분적 공격 유형의 인터넷 웜 공격이 발생하는 경우에는 인터넷 웜 코드에 미리 설정된 일부 IP 주소만을 공격하기 때문에 공격 대상 IP 주소 목록에 제어서버의 IP 주소가 포함되지 않을 가능성이 있다. 이 경우에는 보호대상 서버가 공격을 받더라도 이를 탐지하지 못하기 때문에 보호대상 서버에 생성된 악성 프로세스와 프로그램을 제거할 수 없다.

3.2 제안된 보안 프레임워크와 시스템 배치

3.2.1 제안된 보안 프레임워크

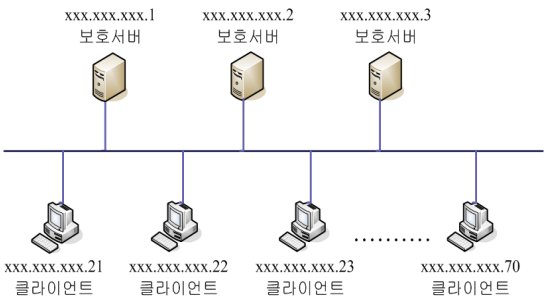
3.1절에서 살펴본바와 같이 기존 시스템은 보호 대상 서버의 수가 증가할 경우, 그에 따른 제어서버 구축비용이 크게 증가한다는 문제점이 있다. 이를 해결하기 위해 본 논문에서 제안하는 보안 프레임워크는 그림 1과 같다. 하나의 물리 서버에 다수의 가상머신이 설치되고 각각의 가상머신에 제어서버가 구축되어 운용되기 때문에 다수의 물리 서버 구축에 소요되는 비용이 크게 절감될 수 있다. 가상머신에는 보호 대상 서버들과 동일한 운영체제와 서버 프로그램들이 설치된다. 그리고 제어서버는 실제로 사용자에게 서비스를 제공하기 위한 목적이 아니라 공격을 탐지하기 위한 목적의 서버이기 때문에 보호 대상 서버에 저장되는 중요 파일이나 DB 데이터들은 제공되지 않으며, 단지 서버 프로그램들에 필요한 시스템 및 설정 파일만이 저장된다.

현재 네트워크에서 사용 중이지 않은 IP 주소로 유입되는 패킷은 정상적인 서비스 요청이 아닌 공격에 의해 발생하는 패킷의 가능성이 매우 높다. 따라서 사용 중이지 않은 IP 주소로의 접근을 제어서버로 리다이렉트 할 경우, 부분적 공격 유형의 인터넷 웜을 탐지할 가능성을 높일 수 있다. 이에 개선된 제어서버 내의 DHCP 서버와 IP 관리 모듈은 인터넷 웜 공격 탐지 성능을 높이기 위해 현재 사용 중이지 않은 IP 주소를 리다이렉터에게 할당하는 역할을 한다. 이후 리다이렉터에 할당된 IP 주소로의 모든 접근은 제어서버로 리다이렉트되기 때문에 인터넷 웜 공격을 효과적으로 탐지할 수 있다. 제어서버는 인터넷 웜 공격을 탐지하면 이에 대한 정보를 보호대상 서버 내에 존재하는 에이전트에게 전달하고 에이전트는 전달된 정보를 기반으로 악성 프로세스와 프로그램을 제거한다.

3.2.2 보안 프레임워크가 적용된 시스템 배치

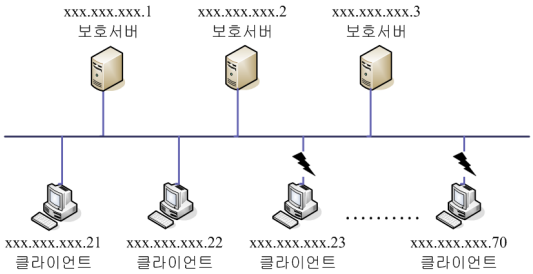
서버 컴퓨터는 항상 클라이언트에게 서비스를 제공해야 하기 때문에 24시간 내내 인터넷에 연결되어야 한다. 따라서 고정된 IP 주소가 24시간 내내 서버 컴퓨터에 할당되어야 한다. 반면, 클라이언트 컴퓨터는 서버 컴퓨터와는 달리 인터넷을 사용하는 경우에만 IP 주소를 할당받아 인터넷에 연결되면 된다. 예를 들면, 회사의 경우 직원들이 하루 업무를 끝내고 퇴근할 때 컴퓨터를 종료하면 그 이후부터는 IP 주소가 필요 없다. 또 다른 예로, 캠퍼스내의 모바일 서비스를 대학생들에게 제공하기 위해 사용되는 IP 주소 역시 일정시간 동안만 할당해주면 된다. 기존의 악성 프로세스 제어 시스템은 부분적 공격 유형의 인터넷 웜 공격이 발생했을 때 제어서버가 이를 탐지할 가능성이 낮다는 문제가 있었다.

고정 IP 주소가 24시간 내내 할당되어야 하는 컴퓨터는 보호대상 서버와 제어 서버, 리다이렉터, 일부 특수 목적의 컴퓨터들이며, 이들에 할당된 IP 주소를 제외한 나머지 IP 주소들은 클라이언트 컴퓨터에 할당 가능한 IP 주소이다. 클라이언트에게 나머지 IP 주소를 모두 할당한 이후에 남은 여분의 IP 주소는 인터넷 웜 공격을 탐지하기 위한 목적으로 사용될 수 있다. 인터넷 웜은 빠른 전파를 목적으로 악성 패킷을 네트워크 내에 멀티캐스트한다. 따라서 현재 사용 중이지 않은 IP 주소로 유입되는 패킷을 인터넷 웜 공격에 의한 패킷으로 간주할 수 있다. 여분의 IP 주소를 관리자가 리다이렉터라는 프록시 서버에 등록해두고, 이후 이 IP 주소로 유입되는 모든 악성 패킷들을 라다이렉터에 의해 제어서버로 전달함으로써 부분적 공격 유형의 인터넷 웜 탐지 성능을 높일 수 있다. 하지만 클라이언트의 IP 주소 사용은 일시적이기 때문에 IP 주소 사용이 종료되었을 때 관리자가 이를 매번 확인하여 리다이렉터에 등록하는 데에는 많은 어려움이 있다. 좀 더 상세히 기술하면, 네트워크에 (그림 2)와 같이 서버와 클라이언트가 배치된 상황을 고려해보자.



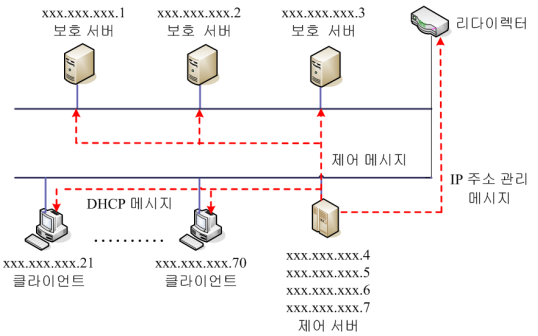
(그림 2) 50대의 클라이언트와 3대의 서버가 인터넷에 연결된 상태
(Figure 2) 50 clients and 3 servers using internet

네트워크에는 현재 클라이언트 컴퓨터 50대가 배치되어 있으며 인터넷 사용을 위해 50개의 IP 주소가 할당되어 있다. 그리고 3대의 서버 컴퓨터가 배치되어 3개의 IP 주소가 할당되어 있다. 클래스 C 네트워크에서는 네트워크 식별에 사용되는 0과 브로드캐스트에 사용되는 255번을 제외한 1번부터 254번까지의 IP 주소가 컴퓨터에 할당 가능하다. 그림 2에서는 현재 53개의 IP 주소가 사용되고 있으므로 관리자는 53개를 제외한 나머지 IP 주소를 리다이렉터에 등록하여 이 IP 주소로 유입되는 악성 패킷을 제어서버로 리다이렉트 할 수 있다.



(그림 3) 48대의 클라이언트가 인터넷으로부터 끊긴 상태
(Figure 3) 48 clients not using internet

하지만, 그림 3과 같이 IP 주소가 xxx.xxx.xxx.23부터 xxx.xxx.xxx.70인 컴퓨터가 종료되어 해당 IP 주소가 사용되지 않는 상황이 발생한다. 이 경우 클라이언트 컴퓨터가 종료되어 IP 주소를 사용하고 있지 않더라도 처음에 리다이렉터에 등록된 IP 주소로 유입되는 악성 패킷만을 제어서버로 리다이렉트 할 수 있다. 만일 현재 클라이언트가 사용 중이지 않은 IP 주소로 유입되는 악성패킷을 제어서버로 리다이렉트하기 위해서는 리다이렉터에 현재 사용 중이지 않은 IP 주소를 매번 확인하여 등록해야 하는 어려움이 발생한다. 이와 반대로, 클라이언트가 다시 인터넷 연결을 위해 IP 주소가 필요한 경우, 리다이렉터에 등록된 IP 주소를 삭제하고 클라이언트에게 할당해 주어야하는 문제가 발생한다.



(그림 4) 보안 프레임워크가 적용된 시스템 배치도
(Figure 4) Deployment of system adopting security framework

그림 4는 기존의 제어서버에 DHCP 서버를 설치하여 인터넷 웜 공격 탐지 성능을 향상 시킨 시스템 배치도를 나타낸다. 현재 사용하지 않는 IP 주소를 이용하여 제어서버의 성능을 향상시키기 위해 제어서버에는 추가적으로 DHCP 서버 프로그램이 설치되며 IP 주소 관리를 위한 IP 관리 모듈이 구현된다. DHCP 서버 프로그램은 IP 관리 모듈과 함께 제어서버에서 동작하면서 현재 사용중이지 않은 IP 주소를 리다이렉터에 등록하고, 클라이언트에 IP 주소를 할당해 주어야 하는 상황이 발생하면 이를 삭제하고 클라이언트에게 배정하는 역할을 한다.

3.2.3 동적 IP 주소 관리를 위한 개선된 제어서버

서버와 같이 24시간 운용되면서 인터넷과 연결되어야 하는 컴퓨터들은 고정된 IP 주소가 항상 할당되어야 한다. 하지만 대부분의 클라이언트 컴퓨터는 서비스를 제공하기 위한 컴퓨터가 아니라 서비스를 이용하는 컴퓨터이기 때문에 필요한 경우에만 동적으로 IP 주소가 할당되면 된다. 처음 보안 프레임워크를 구축하기 위해서는 고정된 IP 주소들을 서버에 할당하고 나머지 IP 주소들은 DHCP 서버가 관리하도록 한다. 그림 5는 DHCP 서버의 dhcpd.conf 파일을 보여준다.

```
subnet xxx.xxx.xxx.0 netmask 255.255.255.0 {
option routers          xxx.xxx.xxx.254;
option subnet-mask      255.255.255.0;
range dynamic-bootp     xxx.xxx.xxx.21  xxx.xxx.xxx.200;
option domain-name-servers      xxx.xxx.xxx.1;
default-lease-time      21600;
max-lease-time          43200;
}
```

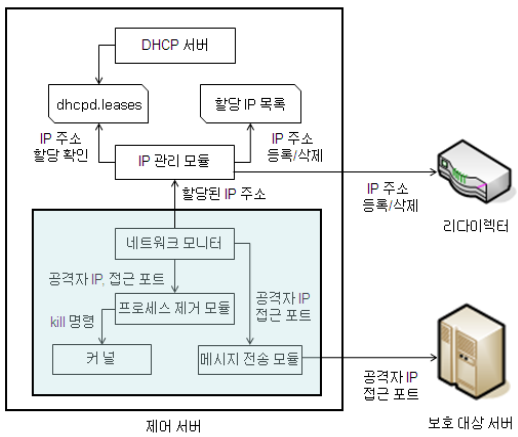
(그림 5) dhcpd.conf 파일 구성
(Figure 5) Example of dhcpd.conf file

클라이언트에게 동적으로 IP 주소를 할당해주기 위한 목적으로 range dynamic-bootp 속성에 xxx.xxx.xxx.21부터 xxx.xxx.xxx.200의 값을 배정하였다. 따라서 클라이언트가 인터넷 사용을 위해 IP 주소 요청을 하면 DHCP 서버는 range dynamic-bootp 속성에 배정된 IP 주소 중 현재 사용 가능한 IP 주소 하나를 클라이언트에게 할당해준다. DHCP 서버가 클라이언트에게 동적 IP 주소를 할당하는 과정은 다음과 같이 4번의 메시지 교환을 통해 이루어진다.

1) DHCPDISCOVER : 클라이언트가 DHCP 서버를 찾기 위해 DHCPDISCOVER 메시지를 브로드캐스트
2) DHCPOFFER : DHCP 서버가 이를 수신하고 사용 가능한 IP 주소가 담긴 DHCPOFFER 메시지를 브로드캐스트
3) DHCPREQUEST : 클라이언트가 DHCP 서버로부터 받은 사용 가능한 IP 주소와 DHCP 서버의 주소가 담긴 DHCPREQUEST 메시지를 브로드캐스트 하여 IP 주소 할당 요청
4) DHCPACK : DHCP 서버가 IP 주소 임대기간, DNS, 디폴트 게이트웨이 등의 값이 담긴 DHCPACK 메시지를 브로드캐스트 하여 클라이언트의 IP 주소 할당을 허락

DHCP 서버가 제공하는 IP 주소는 임대시간 (lease time)이 있기 때문에 클라이언트가 계속 IP 주소를 사용하고자 할 경우에는 임대시간이 끝나기 전에 DHCP 서버로 IP 주소 연장 요청을 보낸다. 만일 이러한 요청이 없는 경우에는 IP 주소 사용 시간이 만료되고 그 IP 주소는 클라이언트로부터 DHCP 서버에게 반납된다. 갱신요청은 다음과 같이 2번의 메시지 교환을 통해 이루어진다.

1) DHCPREQUEST : 클라이언트가 DHCP 서버에게 DHCPREQUEST 메시지를 전송하여 IP 주소 연장 신청
2) DHCPACK : DHCP 서버가 클라이언트에게 새로운 임대시간이 설정된 DHCPACK 메시지를 전송



(그림 6) 기능이 향상된 제어 서버
(Figure 6) Enhanced control server

그림 6은 인터넷 웜 공격 탐지에 사용될 IP 주소를 관리하기 위한 제어서버 구조이며, 그림 7은 제어서버 에 설치된 네트워크 모니터와 IP 관리(추가/삭제) 모듈 내의 주요코드를 나타낸다. DHCP 서버가 클라이언트에게 IP 주소를 할당할 때에 DHCPACK 메시지가 전달되는데, 이 메시지는 UDP 프로토콜을 사용하며 클라이언트의 68번 포트로 전달된다. 또한, 클라이언트의 IP 주소 요청에 따른 서버의 허가 메시지이므로 DHCPACK 패킷의 opcode 필드는 2라는 값을 가진다. 그러므로 네트워크 모니터는 네트워크에 지나가는 패킷의 헤더정보 중 프로토콜이 UDP이며, 목적지 포트번호가 68, opcode가 2인 값을 가진 메시지를 감지하면 이를 IP 관리 모듈에게 알린다. 이 때 IP 관리(추가) 모듈은 dhcpd.leases 파일을 읽어 클라이언트에게 새로 할당된 IP 주소와 임대시각을 할당 IP 목록 파일에 기록한다. 그리고 리다이렉터에 등록되어 있던 IP 주소를 제거하기 위한 메시지를 리다이렉터에게 전송한다. 메시지를 수신한 리다이렉터는 해당 IP 주소를 리다이렉트 목록에서 삭제한다.

클라이언트에게 할당된 IP 주소의 임대시간이 만료되는 경우, 리다이렉터는 이 IP 주소로 유입되는 패킷을 제어서버로 리다이렉트 해야 한다. IP 관리(삭제) 모듈은 crontab에 의해서 주기적으로 할당 IP 목록 파일을 읽어 임대시간이 만료된 IP 주소를 삭제하고 리다이렉터에게 해당 IP 주소에 대한 등록 메시지를 전송한다. 이 메시지

```
network_monitor() {
    ------
    // DHCPACK 메시지를 감지하면
    if((protocol == UDP) && des_port == 68 && opcode == 2)
      // 클라이언트에게 IP가 할당됐음을 IP 관리 모듈에게 알림
      inform_alloc();
    ------
}
ip_add_module() {
    ------
    // dhcpd.leases 파일을 열어 IP가 할당된 클라이언트의 임대시각을 추출
    lease_time = read_leases(ip);
    // 클라이언트 IP 주소와 임대시각을 할당 IP 목록 파일에 기록
    write_info(ip, lease_time);
    // 리다이렉터에 삭제할 클라이언트 IP 전송
    send_remove_message(ip);
    ------
}
ip_remove_module() {
    ------
    // 임대시간이 만료된 IP를 IP 목록 파일로부터 삭제
    remove_ip();
    // 리다이렉터에 추가할 클라이언트 IP 전송
    send_add_message(ip);
    ------
}
```

(그림 7) 네트워크 모니터와 IP 관리 모듈 주요코드
(Figure 7) Code for network monitor and IP management

를 수신한 리다이렉터는 IP 주소를 등록한 후, 이후에 유입되는 패킷들을 제어서버로 리다이렉트 한다.

4. 성능 평가

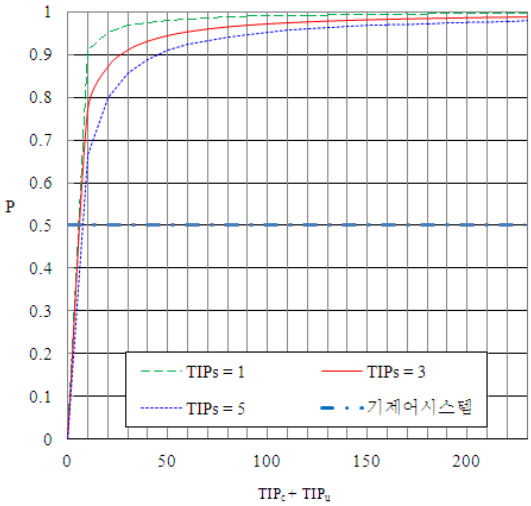
[4]에서 제안한 시스템 배치도에 따라 구성된 네트워크에서는 제어서버가 보호대상 서버와 동일한 개수로 배치된다. 순차적 공격 및 임의적 공격 유형의 인터넷 웜 공격이 발생했을 때에는 모든 IP 주소로 악성 패킷이 전달되기 때문에 보호대상 서버가 공격을 받을 경우 제어서버도 공격을 받게 된다. 따라서 보호대상 서버 내부에서 악성 프로세스가 생성되면, 동일한 시스템 구성을 갖춘 제어서버 내부에서도 악성 프로세스가 생성되기 때문에 제어서버는 이를 탐지하여 보호대상서버에게 악성 프로세스 제거 명령을 전송할 수 있다. 반면, 부분적 공격 유형의 인터넷 웜은 미리 결정된 일부 IP 주소로만 악성 패킷을 전송하기 때문에 보호대상 서버가 공격을 받더라도 제어서버가 공격을 받지 못하여 보호대상 서버로의 공격을 탐지하지 못할 수 있다.

$$P = \frac{TIP_c}{TIP_s + TIP_c} \quad (1)$$

부분적 공격이 발생할 때 보호대상서버와 제어서버가 공격 받을 가능성이 같다는 가정 하에, 제어서버가 보호대상서버로의 공격을 탐지할 확률은 수식 (1)과 같다. 수식 (1)에서 TIP_c는 제어서버에 할당된 총 IP 주소 개수이며, TIP_s 는 보호대상서버에 할당된 총 IP 주소 개수이다. [1]에서 제안한 시스템 배치도에 따르면 제어서버와 보호대상서버의 수가 동일하기 때문에 수식 (1)에서의 P 값은 0.5의 확률 값이 나온다. 수식 (2)는 제안 프레임워크가 적용된 네트워크에서 부분적 공격 유형의 인터넷 웜이 보호대상 서버를 공격하였을 때 제어서버가 이를 탐지할 확률을 나타낸다. 수식 (2)에서 TIP_u 는 DHCP 서버에서 관리하는 동적 IP 주소 중 클라이언트에 할당되지 않은 총 IP 개수를 나타낸다.

$$P = \frac{TIP_c + TIP_u}{TIP_s + TIP_c + TIP_u} \quad (2)$$

그림 8은 [4]에서 제안한 시스템의 탐지 확률과 수식 (2)에 따른 탐지 확률을 그래프로 표현한 것이다. [4]에서 제안한 방식에서는 부분적 공격 유형의 인터넷 웜이 보호대상 서버를 공격하였을 때 제어서버가 이를 탐지할 확률이 0.5의 고정된 확률을 갖지만 본 논문에서 제안하는 보안 프레임워크가 적용된 시스템은 현재 사용되고 있지 않은 IP 주소를 침입탐지에 이용하기 때문에 공격 탐지 확률을 증가시킬 수 있다.



(그림 8) 기 시스템과 제안 프레임워크 탐지 확률
(Figure 8) Probability of detecting internet worm

(표 1) 기 시스템과 제안된 보안 프레임워크 비교
(Table 1) Comparison between the proposed security framework and previous work

대응방법	장점	단점
[2, 3]	신종 인터넷 웜 탐지	인터넷 웜의 자기복제 이전에 발생하는 공격 행위의 탐지 및 차단 불가
[16, 17]	인터넷 웜 침입 탐지	인터넷 웜 차단 불가
[19-21]	시스템 분석을 통한 침입 탐지	디코이 파일 접근 이전에 발생하는 공격 행위의 탐지 및 차단 불가
[4]	신종 인터넷 웜 탐지 및 악성 프로세스와 프로그램 제거 가능	부분적 공격 유형의 인터넷 웜에 대한 낮은 공격 탐지 확률과 높은 구축비용 소요
제안된 보안 프레임워크	제어서버의 높은 공격 탐지 확률과 낮은 구축비용 소요	여전히 제어서버로의 공격이 발생하지 않을 가능성 존재

네트워크 내에 보호대상 서버의 수가 1개인 경우에는 약 10개의 IP 주소를 공격 탐지에 사용할 때 0.9 이상의 공격 탐지 확률을 확보할 수 있다. 그리고 보호대상 서버의 수가 3개, 5개인 경우에는 각각 약 30개와 50개의 IP 주소를 이용하여 0.9 이상의 공격 탐지 확률을 확보할 수 있다. 이메일, 홈페이지, 데이터베이스 등의 여러 서비스를 제공하기 위해 서버를 구축할 때에 단일 서버에 모든 서비스를 통합하는 경우와 각각의 서비스마다 별도의 서버로 구축할 수 있다. 단일 서버로 구축할 경우에는 단지 10개의 IP 주소만을 이용하여 0.9 이상의 탐지 확률을 확보할 수 있지만 해커에 의한 공격으로 인한 서버 탈취로 전체 서비스가 마비될 수 있다. 따라서 가용한 IP 주소의 수가 충분히 확보될 수 있는 경우에는 각각의 서비스를 별도의 서버가 제공할 수 있도록 구축한 후 제안하는 보안 프레임워크를 적용시키는 것이 안전하다.

표 1에서는 기존 보안 시스템들과 제안 프레임워크가 적용된 악성 프로세스 시스템의 장단점을 비교한 결과를 보여준다. 기존 보안 시스템들은 인터넷 웜 탐지만을 목적으로 하거나 차단 기능을 제공하더라도 시스템 내부에서 의심스런 행위가 발생하기 전까지는 인터넷 웜을 탐지할 수 없으며, 탐지 이후에도 이에 대한 대응 방법을 제시하지 않았다. 이러한 문제를 보완하고자 [4]에서는 인터넷 웜 공격이 발생했을 때 생성되는 악성 프로세스와 프로그램을 제거하는 시스템을 제안하였다. 이 시스템은 순차적 공격 유형과 임의적 공격 유형의 인터넷 웜 공격이 발생했을 때는 효과적으로 대응하지만 부분적 공격 유형의 인터넷 웜 공격이 발생하는 경우 제어서버의 공격 탐지 확률이 낮다는 문제가 있다. 반면, 본 논문에서 제안한 보안 프레임워크가 적용될 경우에는 사용하지 않는 여분의 IP 주소를 이용하여 공격 탐지에 이용되기 때문에 그림 8에서 보인 결과와 같이 공격 탐지 확률을 높일 수 있다는 장점을 가지고 있다.

5. 결 론

대부분의 네트워크 기반의 침입탐지 및 차단 시스템은 공격 정보를 기반으로 작성된 탐지룰을 사용하기 때문에 새로운 인터넷 웜 공격이 발생하면 이에 대응할 수 없으며, 시스템 기반의 침입탐지 및 차단 시스템의 경우에도 인터넷 웜 공격이 성공한 후 자기복제와 같은 이상적인 행위가 발생하지 않는다면 탐지 시간이 늦춰질 수 밖에 없다는 문제가 있다. 이에 [4]에서는 인터넷 웜 공격에 대응하기 위한 새로운 방식의 악성 프로세스 제어 시스템을 제안하였지만, 비용과 성능상의 두 가지 문제점이 있었다.

본 논문에서는 이를 보완하기 위해 [4]에서 제안된 악성 프로세스 제어 시스템을 확장한 보안 프레임워크를 제안하였다. 제안된 프레임워크에서는 가상머신을 이용하여 하나의 물리서버에 다중 제어서버를 구축함으로써 보호대상 서버의 증가에 따른 제어서버 구축에 필요한 하드웨어 비용을 절감할 수 있다. 또한, 부분적 공격 유형의 인터넷 웜 공격 발생 시, 제어서버의 낮은 공격 탐지 확률을 높이기 위해 사용 가능한 여분의 IP 주소를 이용함으로써 성능을 향상시킬 수 있다. 악성 프로세스 제어 시스템과 마찬가지로 본 연구는 제어서버로의 공격이 발생하지 않을 가능성이 여전히 존재하지만, 이전 시스템과 비교할 때 부분적 공격 유형의 인터넷 웜 공격 탐지 확률을 더욱 향상시켰으며 시스템 구축에 소요되는 비용을 절감시킬 수 있다는 데에 연구 가치가 있다.

참 고 문 헌(Reference)

[1] Spitzer, L., Honeypots: Tracking Hackers, Addison-Wesley, 2002.

[2] Hwang, Y,. Park, D,. Yoo, S., Yim, H., Jang, J., and Oh, J., “A study of the worm detection method using self-replication,” The Journal of Korea Information and Communications Society, vol.34, No.6, pp.169-178, 2009.

[3] Skormin, V., Volynkin, A., Summerville, D., and Moronski, J., “Prevention of information attacks by run-time detection of self-replication in computer codes,” Journal of Computer Security, vol.15, No.2, pp.273-302, 2007.

[4] Kim, I., “A malicious process control system for protecting servers from internet worm attacks,” The Journal of Korea Information and Communications Society, vol.35, No.3, pp.431-439, 2010.

[5] Viega, J., Bloch, J., Kohno, T., and McGraw, G., “ITS4: A static vulnerability scanner for C and C++ code,” In Proceeding of the 16th Annual Computer Security Applications Conference, 2000.

[6] Wagner, D., Foster, J., Brewer, E., and Aiken, A.k “A first step towards automated detection of buffer overrun vulnerabilities,” In Proceedings of the Network and Distributed System Security Symposium, 2000.

[7] Xie, Y., Chou, A., and Engler, D., “ARCHER: Using symbolic, path-sensitive analysis to detect memory access errors,” In Proceedings of the 9th European Software Engineering Conference, 2003.

[8] Bang, K. and Kong, J., “A study on secure code editor for secure software,” Proc. of the KCC, pp.94-97, 2011.

[9] Cowan, C., Pu, C., Maier, D., Ginton, H., Walpole, J., Bakke, P., Beattie, S., Grier, A., Wagle, P., and Zhang, Q., “StackGuard: Automatic Adaptive Detection and Prevention of Buffer-overflow Attacks,” In proceeding of the 7th conference on USENIX Security, 1998.

[10] Pincus, J. and Baker, B., “Beyond stack smaching: Recent advances in exploiting buffer overruns,” IEEE Security and Privacy, vol.2, No.4, pp.20-27, 2004.

[11] https://projects.honeynet.org/honeyc, accessed Jul. 2012

[12] https://projects.honeynet.org/capture-hpc, accessed Jul. 2012

[13] Sun, X., Wang, Y., Ren, J., Zhu, Y., and Liu, S., “Collecting internet malware based on client-side honeypot,” Proc. of the 9th International Conference for Young Computer Scientists, pp.1493-1498, Nov. 2008.

[14] Kim, D., Cho, S., and Kim, H., “Efficient method to detect malicious web contents based on time-bomb,” Journal of KIISE : Computing Practices and Letters, Vol. 17, No.1, pp.51-55, 2011.

[15] Kim, D., Kim, H., Park, M., and Cho, S., “Combining devide-and-conquer and sequential visitation algorithms on high-interaction client honeypots,” Journal of KIISE : Computer Systems and Theory, Vol.39, No.2, pp.76-83, 2012.

[16] Kim, I., Jo, H., and Kim. M., “Design and implementation of a system to detect intrusion and generate detection rul against scan-based internet worms,” The KIPS Transactions, Vol.12-C, No.2, pp.191-200, 2005.

[17] Song, J. and Kwon, Y., “An RTSD system against various attacks for low false positive rate based on patterns of attacker’s behaviors,” IEICE Transactions on Information and Systems, vol.89-D, No.10, pp.2637-2643, Oct. 2006.

[18] http://www.snort.org

[19] Bowen, B., Hershkop, S., Keromytis, A., and Stolfo, S., “Baiting inside attackers using decoy document,” Proc. of the 5th International ICST Conference, pp.51-70, Sep. 2009.

[20] Bowen, B., Salem, B., Hershkop, S., Keromytis, A., and Stolfo, S., “Designing host and network sensors to mitigate the insider threat,” IEEE Security & Privacy, vol.7, No.6, pp.22-29, Nov. 2009.

[21] Salem, B., and Stolfo, S., “Decoy document deployment for effective masquerade attack detection,” Proc. of the 8th International Conference on DIMVA, pp.35-54, Jul. 2011.

[22] Kim, I., and Kim, M., “Agent-based honeynet framework for protecting servers in campus networks,” IET Information Security, vol.6, No.3, pp.202-211, Sep. 2012.

[23] Jain, P., and Sardana, A., “Defending against Internet Worms using Honeyfarm,” Proc. of the CUBE International Information Technology Conference, pp. 795-800, Sep. 2012.

◐ 저 자 소 개 ◑

김 익 수

2000년 숭실대학교 컴퓨터학부(공학사)
2002년 숭실대학교 대학원 컴퓨터학과(공학석사)
2008년 숭실대학교 대학원 컴퓨터학과(공학박사)
2009년~현재 숭실대학교 컴퓨터학부 교수
관심분야 : 시스템 보안, 네트워크 보안, 모바일 보안
E-mail : iksuplorer@gmail.com

최 종 명

1992년 숭실대학교 전자계산학과(공학사)
1996년 숭실대학교 대학원 컴퓨터학과(공학석사)
2003년 숭실대학교 대학원 컴퓨터학과(공학박사)
2004년~현재 국립목포대학교 컴퓨터공학과 교수
관심분야 : 네트워크 보안, 상황인지 시스템, u-Healthcare 시스템
E-mail : jmchoi@mokpo.ac.kr