情報保護學會論文誌
第16卷 第5號, 2006. 10

사용되지 않는 포트를 이용하여 해커를 허니팟으로 리다이렉트하는 시스템 설계 및 구현

김 익 수,† 김 명 호‡
숭실대학교

Design and Implementation of an Unused Ports-based Decoy System to Redirect Hackers toward Honeypots

Ik-Su Kim,† Myung-Ho Kim‡
Soongsil University

요 약

많은 컴퓨터 보안 시스템들은 공격 시그너처를 기반으로 해커에 대응하기 때문에 가능한 빠르고 정확하게 새로운 공격 정보를 수집하는 것이 중요하다. 이러한 이유로 허니팟과 허니팟 팜에 관한 연구가 활발히 진행되고 있다. 그러나 해커들은 IP 주소가 할당된 서버를 직접 공격하는 성향이 높기 때문에, 허니팟과 허니팟 팜은 많은 공격 정보를 수집할 수 없다. 이에 본 논문에서는 사용되지 않는 포트를 이용하여 해커를 허니팟으로 리다이렉트하는 시스템을 제안한다. 이 시스템은 해커를 유인하기 위해 일반 서버의 사용되지 않는 포트를 유인포트로 사용한다. 이 포트는 서비스를 위한 포트가 아니기 때문에 이 포트로의 모든 접근은 비정상적인 행위로 간주되어 허니팟으로 리다이렉트 된다. 결국 제안 시스템은 허니팟이 아닌 실제 서버를 공격하는 해커들의 공격 정보를 수집할 수 있게 한다.

ABSTRACT

Many computer security systems use the signatures of well-known attacks to respond to hackers. For these systems, it is very important to get the accurate signatures of new attacks as soon as possible. For this reason, honeypots and honeypot farms have been actively researched. However, they can only collect a small amount of information because hackers have a strong tendency to directly attack servers of which IP addresses are allocated. In this paper, we propose an unused ports-based decoy system to redirect hackers toward honeypots. This system opens unused ports to lure hackers. All interactions with the unused ports are considered as suspect, because the ports aren't those for real service. Accordingly, every request sent to the unused ports is redirected to a honeypot. Consequently, this system enables honeypots to collect information about hackers attacking real servers other than themselves.

Keywords : *Computer security, Signature, Honeypot, Intrusion*

Ⅰ. 서 론

지금까지 해커의 공격을 탐지하고 차단하기 위해 방화벽, 침입탐지시스템, 침입차단시스템과 같은 많

접수일: 2006년 5월 8일; 채택일: 2006년 8월 21일
* 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌습니다.
† 주저자, skycolor@ss.ssu.ac.kr
‡ 교신저자, kmh@comp.ssu.ac.kr

은 보안 시스템들이 개발되어 왔다. [1,2] 이 시스템들은 대부분 알려진 공격 유형에 관한 정보를 기반으로 공격에 대응하기 때문에 취약 호스트에 대한 해커의 공격 전략과 공격 도구들에 관한 정보 수집은 보안 시스템들의 능력을 좌우하는 중요 요소라 할 수 있다. 이러한 이유로 허니팟과 허니팟 팜에 관련된 연구가 최근 활발히 진행되고 있다. [3-6]

허니팟은 해커에게 의도적으로 공격을 허용하는 컴퓨터 자원으로서 해커에 대한 공격 정보를 수집하는데 목적이 있다. 허니팟에 의해 수집된 정보는 새로운 공격을 탐지하고 차단하기 위해 보안 시스템에 의해 사용된다. 그러나 해커가 허니팟을 발견하여 공격할 가능성은 매우 적기 때문에 해커에 대한 많은 공격 정보를 수집할 수 없는 한계가 있다. 해커의 공격 정보를 충분히 얻기 위해서는 여러 네트워크에 많은 허니팟을 배치해야 하지만, 이는 많은 재정적 부담과 인적 자원의 낭비를 초래할 수 있다. 허니팟 팜은 하나의 네트워크 내에 다수의 허니팟들이 배치된 통합 환경 기술이다. 일단 해커가 네트워크에서 사용되지 않는 IP 주소에 접근할 경우, 리플렉터라는 프록시 서버가 이를 감지하여 해커들을 허니팟 팜으로 리다이렉트 한다. 그러나 해커가 IP 주소가 할당된 실제 서버에 접근할 경우, 리플렉터는 해커의 공격을 감지할 수 없기 때문에 허니팟 팜으로 해커의 공격을 리다이렉트할 수 없다.

본 논문에서는 사용되지 않는 포트를 이용하여 해커를 허니팟으로 리다이렉트하는 시스템을 제안한다. 이 시스템은 해커를 유인하기 위해 일반 서버의 사용되지 않는 포트를 이용하며, 우리는 이러한 기능을 가지는 포트들을 유인 포트라 부를 것이다. 유인 포트는 서비스를 위한 포트가 아니기 때문에 이 포트로의 모든 접근은 비정상적인 행위로 간주되어 허니팟으로 리다이렉트 된다. 제안 시스템은 허니팟 팜 환경에서 리플렉터가 식별하지 못하는 많은 해커들의 공격을 허니팟 팜으로 리다이렉트하기 때문에 실제 서버를 공격하는 해커들의 공격 정보를 수집할 수 있게 한다. 특히, 이 시스템은 일반 사용자의 서비스 포트에 대한 접근은 허용하지만 사용되지 않는 포트에 접근했던 해커들이 서비스 포트에 접근하는 행위를 막아준다.

본 논문의 구성은 다음과 같다. 2장에서는 허니팟과 허니팟 팜에 관한 배경지식을 좀 더 살펴보고, 3장과 4장에서는 제안 시스템을 설계 및 구현한다. 5장에서는 가상의 공격을 통해 제안 시스템에 대한 테스트를 하며, 마지막으로 6장에서는 결론을 맺는다.

Ⅱ. 배경 지식

2.1 허니팟과 허니팟 팜

많은 컴퓨터 보안 시스템들은 잘 알려진 공격에 대한 시그너처를 기반으로 해커의 공격에 대응한다. 이러한 시스템에서 가장 중요한 요소는 가능한 빠르고 정확하게 새로운 공격에 대한 정보를 수집하는 것이다. [7] 그러나 기존의 보안 시스템에 의해 수집된 데이터는 불필요한 상당량의 데이터가 포함되기 때문에, 이러한 데이터로부터 공격 시그너처를 생성하기 위해서는 많은 시간과 노력이 필요하다. 이러한 이유로 허니팟과 허니팟 팜에 관련된 연구가 최근 활발히 진행되고 있다.

허니팟은 컴퓨터와 인터넷 보안을 위해 고의로 해커로부터의 공격을 허용함으로써 알려지지 않은 해커의 공격 전략과 공격 도구에 대한 자료를 수집하는 컴퓨터 자원이다. [8] 허니팟은 일반 사용자들에게 상업적이거나 공공의 목적으로 서비스를 해주는 자원이 아니기 때문에, 허니팟에 접근하는 모든 행위는 해커들의 불법 행위로 간주된다. 허니팟에 의해 수집된 모든 데이터들은 기존의 보안 시스템에 의해 수집된 데이터들에 비해 적지만 매우 가치 있는 자료들이기 때문에 해커들의 불법 행위를 신속하고 효율적으로 분석할 수 있게 해준다. 허니팟이 많은 공격 정보를 수집하기 위해서는 해커들에게 쉽게 발견되어야 한다. 그러나 인터넷 상에는 무수히 많은 컴퓨터들이 존재하기 때문에, 해커가 허니팟을 발견하여 공격할 가능성은 매우 적다. 해커가 허니팟을 발견하여 공격할 가능성을 높이기 위해 단순히 다수의 허니팟을 설치할 수 있지만, 이는 컴퓨터 자원의 낭비와 허니팟의 설치 및 관리를 위한 관리자의 많은 시간과 노력을 필요로 한다. 게다가 허니팟을 이용한 해커의 공격으로부터 외부 시스템들을 보호하기 위해서는 허니팟이 배치된 네트워크마다 침입탐지시스템과 방화벽을 배치해야 한다. 허니팟 팜은 다수의 허니팟과 여러 보안 시스템의 설치 및 관리에 따른 재정적, 시간적 낭비를 줄이기 위해 하나의 네트워크에 다수의 허니팟들을 배치한 통합 환경이다. [9]

허니팟 팜은 그림 1에서 볼 수 있듯이 모든 네트워크에 허니팟을 배치하는 대신 허니팟 팜에 다수의 허니팟들이 배치되며, 여러 네트워크에서 발생하는

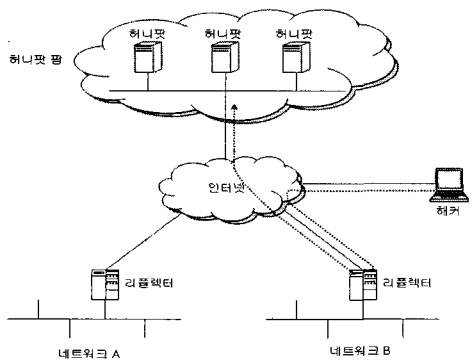

그림 1. 허니팟 팜으로 해커를 리다이렉트하는 리플렉터

해커들의 공격들이 허니팟 팜으로 리다이렉트 된다. 해커들의 공격을 허니팟 팜으로 리다이렉트하기 위해서는 각각의 네트워크마다 리플렉터라 불리는 프록시가 배치되며, 이 리플렉터는 사용되지 않는 IP 주소 중에서 미리 선택된 IP 주소로의 접근을 해커의 공격으로 간주하여 허니팟 팜으로 리다이렉트 한다. 그림 1에 표시된 점선 화살표는 사용되지 않는 IP 주소로 접근하는 해커가 리플렉터에 의해 감지되어 허니팟 팜으로 리다이렉트되는 모습을 나타낸다. 리다이렉트된 해커는 허니팟의 존재를 인식하지 못하며, 해커의 모든 불법적인 행동은 허니팟에 의해 기록된다. 통합된 환경을 제공하는 허니팟 팜은 사용하지 않는 IP 주소를 이용하여 더 많은 해커들의 공격 정보를 수집할 수 있으며, 보안을 위해 설치해야 할 침입탐지시스템과 방화벽의 비용은 물론, 보안 관리를 위한 시간과 인력 낭비를 줄일 수 있다. 그러나 해커가 예약된 IP 주소가 아닌 실제 서버에 할당된 주소로 접근할 경우에는 리플렉터가 해커를 허니팟 팜으로 리다이렉트할 수 없기 때문에 다수의 해커에 관한 공격 정보를 수집할 수 없는 문제가 있다.

2.2 해외의 허니팟 연구 동향

브라질은 2003년 9월 NBSO/Brazilian CERT와 정보통신부산하 소프트웨어 평가기관인 CenPRA가 공동으로 브라질 허니팟 연합을 결성하고 분산 허니팟 프로젝트를 시작하여, 현재 약 20여개 기관이 가입하여 운영 중이다.[10] NBSO는 브라질 인터넷 공간에서 시작된 악성 행위를 식별하고, 식별된 악성 행위들에 대해 해당 네트워크 관리자에게 공지하기 위해서 분산 허니팟을 통해 수집된 자료를 활용한다. 브라질 허니팟 연합과 침해사고 대응팀 간의 상호협력은 다른 나라에서는 찾아볼 수 없는 독특한 특징이다. 분산 허니팟 프로젝트의 목표는 브라질의 인터넷 공간에서 침해사고 탐지, 사건 연관관계, 트렌드 분석 등의 능력을 배양하기 위한 것이다. 대부분의 브라질 IP 주소 공간을 대상으로 하되 서로간의 상호작용은 분산 허니팟 네트워크로 구성되어 있다.

2.3 허니팟과 허니팟 팜 분석

허니팟과 허니팟 팜의 기본 아이디어는 사용되지 않는 IP 주소 중에서 미리 선택한 IP 주소로의 접근은 공격일 가능성이 높다는 것이다. 많은 인터넷 웜[11,12]들은 네트워크 스캔을 통해 취약한 서비스 프로그램을 사용 중인 서버들을 찾아 공격하기 때문에, 허니팟과 허니팟 팜은 인터넷 웜의 공격 정보를 수집하는데 효율적이다. 하지만 허니팟과 허니팟 팜 역시 인터넷 웜의 전파 방법에 따라 공격 정보를 수집하는데 한계가 있다. 인터넷 웜은 여러 서버들을 감염시키기 위해 임의의 IP 주소 혹은 미리 지정된 대역의 IP 주소를 가진 서버들을 공격한다. 예를 들어, 리눅스 플랫폼의 아파치 웹 서버를 대상으로 OpenSSL의 취약점을 이용하여 확산되는 Linux Slapper Worm은 IP 주소의 앞 두 자리를 임의로 선택하고 나머지 두 자리는 0.0에서 255.255까지의 영역의 주소를 가진 서버들을 공격한다. 따라서 해당 네트워크 영역에 리플렉터가 존재하지 않으면 허니팟 팜은 공격 정보를 전혀 수집할 수 없으며, 모든 네트워크마다 리플렉터를 설치하는데도 재정적인 문제가 따르게 된다. 그리고 네트워크 스캔을 통해 서버를 공격하는 인터넷 웜과 달리 해커들은 실제 서버로 직접 공격하는 성향이 강하기 때문에, 미리 선택한 IP 주소를 기반으로 하는 허니팟과 허니팟 팜은 많은 공격 정보를 수집할 수 없다. 또한 실제로 서비스하고 있는 서버로 공격이 일어날 때에는 공격 정보를 전혀 수집할 수 없다.

Ⅲ. 시스템 설계

앞서 기술했듯이 허니팟과 허니팟 팜은 미리 예약된 IP 주소에 접근하는 해커들의 공격 정보를 수집하기 때문에 많은 공격 정보를 수집하기 위해서는 서버에 할당되지 않은 더 많은 IP 주소를 필요로 한다. 하지만 오늘날과 같이 IP 주소가 부족한 현실에서

해커의 공격 식별을 위한 목적으로 더 많은 IP 주소를 확보하는 것은 거의 불가능하다. 이 장에서는 해커의 공격을 식별하기 위해 예약된 IP 주소를 사용하지 않고, 일반 서버의 사용되지 않는 포트를 이용하여 해커를 허니팟으로 리다이렉트하는 시스템을 설계한다.

3.1 제안 시스템 원리

일반 사용자들은 서버가 제공하는 특정 서비스를 이용하기 위해 해당 포트로 요청을 보낸다. 이 포트를 통해 수신된 요청은 서비스 데몬 프로그램이 처리하며, 처리된 결과는 이 포트를 통해 다시 사용자에게 전달된다. 일반적으로 서버들은 잘 알려진 서비스들인 FTP, TELNET, SMTP, HTTP 서비스를 위해서 각각 21, 23, 25, 80번 포트를 사용한다. 잘 알려진 서비스를 위해서는 예약된 0-1023번 포트가 주로 사용되기 때문에, 해커들은 0-1023번 사이의 열린 포트들을 공격한다. 따라서 제안 시스템은 해커의 공격을 허니팟 팜으로 리다이렉트하기 위해서 일반 서버의 0-1023번 포트 중 현재 인터넷 서비스를 위해 사용되고 있지 않은 포트를 이용한다.

그림 2는 웹 서비스만을 제공하는 소규모 서버에서 제안 시스템이 해커의 공격을 리다이렉트하는 원리를 나타낸다. 그림에서 해커는 실제 IP 주소가 할당된 서버 A와 B에 접속 시도를 하고 있기 때문에 리플렉터가 해커의 공격 시도를 식별할 수 없다. 그러나 제안 시스템은 사용되고 있지 않은 포트로의 요청을 해커의 공격으로 판단하여 허니팟 팜으로 리다이렉트 한다. 그림 2에서 서버 A는 웹 서비스를 위한 80번 포트와 해커를 유인하기 위한 21번 포트를 함께 열어두고 있으며, 서버 B는 웹 서비스를 위한 80번 포트와 해커를 유인하기 위한 23번 포트를 열어두고 있다. 서버 A와 B가 반드시 해커를 유인하기 위한 포트로 21번과 23번 포트만을 사용해야 하는 것은 아니며, 관리자의 선택에 따라 다수의 포트를 열 수 있다. 허니팟 팜에 배치된 허니팟은 여러 서버로부터 리다이렉트되는 해커의 요청을 처리하기 위해 21, 23번 포트를 열어두고 있다.

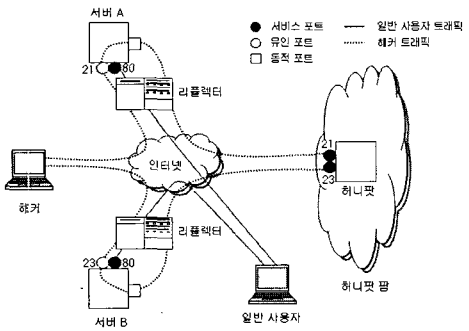

그림 2. 리플렉터를 통과하는 해커를 허니팟 팜으로 리다이렉트하는 제안 시스템

해커는 특정 서버의 취약한 서비스 프로그램을 공격하기 위해 열린 포트들을 검색한다. 스캔을 통해 서버 A의 21, 80번 포트가, 서버 B의 23, 80번 포트가 열려있는 것을 확인한 해커는 서버 A의 21번 포트 혹은 서버 B의 23번 포트로 요청을 보낼 경우, 제안 시스템에 의해 허니팟 팜에 배치된 허니팟으로 리다이렉트 된다. 그림 2에서 보이는 동적 포트들은 해커로부터 받은 패킷을 허니팟으로 포워딩하기 위해 제안 시스템에 의해 임의로 열린 포트들이다.

3.2 시스템 구조

제안 시스템은 인터넷 상에 존재하는 많은 서버들에 설치될 수 있기 때문에, 리플렉터에 의해 식별되지 않는 해커는 물론, 실제 서버를 공격하는 여러 해커들을 허니팟 팜으로 리다이렉트할 수 있다. 그림 3은 웹 서비스만을 제공하는 서버에서의 제안 시스템 구조를 나타낸다.

그림 3에 나타난 화살표는 패킷의 흐름을 나타내며, 점선 화살표는 모듈간의 데이터 및 제어 흐름을 나타낸다. 80번 포트는 실제 웹 데몬 프로그램에 의해 열린 서비스 포트이며, 나머지 포트들은 패킷 포워딩 모듈에 의해 생성된 유인 포트들이다. 만일 해커가 웹 서버를 공격하기 위해 유인 포트에 요청을 보낼 경우에는 패킷 포워딩 모듈에 의해 허니팟 팜으로 리다이렉트되기 때문에 해커의 공격 정보를 수집

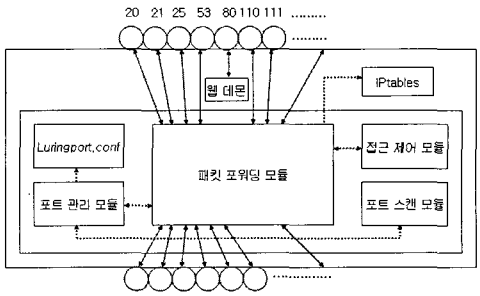

그림 3. 제안 시스템 구조도

할 수 있다. 이 시스템을 구성하는 주요 모듈은 다음과 같다.

- 포트 스캔 모듈 : 이 모듈은 서버의 닫혀있는 포트들중에서 유인 포트로 사용가능한 후보 포트 번호들을 추출하는 모듈이다. 유인 포트는 서비스를 제공하기 위해 열려있는 포트를 제외한, 닫혀있는 포트들로부터 생성되어야 한다. 그리고 많은 공격 정보를 수집하기 위해서는 임의의 포트가 아닌 잘 알려진 0-1023번 포트에서 후보 유인 포트 번호를 추출하는 것이 중요하다. 추출된 포트 번호는 유인 포트를 생성하기 위한 패킷 포워딩 모듈과 유인 포트를 동적으로 관리하기 위한 포트 관리 모듈에게 전달된다.
- 패킷 포워딩 모듈 : 이 모듈은 포트 스캔 모듈에 의해 전달받은 정보를 통해 유인 포트를 생성하고, 이 포트로 유입된 패킷들을 수신하여 허니팟 팜에 배치된 허니팟의 IP 주소로 재전송하는 역할을 한다. 또한 유인 포트에 접근한 해커들이 서비스 포트에 접근하는 것을 제한하기 위해 해커의 IP 주소를 리스트에 저장한다.
- 포트 관리 모듈 : 이 모듈은 유인 포트를 동적으로 관리하기 위한 모듈이다. 각 서버들은 종종 서비스 데몬 프로그램을 종료하거나 새로 가동되는 경우가 발생하기 때문에 관리자의 지시에 따라 유인 포트를 동적으로 열고 닫는 역할을 한다.
- 접근 제어 모듈 : 서버는 유인 포트를 열어둠으로써 해커에 의해 발생할 수 있는 네트워크 부하를 최소화해야 한다. 이 모듈은 유인 포트들로부터 유입되는 네트워크 트래픽을 감시하고, 과도한 트래픽으로부터 서버를 보호하기 위해 패킷 포워딩 모듈의 패킷 처리량을 조절한다. 또한, 유인 포트에 접근한 해커가 서비스 포트에 접근하는 것을 제한하기 위해 허니팟 팜으로 리다이렉트하는 IP-tables 룰을 생성한다.

Ⅳ. 시스템 구현

4.1. 포트 스캔 모듈

해커들은 취약한 서비스 데몬 프로그램을 공격하기 위해 잘 알려진 0-1023번 포트들을 스캔한다. 따라서 포트 스캔 모듈은 해커를 유인하기 위한 포트를 결정하기 위해 0-1023번 포트들의 사용 여부를 조사한다. 리눅스에서 netstat 명령어는 현재 시스템의 네트워크 상태를 보여주는 명령어로서, 포트 스캔 모듈은 이 명령어를 이용하여 열린 포트의 목록을 생성한다. 열린 포트 번호들은 서버가 서비스를 위해 열어둔 포트 번호들이며, 닫혀있는 포트 번호들은 해커를 유인하기 위한 포트로 사용 가능한 후보 포트 번호들이다. 이 모듈에 의해 생성된 후보 유인 포트 목록은 유인 포트 생성을 위해서 패킷 포워딩 모듈로 전달된다.

4.2. 패킷 포워딩 모듈

패킷 포워딩 모듈은 해커로부터 수신한 패킷들을 허니팟 팜으로 리다이렉트하기 위한 프로세스를 생성한다. 이 프로세스는 Luringport.conf 파일과 포트 스캔 모듈에 의해 전달된 정보를 이용하여 유인 포트를 생성하고, 이 포트를 통해 수신된 패킷을 허니팟 팜으로 포워딩한다. Luringport.conf 파일은 FTP, TELNET, SMTP, DNS, HTTP와 같이 자주 사용되는 서비스에 대한 포트 번호들을 가지고 있다. 만일 포트 스캔 모듈에 의해 전달된 후보 포트 번호가 Luringport.conf 파일에 저장된 포트 번호와 일치하면, 패킷 포워딩 모듈은 그 번호에 해당하는 포트를 유인 포트로 사용한다. 그림 4는 유인 포트를 통해 수신된 패킷의 헤더가 패킷 포워딩 모듈에 의해 변경되는 모습을 나타낸다.

유인 포트를 통해 수신된 패킷은 패킷 포워딩 모듈내의 recv()함수를 통해 헤더가 제거되어 패킷 데이터가 버퍼에 일시적으로 저장된다. 이후 패킷 데이터는 send()함수에 의해 허니팟의 IP 주소와 포트 번호가 포함된 패킷 헤더가 추가되어 인터넷에 전송된다. 인터넷을 통해 허니팟에 전달된 패킷은 허니팟

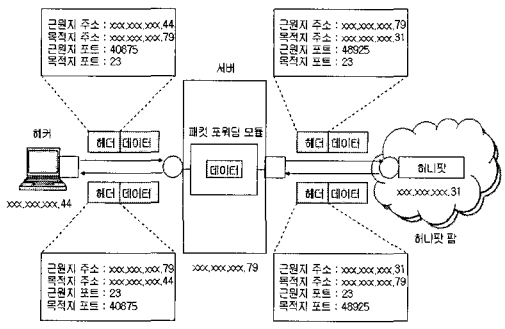

그림 4. 패킷 포워딩 모듈에 의한 패킷의 헤더 변경

의 서비스 데몬 프로그램에 의해 처리된 후 다시 서버로 전송되며, 헤더가 제거된 데이터는 패킷 포워딩 모듈에 의해 새로운 패킷으로 생성되어 해커에게 재전송된다. 따라서 해커는 허니팟의 존재를 인식하지 못하며, 자신의 공격이 피해 서버에서 수행된 것으로 인식한다.

패킷 포워딩 모듈은 유인 포트에 접근한 해커들이 서비스 데몬 프로그램을 공격하는 행위를 막기 위해 패킷 헤더로부터 해커들의 IP 주소를 추출하고, 과도한 네트워크 부하를 막기 위해 유인 포트를 통해 수신되는 초당 패킷량을 측정하여 접근 제어 모듈에게 전달한다.

4.3. 포트 관리 모듈

포트 관리 모듈은 동적으로 리다이렉트 포트를 닫거나 열기 위해서 패킷 포워딩 모듈에게 프로세스를 생성 및 종료하기 위한 메시지를 보낸다. 그림 5는 포트 관리 모듈에 의해 프로세스가 생성 및 제거되는 모습을 나타낸다.

관리자가 새로운 서비스 데몬 프로그램을 실행할 때에, 그 데몬 프로그램이 사용하는 포트가 제안 시스템에 의해 유인 포트로 사용될 경우가 발생한다. 이 때 관리자는 해당 포트를 먼저 닫아야 하는데, 이는 포트 관리 모듈을 통해 가능해진다. 관리자가 닫고자 하는 포트 번호를 입력하면, 포트 관리 모듈은 fuser 리눅스 명령어를 통해 해당 포트를 사용하고 있는 프로세스의 ID를 조사한다. 이 후 포트 관리 모듈은 패킷 포워딩 모듈에게 해당 프로세스를 종료하기 위한 명령과 프로세스 ID를 함께 보내며, 메시지를 수신한 패킷 포워딩 모듈은 SIGKILL 시그널을 보냄으로써 프로세스를 종료한다. 프로세스 종료와 함께 열려있던 유인 포트는 닫히기 때문에 새로운 서비스 데몬 프로그램은 유인 포트로 사용되었던 포트를 사용할 수 있다. 그리고 관리자가 실행중인 서비스 데몬 프로그램을 종료할 경우, 포트 관리 모듈을 통해 데몬 프로그램에 의해 사용 중이던 포트를 유인 포트로 사용할 수 있다. 포트 관리 모듈은 패킷 포워딩 모듈에게 새로운 유인 포트를 열기 위해 포트 번호와 프로세스 생성을 위한 명령을 함께 보내며, 메시지를 수신한 패킷 포워딩 모듈은 해당 포트 번호를 인자 값으로 사용하여 새로운 프로세스를 생성한다.

4.4. 접근 제어 모듈

접근 제어 모듈은 서버에 존재하는 유인 포트의 네트워크 트래픽을 감시하여 해커의 서버 접근을 제어하는 모듈이다. 일반적으로 해커는 침입에 성공한 허니팟에 다른 호스트를 공격하기 위한 공격 도구들을 설치한다. 따라서 유인 포트를 통해 리다이렉트된 해커에 의해 발생할 수 있는 네트워크 부하로부터 서버의 자원을 보호해야 한다.

그림 6은 해커의 공격으로부터 발생할 수 있는 서버의 네트워크 부하를 최소화하기 위해서 유인 포트를 통해 포워딩되는 초당 패킷량을 조절하는 코드를 나타낸다. 접근 제어 모듈은 유인 포트를 통해 포워딩 되는 초당 패킷량이 MAX_THRESHOLD 값을 초과하지 않도록 제어한다. MAX_THRESHOLD 값은 해커로부터 서버의 네트워크 자원을 보호하기 위해 사용되는 초당 최대 패킷 허용량을 나타내며, interval 변수는 패킷을 포워딩하기 위한 함수 호출의 지연시간을 나타낸다. 접근 제어 모듈이 유인 포트를 통해 포워딩 되는 초당 패킷량을 조절함으로써

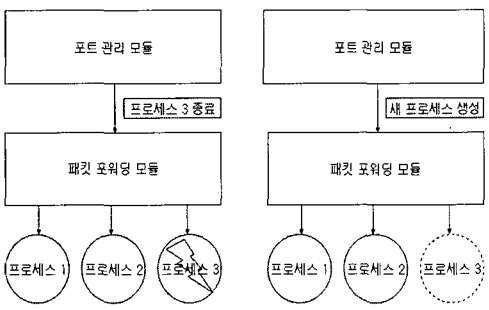

그림 5. 포트 관리 모듈에 의한 유인 포트 프로세스 생성 및 종료

```
접근제어모듈()
{
  if(유인 포트를 통해 포워딩 되는 초당 패킷량
    〉 MAX_THRESHOLD)
      interval += 200000;
  else if((유인 포트를 통해 포워딩 되는 초당 패킷
량      〈 MAX_THRESHOLD) && (interval
!= 0))
      interval -= 200000;
}
```

그림 6. 유인 포트를 통해 포워딩되는 초당 패킷량을 조절하는 접근 제어 모듈

서버는 일반 사용자에게 지속적으로 서비스를 제공할 수 있다. 유인 포트는 서비스를 제공하기 위한 포트가 아니기 때문에, 이 포트로의 접근은 해커의 공격으로 간주된다. 접근 제어 모듈은 이 포트에 접근했던 해커가 서비스 포트에 접근할 경우, 해커를 허니팟 팜으로 리다이렉트하기 위해 다음과 같은 룰을 생성한다.

iptables -t nat -A PREROUTING -p tcp -s hacker_ip --dport service_ports -j DNAT --to honeypot_ip

iptables -t nat -A POSTROUTING -p tcp -s hacker_ip --dport service_ports -j SNAT --to server_ip

hacker_ip는 패킷 포워딩 모듈에 의해 추출된 해커의 IP 주소이며, service_ports는 서버가 실제로 서비스를 제공하기 위해 열어둔 포트들이다. 이 룰을 통해 service_ports에 명시된 서비스 포트에 접근하는 해커들은 honeypot_ip에 지정된 허니팟 주소로 리다이렉트되며, 허니팟으로부터 수신된 패킷은 다시 hacker_ip에 지정된 해커에게 전달된다.

V. 실 험

제안 시스템에 대한 실험 환경은 표 1과 같다. IPv4 주소 공간에서 IP 주소가 부족한 현실을 반영하기 위해, 리플렉터는 해커의 공격을 식별하기 위한 용도의 IP 주소를 하나만 사용하였다. 반면, 제안 시스템은 어떠한 서버에도 설치가 가능하기 때문에 3대의 리눅스 서버에 설치하였다. 제안 시스템이 설치된 서버들은 웹 서비스를 위해 80번 포트를, 해커를 유인하기 위해 4개의 포트를 열어두고 있으며, 허니팟은 3대의 리눅스 서버로부터 리다이렉트되는 해커의 공격에 반응하기 위해 7개의 서비스 포트를 열어두고 있다.

제안 시스템의 리다이렉트 기능을 테스트하기 위한 시나리오는 다음과 같다. 해커는 공격하고자 하는 서버를 결정하였으며, 서버의 열린 포트를 검색하기 위해 포트 스캐너를 사용한다. 해커는 열린 포트를 사용하고 있는 서비스 데몬 프로그램을 공격하기 위해 버퍼 오버플로우 취약점을 이용한다. 공격에 성공한 해커는 루트쉘을 획득하고 루트 패스워드를 변경하며, 외부로부터 루트권한의 로그인이 가능하지 않다는 것을 인식하고 재침입을 위한 새로운 사용자 계정을 만든다. 이후에 해커는 새로운 사용자 계정을 통해 서버에 로그인한 후 일반 사용자 권한에서 루트권한으로 변경한다.

그림 7은 Sphere라 불리는 포트 스캐너를 사용하여 IP 주소가 xxx.xxx.xxx.79인 피해 서버를 스캔한 결과이다. 스캔 결과로부터 피해 서버가 21, 22, 23, 25, 80번 포트를 통해 서비스를 제공한다는 것을 알 수 있다. 그러나 실제로 서비스 포트는 단지 80번 포트뿐이며, 이를 제외한 나머지 포트들은 유인 포트들이다.

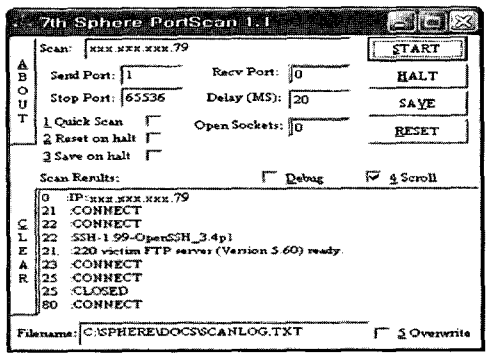

그림 7. 피해 서버의 포트 스캔 결과

표 1. 제안 시스템 실험 환경

	제안 시스템이 설치된 서버	허니팟	사용되지 않는 IP
IP 주소	xxx.xxx.xxx.55 xxx.xxx.xxx.63 xxx.xxx.xxx.79	xxx.xxx.xxx.31	xxx.xxx.xxx.234
운영체제	Red Hat Linux	Red Hat Linux	무
유인포트	21, 22, 23, 25	무	무
서비스포트	80	21, 22, 23, 25, 53, 80, 443	무

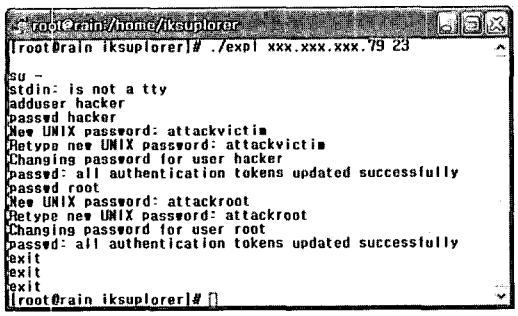

그림 8. 취약성을 가지는 텔넷 서비스 프로그램에 대한 버퍼 오버플로우 공격

그림 8은 해커가 버퍼 오버플로우 취약점을 이용하여 23번 포트와 연결된 텔넷 서비스 프로그램을 공격하는 모습이며, 공격에 성공한 이후 hacker라는 계정을 만들고 루트 계정의 패스워드를 변경하였다. 그러나 23번 포트는 제안 시스템에 의해 열린 유인 포트이기 때문에 결국 해커는 피해 서버의 루트 계정이 아닌 허니팟의 루트 계정을 획득한 것이다.

그림 9는 해커가 제안 시스템에 의해 피해 서버로부터 허니팟으로 리다이렉트된 모습이다. 앞선 공격을 통해 생성한 hacker 계정을 이용하여 피해 서버에 로그인을 한 후, 사용자 권한을 루트 권한으로 변경하였으며 ifconfig 명령어를 통해 IP 주소를 확인하였다. IP 주소가 xxx.xxx.xxx.79인 피해 서버의 텔넷 서비스 프로그램을 공격하였지만 제안 시스템을 통해 허니팟으로 리다이렉트 되었기 때문에 결국 ifconfig 명령어를 통해 확인된 IP 주소는 허니팟의 IP 주소인 xxx.xxx.xxx.31이다.

제안 시스템의 공격 수집 가능성 향상 테스트를 위한 공격 시나리오는 다음과 같다. 해커는 보안상 취약한 임의의 서버를 선택하여 직접 공격한다. 공격에 성공한 해커는 더 많은 서버들을 공격하기 위해 인터넷 웜과 같은 자동화 된 공격 프로그램을 이용한다. 인터넷 웜은 보안상 취약한 여러 서버들을 찾기 위해 네트워크 스캔을 하며 취약한 서비스 데몬 프로그램이 실행중인 서버를 발견하여 공격을 시도한다.

실제로 해커가 직접 공격하는 서버의 IP 주소를 예측하는 것은 어렵다. 또한 인터넷 웜은 공격하고자 하는 서비스 데몬 프로그램에 따라 공격하는 포트와 IP 주소가 다르다. 이에 우리는 앞서 제시한 공격 시나리오를 제안 시스템에 적용하기 위해 특정 IP 주소를 공격하거나 특정 공격 패턴을 갖는 인터넷 웜 대신, 랜덤 패킷 생성기를 구현하고 실험하였다. 랜

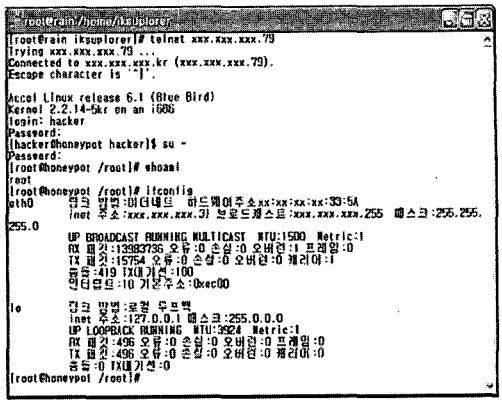

그림 9. 해커가 허니팟으로 리다이렉트된 모습을 나타내는 결과 화면

덤 패킷 생성기는 포트 번호 0 ~ 1023, IP 주소 xxx.xxx.xxx.1 ~ xxx.xxx.xxx.254의 범위에서 포트 번호와 IP 주소를 임의로 선택하여 네트워크 스캔을 하며, 이 때 허니팟에 패킷이 전달된 회수를 측정함으로써 제안 시스템의 성능을 평가하였다.

표 2는 제안 시스템이 랜덤 패킷 생성기에 의한 패킷을 허니팟으로 리다이렉트한 회수를 나타낸다. 총 다섯 번의 실험에서 허니팟은 각각 28, 33, 23, 26, 32번의 패킷을 수집하였으며, 허니팟 팜과 제안 시스템에서도 패킷 수집 회수에 차이가 발생하는 것을 볼 수 있다. 각 실험마다 같은 환경에서 패킷 수집 회수가 다른 이유는 랜덤 패킷 생성기가 랜덤 함수를 이용하여 임의의 IP 주소와 포트번호를 결정하기 때문이다. 이에 기존 시스템과 제안 시스템의 비교를 위해서 각 실험에서의 평균값을 구하고, 이 값을 이용하여 각 시스템의 성능을 비교하였다. 기존의

표 2. 가상 공격을 통한 제안 시스템 정보 수집 가능성

	실험 1	실험 2	실험 3	실험 4	실험 5	평균
허니팟	28	33	23	26	32	28.4
허니팟 팜	60	62	44	44	61	54.2
허니팟 팜과 제안 시스템 1대	80	79	52	58	79	69.6
허니팟 팜과 제안 시스템 2대	103	93	74	73	87	86
허니팟 팜과 제안 시스템 3대	117	106	89	85	94	98.2

허니팟 팜 환경에 비해 제안 시스템이 설치된 서버 1대가 추가되었을 경우 약 28%, 2대의 경우 58%, 3대의 경우 81%의 공격 정보 수집 능력이 향상된 것을 알 수 있다. 특히, 제안 시스템은 더 많은 해커들의 정보를 수집하기 위해 추가적인 허니팟 설치를 필요로 하지 않으며, 해커의 공격을 식별하기 위한 여분의 IP 주소를 요구하지 않는다.

Ⅵ. 결 론

본 논문에서는 해커의 공격을 식별하기 위해 예약된 IP 주소를 사용하지 않고, 일반 서버의 사용되지 않는 포트를 이용하여 해커를 허니팟으로 리다이렉트하는 시스템을 제안하였다. 허니팟과 허니팟 팜의 기본 아이디어는 사용되지 않는 IP 주소 중에서 미리 선택된 IP 주소로의 접근은 공격일 가능성이 높다는 것이다. 많은 인터넷 웜들은 네트워크 스캔을 통해 취약한 서비스 프로그램을 사용 중인 서버들을 찾아 공격하기 때문에 허니팟과 허니팟 팜은 인터넷 웜의 공격 정보를 수집하는데 효율적이다. 하지만 인터넷 웜과 달리 해커는 실제 서버로 직접 공격하는 성향이 강하기 때문에 미리 선택한 IP 주소를 기반으로 하는 허니팟과 허니팟 팜은 많은 공격 정보를 수집할 수 없다는 문제가 있다. 또한 실제로 서비스하고 있는 서버로 공격이 일어날 때에는 공격 정보를 전혀 수집할 수 없다.

제안 시스템은 IP 주소가 할당된 서버에 설치되어 해커의 공격을 식별하기 때문에 허니팟을 직접 공격하지 않는 해커와 허니팟 팜 환경에서 리플렉터가 식별하지 못하는 해커의 공격을 허니팟 팜으로 리다이렉트할 수 있다. 실험에서도 알 수 있듯이 제안 시스템이 설치된 서버가 증가할수록 허니팟 팜은 해커의 공격 정보를 수집할 수 있는 가능성이 높아졌다. 또한, 더 많은 해커들의 정보를 수집하기 위해 추가적인 허니팟의 설치나 해커의 공격을 식별하기 위한 여분의 IP 주소를 필요로 하지 않는다. 특히, 제안 시스템은 유인포트에 접근했던 해커를 허니팟 팜으로 리다이렉트하기 때문에 해커가 서비스 데몬 프로그램을 공격하는 것을 막을 수 있다.

제안 시스템은 현재 리눅스 운영체제가 설치된 서버에서만 동작한다. 더 많은 해커들의 공격 정보를 수집하기 위해서는 윈도우즈 운영체제에서도 동작하는 시스템을 구축하는 것이 중요하다. 향후과제로는 윈도우즈 기반의 서버 및 클라이언트에서도 동작하는 시스템을 구축하는 것이며, 일반 사용자들도 쉽게 제안 시스템을 설치 및 관리할 수 있는 그래픽 인터페이스 기능을 추가하는 것이다.

참 고 문 헌

[1] Martin Roesch, "Snort-Lightweight Intrusion Detection for Networks", *Proceedings of the LISA*, 1999.

[2] Brian Laing, Jimmy Alderson, *How to Guide: Implementing a Network Based Inrusion Detection System*, Internet Security System, 2000.

[3] L. Spitzner, *Know Your Enemy: Sebek2 A Kernel Based Data Capture Tool*, http://www.honeynet.org, 2003.

[4] Xing-Yun He, Knok-Yan, Siu-Leung Chung, Chi-Hung Chi, Jia-Guang Sun, "Real-Time Emulation of Intrusion Victim in HoneyFarm," *Proceedings of the AWCC*, 3309, pp. 143-154, Nov 2004.

[5] Miyoung Kim, Misun Kim, Youngsong Mun, "Design and Implementation of the HoneyPot System with Focusing on the Session Redirection", *Proceedings of the ICCSA*, 3043, pp. 262-269, May 2004.

[6] John G. Levine, Julian B. Grizzard, Henry L. Owen, "Using Honeynets to Protect Large Enterprise Networks," *IEEE Security and Privacy*, 2, pp. 74-75, 2004.

[7] Joseph Reves, Sonia Panchen, *Traffic Monitoring with Packet-Based Sampling for Defense against Security Threats*.

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

[9] L. Spitzer, *Honeypot Farms*, *http://www.securityfocus.com/infocus/1720*, 2003.

[10] *브라질 사이버테러 정보보호 현황 및 대응기구*, 국가사이버안전센터, Monthly 사이버 시

큐리티 1월호.
[11] A. Machie, J. Roculan, R. Russell, M. V. Velzen, *Nimda Worm Analysis*, Security Focus, Tech. Rep, Incident Analysis, 2001.
[12] A. Machie, R. Russell, *Code Red Worm*, Security Focus, Tech. Rep, Incident Analysis, 2001.

〈著者紹介〉

김 익 수 (Ik-Su Kim) 학생회원
2000년 2월: 숭실대학교 컴퓨터학부 졸업
2002년 2월: 숭실대학교 컴퓨터학과 석사
2002년 3월~현재: 숭실대학교 컴퓨터학과 박사과정
〈관심분야〉 정보보호, 시스템 보안, 인터넷 보안

김 명 호 (Myung-Ho Kim) 종신회원
1989년 2월: 숭실대학교 전자계산학과 졸업
1991년 2월: 포항공과대학교 전자계산학과 석사
1995년 3월~현재: 숭실대학교 컴퓨터학과 박사과정
1998년~1999년 University of Tennessee 전자계산학과 교환교수
1995년~현재 숭실대학교 컴퓨터학부 부교수
〈관심분야〉 병렬/분산처리, 컴퓨터 보안, BI, 클러스터링, 리눅스