제 19 회 한국정보처리학회 춘계학술발표대회 논문집 제 10 권 제 1 호 (2003. 5)

[1]세션 재지정을 이용한 능동적 HoneyPot 시스템 설계에 관한 연구

김종학, 김미영, 진봉재, 문영성
숭실대학교 컴퓨터학과
e-mail : {mygiant, mizero31, yottaman}@sunny.ssu.ac.kr, mun@computing.ssu.ac.kr

A Study on Design of the Active HoneyPot System with Session Redirection

Jonghak Kim, Miyoung Kim, Bongjae Jin, Youngsong Mun
Dept. of Computing, Soongsil University

요 약

인터넷의 급속한 발전과 네트워크 시스템의 다양화는 인간에게 편리함을 주는 여러 인프라의 혜택을 주는 반면에 악의적인 사용자로부터의 독창적이고 새로운 유형의 침입들을 야기하고 있다. 하지만, 현재 대부분의 보안 시스템이 침입에 대한 탐지 및 대응 기술에 역점을 두고 알려지지 않은 침입에 대한 탐지 및 신속한 대응이 어렵다. 본 논문에서는 세션 재지정을 이용한 HoneyPot 시스템과 다른 보안 툴과의 연동을 위한 설계 및 구현에 관해서 연구함으로써 기존의 HoneyPot 시스템이 가지는 대응 방법뿐 아니라 능동적이고 효과적인 대응방법에 대해서 제시한다.

1. 서론

인터넷의 눈부신 발전으로 현대 사회는 다양한 분야에서 인터넷 인프라의 혜택을 누리고 있다. 이러한 인터넷에 대한 의존도가 높아 질수록 그에 따른 사용자의 위험 부담은 비례하여 증가하고 있으며, 네트워크 상의 시스템에 보안을 위협하는 여러 가지 요소가 존재 하고 있다. 그 중 악의적인 사용자에 의한 침입 및 서비스 거부 공격 등은 최근 발생한 일련에 피해 사건을 통해서 더욱 관심이 집중되고 있다.

인터넷 보안 기술의 발전 속도 못지않게 이러한 악의적인 사용자의 활동은 더욱 다양화, 악성화 되는 양상을 띠고 있으며 다양한 네트워크 시스템들 사이의 연결은 독창적이고 새로운 침입 형태를 야기 하고 있어 신속하고 적절한 대응이 더욱 어려워지고 있는 실정이다.

방화벽은 현재 인터넷에 연결하고자 하는 사설 네트워크를 보호하는 수단으로 가장 널리 채택되고 있지만 보안의 정책에 의존하는 정적인 방법으로서 외부와 연결하고 있는 네트워크의 약점을 다양하고 교묘하게 이용하는 침입을 완전히 배제하는 것에는 한계가 있다.

본 연구는 악의적인 침입에 대한 적극적인 대응에 관한 대표적인 솔루션인 HoneyPot 시스템 설계에 관한 것이다. 2 장에서 HoneyPot 시스템의 간단한 소개와 관련 연구들을 살펴보고, 이어 3 장에서는 세션 재지정을 이용한 기존의 HoneyPot 시스템과 다른 보안 툴과의 연동으로 능동적인 시스템으로서 완성도를 높이고 나아가 단일 보안 시스템의 역할을 하기 위한 설계 및 구현에 대하여 제시한다.

2. 본론

2.1 HoneyPot 소개

HoneyPot 은 간단히 '해커 잡는 덫' 이란 뜻의 전문 용어 이다. 즉 해커를 잡는 유혹의 꿀단지라는 의미다. HoneyPot 은 보통 해커를 유인하기 위한 위장 서버와 추적 탐지용 소프트웨어로 구성되어 있다. 이 중 핵심은 위장서버이며, 해커를 끌어들이는 덫 역할을 한다. 접근 방법으로는 Virtual Network 내에 HoneyPot 시스

[1] "이 논문은 정보통신부 대학정보통신 연구센터 지원사업의 지원 및 한국소프트웨어진흥원의 관리로 수행되었음"

템을 추가하는 방법과 위장된 응용프로그램을 통해 가상의 서비스를 제공함으로써 HoneyPot 을 구성하는 방법이 있다. 궁극적으로는 호스트 전체를 가상 시스템으로 구축하여야 한다. Honey Pot 시스템에는 침입자에 대한 집중 감시 기능, 증거 수집 기능, 침입자 추적기능이 있어야 하며, 침입자가 HoneyPot 이라는 것을 눈치채지 못하게 해야 한다. [1~4]

HoneyPot 은 크게 "Research HoneyPot"과 "Production HoneyPot"으로 분류한다. "Research HoneyPot"은 침입자에 대한 정보를 획득하고자 설계된 HoneyPot 으로 보안에 직접적인 가치를 더해 주지는 않지만 침입자로부터의 위협을 연구하기 위해 그리고 이러한 위협에 대해 어떻게 더 나은 방어를 해야 할 지를 연구하기 위해 사용된다. "Production HoenyPot"의 목적은 위협을 완화시켜 주는데 있다. HoneyPot 은 보호, 탐지, 대응을 통해 보안에 대한 부가적인 기능을 제공해 준다.

본 연구에서 HoneyPot 시스템은 이 둘의 기능을 혼합한 형태의 HoneyPot 으로서 가상 서비스 에뮬레이션을 통해 가상 호스트를 구현하고 공격 패턴에 대한 룰을 기반으로 공격의 시점 및 종류 등을 지능적으로 판단한다. 또한 실시간으로 수집되는 패킷의 내용 및 특정 필드를 기록하고 정의된 대응 정책에 의해 수동적 또는 능동적으로 대응을 한다. 이 둘을 혼합함으로써 침입 방법에 대한 연구가치를 제공하고, 그 자체로서도 하나의 보안 기능을 제공하게 되는 것이다.

2.2 관련연구

2.2.1 IDS(Intrusion Detection System)

침입 탐지 시스템은 시스템의 불법적인 사용, 오용, 또는 불법적인 사용자나 외부 침입자에 의한 컴퓨터 시스템을 남용하는 침입을 알아내려는 시스템으로서 감사 기록, 시스템 테이블, 네트워크 부하 기록 등으로부터 사용자 행위에 대한 정보를 분석하여 침입을 판단한다. [5], [6]

일반적으로 IDS 는 감사 자료를 어디로부터 얻는가에 따라서 N-IDS(Network-based IDS)와 H-IDS(Host-based IDS)로 구분한다.

N-IDS 는 주로 네트워크 패킷이나 SNMP MIB, 응용프로그램 로그 등을 분석하여 침입탐지 여부를 분석한다. N-IDS 는 네트워크 기반의 공격을 탐지하여 네트워크 기반 구조를 보호하고자 하는 만큼 대부분의 경우 H-IDS 에서처럼 특정 호스트의 공격을 탐지하거나 상세한 기록을 남길 수는 없지만, 구현이 쉽고 하나의 시스템으로 여러 시스템을 보호할 수 있는 등의 장점이 있다.

H-IDS 는 호스트의 내부에서 얻을 수 있는 감사 자료를 수집해서 침입탐지를 수행하는 IDS 로 호스트에서 감사 자료를 얻으므로 자기 자신 이외의 컴퓨터의 침입을 탐지해 낼 수 없다. 일반적으로 보호하고자 하는 호스트 내부에서 동작을 하기 때문에 호스트 내부에의 침입 및 무결성 검출 등의 N-IDS 와는 다른 장을 가지고 있다.

본 연구에서는 Snort 를 이용하여 네트워크 기반 침입을 탐지하고, 로그기반 Audit 도구를 이용하여 호스트 기반 침입탐지를 함으로써 N-IDS 와 H-IDS 의 정보를 공유를 통해서 장·단점을 보완하도록 하고 있다.

2.2.2 Snort 를 이용한 네트워크 기반 침입 탐지

Snort 는 IP 네트워크 상에서 실시간 트래픽 분석과 패킷 로깅 수행하는 작고 가벼운 뛰어난 네트워크 침입 탐지 시스템이다. 프로토콜 헤더 분석 및 패킷의 패이로드 조사, 패턴매칭이 가능하며, 버퍼 오버플로우나 스텔스 포트스캔, CGI 공격, SMB 탐지, OS fingerprinting 시도 등 다양한 공격과 탐지를 발견하는데 사용할 수 있다. [7], [8]

Snort 는 완전한 패킷 스니핑 기능을 제공하며 이를 위해 실시간 패킷 수집 라이브러리인 'libpcap'을 사용하여, 리눅스 부팅시에 네트워크 인터페이스 모드를 ' promicious'로 전환함으로써, 지정한 인터페이스를 통해 송수신되는 모든 패킷을 수집할 수 있다. 또한 사용자가 쉽게 접근할 수 있는 유연한 룰 기반의 탐지 엔진을 갖추고 있으며, 경보 발생시 패킷 내용을 ASCII 및 tcpcump 형태로 실시간 저장이 가능하며, syslogd 및 윈도우 팝업 등의 다양한 방법을 통해 탐지 사실을 관리자에게 보고할 수 있다. 룰들은 상호 연관성을 가지는 signature 를 구성하며, Stealth Scan 공격, 잘못된 ICMP 코드 사용 등의 네트워크 공격 징후를 탐지할 수 있다. 또한 새로운 탐지 유형을 추가하기 위해 사용자가 쉽게 새로운 룰을 생성할 수 있다.

2.2.3 로그분석을 통한 호스트 기반 침입 탐지

로그 파일과 시스템 파일은 예측하지 못한 침입자의 침입 단서나 시스템 동작 오류의 분석을 위한 중요한 자료로 사용된다.

리눅스 로그는 시스템 데몬(syslogd)에 의해 관리되며, 파일의 위치는 /var/log 이며, 각각의 용도 및 역할에 맞게 다음과 같은 파일들로 저장된다. [9]

- lastlog: 각 사용자의 가장 최근의 로그인 및 시스템 사용을 기록한다.
- lastb: 사용자의 로그인 실패 정보를 기록한다.
- secure: Telnet, Ftp 등 인증을 요구하는 모든 네트워크 서비스에 대한 정보를 기록한다.
- utmp: 현재 로그인한 사용자의 기록을 나타낸다.
- wtmp: 모든 접속 사용자의 telnet, ftp 로그인 및 로그아웃 정보를 기록한다.
- messages: 시스템 로그, su, 데몬, syslogd 에 의해 생성된 메시지 및 커널 디버깅 메시지를 기록한다.

각 로그들은 용도에 맞게 활용이 가능하며 네트워크를 통한 클라이언트 서버 연결을 통해 로그파일을 주고 받을 수 있다. 또한 cron 데몬과 연동해서 원하는 시점에 해당 로그파일을 분석해서 email 이나 다른 형태로 로그 내용을 관리자에게 전송할 수 있다. 본

연구에서는 각 로그를 참조하여, 호스트 내에서의 침입을 탐지하고, 시스템 관리 서버로 실시간 전송하여 침입에 대응할 수 있도록 한다.

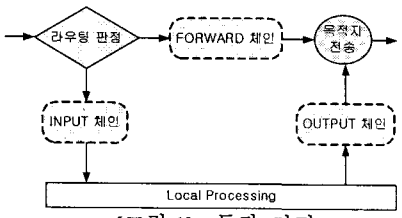
2.2.4 객체 보호

HoneyPot 서버 자체도 하나의 완벽한 시스템이므로, HoneyPot 서버가 공격으로 손상되면 침입자에 대한 분석이 어려워 지므로, 침입자로부터 최대한 시스템을 보호할 수 있어야 한다. HoneyPot 시스템의 가상도를 높이고, 침입 생존성을 향상시키기 위해 본 연구에서는 HoneyPot 시스템의 파일 시스템 보호, 패스워드 및 주요 정보 보호, /proc 파일 접근 보호, 특정 명령어 차단 그리고 로그 정보 접근 제한 등의 보호 기능을 수행하도록 한다.

2.2.5 이용한 Firewall

본 연구에서는 을 이용하여 내부 네트워크와 외부 네트워크를 분리하고 실시간으로 패킷 필터링을 지원함으로써 Firewall 시스템 구축에 사용한다. [10]

Firewall 시스템의 커널은 패킷 필터링 테이블에 있는 INPUT, OUPUT, FORWARD 세 개의 체인설정에 의해서 패킷의 포워딩 및 드롭을 결정한다. 이때 은 필터링 테이블에 필터링 규칙을 추가하고 삭제하는 기능을 수행한다.



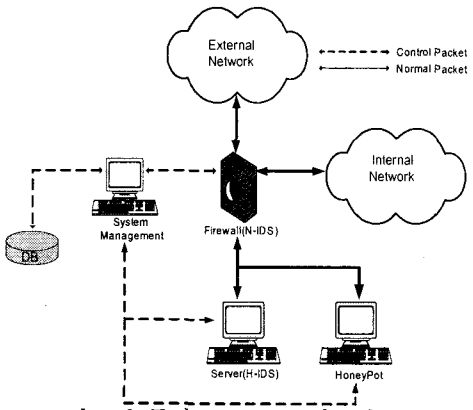
[그림 1] 동작 과정

[그림 1]은 정책에 의한 패킷 처리를 위한 동작 과정이다. 패킷이 커널에 도착하면 그 패킷의 목적지를 확인하는데 이것을 라우팅이라고 한다. 이것의 목적지가 이곳이면, 패킷은 입력 체인으로 전달된다. 그렇지 않고 커널이 포워딩 해야 하는가를 알지 못하면, 그 패킷은 Drop 된다. 포워딩이 가능하고 다른 곳이 목적지이면, 패킷은 포워딩 체인으로 간다. 이 체인이 Accept 하게 되면 이것은 포워딩 할 네트워크로 보내진다. 마지막으로 이곳에서 돌아가던 프로그램은 네트워크로 패킷을 전송할 수 있게 된다. 이 패킷은 즉시 출력체인에 보내진다. 이 체인이 Accept 하게 되면, 이 패킷은 그 목적지가 어디든지 보내진다.

3. 본 연구의 시스템 구현 방법

3.1 전체적 시스템 구조

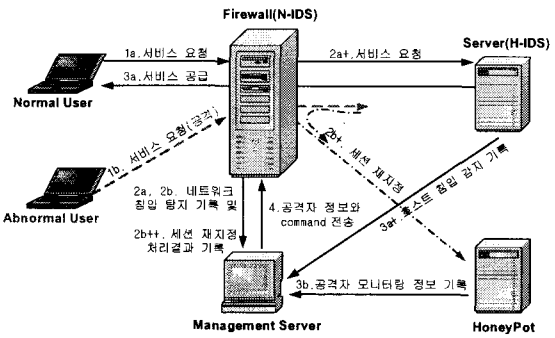
[그림 2]는 본 연구에서 사용되는 전체적인 시스템 구조이다. 시스템 구조는 정상적인 서비스 기능을 수행하는 일반 패킷과 시스템 관리 서버에 의해서 제어 및 관리를 위한 제어 패킷의 흐름으로 분류되어 있으며, 다음과 같이 각각의 다른 역할을 수행하는 4 대의 서버로 구성되어 있다.



[그림 2] 능동적 HoneyPot 시스템 구조

강제 세션 재지정 및 접근 제어를 담당하는 Firewall 서버, 일반 사용자의 Telnet 서비스 제공 그리고 서버 상에서 발생되는 비정상 및 오용 행위를 탐지하기 위한 호스트 기반 침입 감지 기능을 수행하는 Telnet 서버, 침입자들의 침입 과정 그리고 관리자 권한을 얻는 방법 및 관리자 권한을 얻은 후의 침입자들의 행위를 추적하는 HoneyPot 서버 그리고 이 모든 서버들을 동작을 제어하고 관리하며 체계적으로 침입 정보를 DB 에 저장하는 시스템 관리 서버이다.

3.2 시스템 동작 시나리오



[그림 3] 시스템 동작 시나리오

[그림 2]은 시스템 동작 시나리오로 정상 사용자(a)와 비정상 사용자(b)로 구분하여 설명하고 있다. Firewall 서버는 정상 사용자와 비정상 사용자의 서비스 요청에 대해서 두 번에 검사를 수행한다. 1 차적으로 N-IDS 를 통하여 침입을 검출하도록 하고, 침입이 검출되면 위협정보를 시스템 관리서버에 기록한다. 2 차적 검사는 의 설정에 따라서 수행한다. 검사에 의해 부적합한 사용자로부터의 서비스 요청에 대해서는 미리 구성된 HoneyPot 서버로 세션 재지정하고 적합한

사용자인 경우 서버로 서비스 요청을 전달한다. 이때 세션 재지정한 연결에 대한 처리 결과를 시스템 관리 서버로 실시간 전송한다.

Telnet 서버는 서비스 사용자의 사용상황을 호스트 IDS 에서 감시하며, 부정사용으로 감지되면 Audit 데몬이 서비스 사용자의 IP 주소, 포트 등의 부정사용 정보를 시스템 관리 서버에 보고한다. 시스템 관리 서버는 적용된 정책과 부정 사용정보를 비교하여 공격자를 규정, 서비스 거절 및 HoneyPot 서비스로의 세션 재지정을 지시하기 위한 커맨드를 Firewall 에 전송한다.

3.3 시스템 핵심 모듈

본 연구에서의 주요 기능들은 다음에 설명되는 세 가지 핵심 모듈에 의해서 동작한다.

첫째는 정상 연결에 대하여 호스트 침입 감지 시스템에서 부정사용을 탐지 시에 HoneyPot 시스템으로의 강제이동 모듈이다. 이 모듈은 Telnet 용 패킷에 대하여 정책에 따른 실시간 라우팅 변경을 지원한다. 정상 서버로 지정된 패킷의 목적지 주소를 HoneyPot 서버의 주소로 변경, 방화벽 기능을 통해 정상서버로 전달되는 텔넷 요청 패킷의 목적지 주소를 변경하고 기타 관련 헤더필드를 변경해서 HoneyPot 서버로 패킷 경로를 재지정해서 넘겨준다.

둘째는 네트워크 및 호스트 상에서의 침입 탐지 모듈이다. 공개 침입 탐지 소프트웨어인 snort 를 Firewall 서버에 설치하고, 정책에 맞게 규칙을 설정하여 ICMP, ARP 등의 취약점을 이용한 서비스 거부 공격 형태의 네트워크 상에서의 위협을 탐지하고, 서비스 서버에의 루트 세션 탈취 및 일반 사용자의 패스워드 파일 접근 등의 비정상적인 행위를 하는 호스트 상에서 침입을 로그기반 Audit 모듈을 통해 탐지한다.

셋째로 정책에 따른 통합 관리 및 실시간 모니터링 모듈이다. 이 모듈은 Firewall 및 Telnet 서버에서의 감시 데이터에 대한 전송 모듈, HoneyPot 서버로 재지정된 공격자 행위 감시 데이터에 대한 전송 모듈 그리고 전송된 데이터의 데이터베이스 저장 및 실시간 화면 출력을 위한 모듈들로 구성되어 있으며, 효율적인 모니터링 및 편리한 Firewall 정책 설정을 위한 관리 인터페이스를 지원한다.

4. 결론

본 연구에서의 공격자 유인을 위한 세션 재지정 기술은 네트워크 침입에 따르는 적극적인 대응에 대한 기술로 기존의 HoneyPot 시스템과 의 경로 재지정 기술을 연동하여 침입자의 경로를 강제로 재지정하게 함으로써 보다 능동적인 HoneyPot 시스템을 설계하고자 하였다.

이 시스템의 특징으로는 사용자 세션 재지정 구현을 통해 공격자의 위협 정보를 구현된 HoneyPot 서버를 통해서 알아내면서 서버의 정보를 안전하게 보호할 수 있게 한다. 또한 네트워크 침입 감지를 위해서 snort 를 이용함으로써 snort 에서 지원하는 수많은 침입 감지 규칙 및 효율적인 로깅 등을 이용 할 수 있으며 동시에 호스트 침입 감지를 지원함으로써 네트워크 침입 감지에서 미처 검출하지 못하는 위협들에 대해서 대처 할 수 있다.

에이전트 시스템의 통합적인 관리를 통해 강력한 모니터링 및 리포트 기능을 지원하고 공격자의 서비스 거절 및 세션 재지정 등의 적극적인 공격 대응을 위한 Firewall 규칙 추가 및 수정을 편리하게 수행 할 수 있게 할 수 있다.

향후 연구에서는 기존의 HoneyPot 관리 기능과 완벽한 연동을 통해서 네트워크 분산 컴퓨팅 환경에서 HoneyPot 만으로 하나의 단일 보안 시스템 역할을 할 수 있도록 보완, 발전 시켜 나가야 할 것이며, 무선 인프라를 이용한 실시간 이동형 에이전트 시스템의 개발을 통해서 정보 보호의 시간과 장소의 제약을 초월하는 보안 시스템의 연구가 필요 하겠다.

참고문헌

[1] 김미영, 문영성, "리눅스 기반의 실시간 침입탐지를 사용한 보안 기술에 관한 연구", 한국정보처리학회 춘계 학술 발표논문집, vol 9, No1, April, 2002, pp.134-136

[2] 김병구, 김동성, 정태명, "계층적 구조를 갖는 침입탐지 통합 시스템 설계", 정보처리학회지, vol.6, No.2, Jan.,1999

[3] 정훈조, 김병구, 정태명, "침입의 유형과 탐지 시스템의 분류", 정보처리학회지, vol.6, No.2, Jan.,1999

[4] Miyoung Kim, Youngsong Mun, "The Development of HoneyPot System", SAM'02, June, 2002, pp. 484-489

[5] Brian Laing, Jimmy Alderson, "How to Guide-Implementing a Network Based Intrusion Detection System", Internet Security System, 2000

[6] Teresa F. Lunt, "A Survey of Intrusion Detection Techniques", Computer & Security, 12(4), June 1993.

[7] M. Roesch, "Writing Snort Rules: How To write Snort rules and keep your sanity", www.snort.org.

[8] Martin Roesch and Chris Green, "Snorts Users Manual Snort Realse:1.9.0", www.snort.org, 13th August 2002.

[9] syslog(3), UNIX documentation.

[10] Rusty Russell, "Linux 2.4 NAT HOWTO", mailing list netfilter@lists.samba.org,, 29. July 2001.