거짓 세션과 허니팟을 이용한 능동적 침입 대응 기법

정회원 이 명 섭*, 신 경 철*, 박 창 현*,

An active intrusion-confronting method using fake session and Honeypot

Myung-Sub Lee*, Kyung-Chul Shin*, Chang-Hyeon Park* *Regular Members*

요 약

차세대 정보전에서는 자신의 정보 시스템에 대한 침해방지, 복구 등의 수동적인 형태의 보호뿐만 아니라 상대방의 정보 기반구조(Information Infrastructure)에 대한 공격과 같은 적극적인 형태의 보호가 요구된다. 침입이 발생함과 동시에 시스템에 대한 피해를 최소화하고 침입자 추적 등의 즉각적인 대응을 위해 정보 보호 시스템은 침입에 대한 정보를 능동적으로 분석하고 실시간으로 대응하는 기능을 제공할 필요가 있다. 본 논문에서는 거짓 세션(fake session)과 허니팟(Honeypot) 모니터링 기법을 기반으로 설계된 능동적 침입 대응 시스템을 제시한다. 본 논문에서 제시하는 능동적 침입 대응 시스템은 거짓 세션을 이용하여 DoS(Denial of Service)나 포트 스캔과 같은 공격에 대응할 수 있는 기능을 수행한다. 또한 침입 규칙 관리자(IRM : Intrusion Rule Manager)를 이용하는 허니팟과 침입자 이주를 통한 허니팟 모니터링은 모니터링 정보 수집과 침입에 대한 능동적 대처 기능을 제공함으로써 침입탐지와 침입대응에 정확도를 높인다.

Key World : DoS(Denial of Service), Port Scan, Fake Session, Honeypot

ABSTRACT

In the coming age of information warfare, information security patterns need to be changed such as to the active approach using offensive security mechanisms rather than traditional passive approach just protecting the intrusions. In an active security environment, it is essential that, when detecting an intrusion, the immediate confrontation such as analysing the intrusion situation in realtime, protecting information from the attacks, and even tracing the intruder. This paper presents an active intrusion-confronting system using a fake session and a honeypot. Through the fake session, the attacks like Dos(Denial of Service) and port scan can be intercepted. By monitoring honeypot system, in which the intruders are migrated from the protected system and an intrusion rule manager is being activated, new intrusion rules are created and activated for confronting the next intrusions.

Ⅰ. 서 론

정보통신 기술의 발전으로 네트워크 환경이 광역화, 고속화되어 이를 통한 중요 정보의 유출문제가 날로 심각해지고 있다. 특히, 현재의 정보통신 기반 구조는 서로 밀접하게 연관된 단위 구조들로 구성되어 있어 네트워크 구조의 복잡도 증가로 인한 문제는 더욱 확산되고 있다. 이러한 환경 하에서 침입자(Invader)의 공격으로 인한 중요 정보의 유출과 정보 시스템 자체에 대한 공격은 보안에 대한 심각한 문제점으로 대두되고 있다[1].

침입자에 대한 정보시스템의 보호를 위해 정보시스템 내부에 마련되어 있는 보안 기능은 기본적으로 접근제어(access control), 식별(identificatIon), 인증

* 영남대학교 컴퓨터공학과 인공지능 및 지능정보시스템 연구실(skydream@yu.ac.kr)
논문번호 : 040017-0112, 접수일자 : 2004년 1월 12일

(authentication), 인가(authorization) 그리고 기초적인 로그정보의 수집 등에 국한되어 있다. 특히, 정보시스템 자체에 다양한 보안 취약성을 내재하고 있어 정보시스템 내외부로부터의 다양한 공격에 대해 완전한 대책을 갖추지 못하고 있다. 이를 해결하기 위해서는 공격 목적에 따른 다양한 보안 기술들이 정보시스템 내부에 추가되어야 할 필요가 있다. 정보시스템 내부에서 불법적인 행동을 감시하고, 추가적인 피해를 막기 위한 대표적인 보안 요소 중의 하나가 바로 침입 탐지 시스템(IDS : Intrusion Detection System)이다[2]. 침입 탐지 시스템은 침입자로부터의 공격을 탐지하기 위해 미리 정의된 침입 규칙에 의거하여 시스템과 네트워크를 감시하는 기능을 제공한다. 그러나 다양한 유형의 공격과 침입 규칙에 포함되지 않은 새로운 공격방식에 대한 대처방안이 없으며, 침입 대응 시간에서 많은 문제점을 가지고 있다.

이러한 문제점을 해결하기 위해 1990년대 중반 미국 MIT대학의 데이비드 클록(David Clack) 교수에 의해 제안된 허니팟 시스템(Honeypot System)의 개념이 등장하였으며, 최근 들어 국내외 보안 업체들에 의해 개발되고 있다. 허니팟 시스템은 침입자를 유인하기 위한 위장 서버와 추적 탐지용 소프트웨어로 구성되어 있다[3]. 위장 서버는 침입자들의 호기심을 끌 만한 가상 정보들을 올려놓고 침입자들의 침입과 동시에 자동 추적 탐지용 소프트웨어가 작동하여 침입자를 추적한다. 그러나 허니팟 시스템은 능동적으로 작동하지 못하고 침입자가 시스템에 접속하기만을 기다려야 한다. 그리고 독립적으로 존재하는 시스템들이 갖고 있는 특징으로 인해 각각의 시스템들이 그 기능을 발휘하지 못하는 문제점을 드러내고 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 거짓 세션(Fake Session)과 허니팟 시스템을 이용한 능동적 침입 대응 기법을 제안한다. 거짓 세션은 침입자로 추정되는 부정 사용자를 강제로 허니팟 시스템으로 보내고, 서비스 거부(DoS : Denial of Service) 공격과 네트워크의 수많은 호스트에 대해 무차별적으로 이뤄지는 포트 스캔(Port Scan) 공격에 대응하는 기능을 수행한다. 허니팟 시스템에서는 이주된 침입자들의 행동을 감시하여 침입 규칙 관리자(IRM : Intrusion Rule Manager)에 의해 침입 대응 규칙을 유도함으로써 새로운 침입 규칙을 생성할 수 있다. 또한, 침입자들을 효율적으로 탐지하기 위해 규칙기반 탐지모델을 이용한다. 이 방법은 이미 알려진 침입 탐지 방식에 대한 행동패턴을 미리 규칙들로 구성해 놓고, 이를 만족하는 행위가 발생했을 시 해당되는 사용자를 침입자로 판단하게 되는 기법이다[4]. 이 기법은 이미 알려진 침입 방식을 이용하여 시스템에 침입하는 사용자를 확실하게 탐지해낼 수 있다는 장점이 있는 반면, 알려지지 않은 부정 사용자의 행위에 대해서는 탐지를 할 수 없다는 단점이 있다[5]. 이러한 문제점을 해결하기 위해 침입 규칙 관리자에 의해 생성된 규칙들을 모듈화 함으로써 프로그램 수행 도중에 동적으로 추가할 수 있게 한다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 기존의 보안 시스템인 침입 차단 시스템과 침입 탐지 시스템 그리고 두 시스템을 연동한 상호연동 보안모델에 대해 기술한다. 또한, 새롭게 대두되고 있는 허니팟 시스템에 대해 기술한다. 3장에서는 본 논문에서 제안하는 침입 대응 기법의 시스템 구조 및 동작 방식을 기술하며, 4장에서는 제안 기법의 실험 결과 및 분석을 보인다. 끝으로 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

Ⅱ. 관련 연구

본 논문에서 제안하는 시스템은 침입 탐지 시스템과 허니팟 시스템을 연동하고자 하는데 그 특징이 있다. 침입 탐지 시스템의 경우 국내외적으로 다양한 시스템들이 제안되고 있다. 그리고 허니팟 시스템 역시 ManTrap과 같은 시스템이 있으며 허니팟 프로젝트와 같은 연구가 진행되고 있다. 그러나 본 논문에서 제안하는 시스템과 같은 형태의 시스템은 개발되어 있지 않은 상태이다. 본 장에서는 기존의 보안 시스템의 연구 현황을 보인다.

2.1 침입 차단 시스템

침입 차단 시스템은 외부의 침입으로부터 내부의 특정 호스트나 네트워크를 보호하기 위한 보안기술로 접근정책을 통한 내외부간 트래픽을 제어하는 기술이다. 그 기본 목표는 보호대상이 되는 네트워크에 침입자들이 접근해 컴퓨팅 자원들을 사용하는 것을 방지하고, 자신의 정보들을 불법적으로 외부에 유출되는 것을 방지하는 것이다[6].

침입 차단 시스템은 방화벽(Firewall)이라는 구현 기술로 연구개발 및 구축이 되고 있다. 방화벽은 그 동작 원리에 근거하여 패킷 필터링 방화벽(Packet Filtering Firewall), 서킷 레벨 방화벽(Circuit Level

Firewall), 응용 게이트웨이 방화벽(Application Gateway Firewall), 하이브리드 게이트웨이 방화벽(Hybrid Gateway Firewall) 등이 있다.

패킷 필터링 방화벽은 OSI 7계층에서 네트워크 계층과 전송 계층에서 동작한다. 패킷 필터링 방화벽은 데이터 링크 계층에서 네트워크 계층으로 전달되는 패킷을 검사하여 해당 패킷내의 주소와 서비스 포트를 검색하여 정의된 보안 규칙에 따라 서비스의 접근 허용 여부를 결정하게 된다. 패킷 필터링 방화벽은 네트워크 계층과 전송 계층에서 동작하므로 어플리케이션(Application)에 구애받지 않아 다양한 인터넷 어플리케이션을 수용할 수 있다. 그러나 단순히 방화벽을 통과하는 패킷의 근원지 및 목적지 주소와 서비스 포트만을 검색하므로 세션(Session)에 대한 정보를 추적하지 못하며 데이터 크기에 대한 제어가 불가능하다.

응용 게이트웨이 방화벽은 프록시(Proxy) 서버 방식이라고도 하며 OSI 7계층에서 상위 계층인 응용 계층에서 동작한다. 응용 게이트웨이 방화벽은 클라이언트로부터 요청을 받은 후, 자신의 규칙과 비교하여 허용할 것인지를 결정한다. 요청이 받아들여진 경우 방화벽이 클라이언트 대신 목적 서버에게 요청을 전송하고, 서버로부터 응답을 받아 이를 다시 해당 클라이언트에 전송한다. 이 때 클라이언트-방화벽-서버로 이어지는 세션이 형성된다. 응용 게이트웨이 방화벽은 사용자 및 응용 서비스에서 접근 제어를 제공하며 응용 서비스 프로그램의 사용을 기록하여 감시와 추적을 위해 사용될 수 있다. 응용 게이트웨이 방화벽은 미리 정의된 어플리케이션만 수용 가능하므로 다양하고 빠르게 발전하는 인터넷 어플리케이션에 대응하지 못한다. 새로운 어플리케이션의 추가는 새로운 방화벽의 게이트웨이 프로그램을 요구하게 된다.

하이브리드 게이트웨이 방화벽은 패킷 필터링 방화벽의 장점과 응용 게이트웨이 방화벽의 장점을 결합한 방식이다. 하이브리드 게이트웨이 방화벽은 패킷 레벨의 접근 제어뿐만 아니라 응용 프로그램의 사용자를 제어할 수 있는 장점을 가진다. 그리고 어플리케이션 방식의 최대 단점인 다양한 응용 서비스의 수용은 패킷 필터링 방식으로 제공한다.

앞에서 보인 침입 차단 시스템의 문제점은 다음과 같다. 첫째, 접근정책의 중요성에 전적으로 의존한다. 이것은 보안정책을 어떻게 결정하느냐에 따라 영향을 준다. 둘째, 내부의 침입자에 대해서는 대책이 없다. 방화벽은 내외부간 트래픽을 제어하는 보안 시스템이기 때문에 내부의 상황에 대해서는 제어할 방법이 없다. 이러한 문제점은 다음 절에서 소개될 침입 탐지 시스템으로 해결할 수 있다.

2.2 침입 탐지 시스템

침입 탐지 시스템은 1987년, 데이닝(Denning)에 의해서 최초로 모델이 제안되었다. 이 보안기술은 정보 접근, 조작, 시스템 무력화 등의 특정 시스템에 불법적으로 접속하여 시스템을 사용, 오용, 남용하는 침입자들을 탐지하고 문제점에 대해서 대응하는 소프트웨어 기술을 말한다[7]. 이 기술은 침입을 탐지하는 방법에 따라 <표 1>에서와 같이 비정상 탐지(Anomaly detection)와 오용 탐지(Misuse detection)로 구성되고, 침입을 분석하는 영역에 따라 호스트 기반과 네트워크 기반으로 구성된다[8][9].

<표 1> 침입 탐지 모델 기반의 분류

	비정상 탐지	오용 탐지
탐지기법	• 통계적인 방법 • 특징 추출 • 예측 가능한 패턴 생성 • 행위 측정 방식들의 결합 • 신경망	• 조건부 확률 • 전문가 시스템 • 상태전이 분석 • 키-스트로크 관찰 • 모델에 근거한 탐지 • 패턴 매칭

호스트 기반의 침입 탐지는 시스템 로그정보와 특정 행위에 대한 감사자료 분석 등 시스템 내부에서 생성되는 정보를 기반으로 침입을 탐지하며, 네트워크 기반의 침입 탐지는 네트워크상의 패킷 헤더(packet header) 및 데이터를 분석하거나 패킷의 트래픽량을 분석하여 침입을 판단한다.

침입 탐지 시스템의 중요 요구사항은 새로운 침입 유형의 변화에 대한 자체 학습 기능, 결함 허용, 시스템 환경 변경 시 유지 및 관리에 있다. 그러나 지금까지 제시된 시스템들은 그러한 내용을 대부분 만족하지 못하고 있다[10][11][12].

이러한 한계점을 해결하기 위해서 가장 중요한 점은 정확한 침입의 판정 유무이다. 침입을 탐지하기 위해 여러 가지 적용되고 있는 방식으로는 IDES(Intrusion Detection Expert System), NIDES(Next Generation Intrusion Detection Expert System), EMERALD(Event Monitoring Enabling Responses to Anomalous Live Disturbances), STAT(State Transition Analysis Tool) 등이 있다

[13][14][15][16].

IDES는 SRI International에서 개발한 실시간 침입 탐지 시스템으로서 통계적 분석 기능을 가진 규칙기반의 전문가 시스템으로 시스템 외부에서 시스템 침입과 시스템 내부에서의 내부 위협을 감시한다. IDES는 시스템을 감시하면서 각 사용자의 정상적인 행위를 정의하고 학습하는 기능이 있으며 단일 시스템에서 동작한다. IDES에서는 제안된 속성 파일과 동작 규칙을 바탕으로 하여 감사 레코드를 수집하여 감사 기록(Audit Records), 동작 상황(Profiles), 예외 동작(Anomaly Records)의 자료들을 기록한다.

NIDES는 통계 알고리즘을 이용한 비정상 탐지 기법과 알려진 침입 형태에 대한 전문가 시스템을 적용하는 오용 탐지 기법들을 이용하여 IDES 시스템을 확장한 것이다. NIDES의 프로토타입(prototype)은 구성 요소의 재사용과 구성을 용이하게 하는 형태를 지닌 시스템으로 평가받고 있다. 구조는 네트워크 데이터의 수집과 이에 대한 규칙기반과 통계 분석을 통한 네트워크 모니터링(monitoring)과 침입 탐지를 수행할 수 있도록 확장될 수 있다. 그리고 여러 NIDES간 상호 협력이 가능하다.

EMERALD는 실시간 네트워크 기반 침입 탐지 시스템이다. 모니터라 불리는 분석과 응답 단위로 구성되며, 가능한 분석과 응답 지연을 줄이도록 설계되었다. EMERALD는 광범위한 침입 탐지 기능과 다양한 침입에 대한 대응 능력을 제공하기 위해 분산되어 있는 모니터들로부터 전달받은 자료들을 분석해서 대응한다.

STAT는 실시간의 전문가 시스템을 이용한 침입 탐지 시스템으로 침입을 상태전이 다이어그램으로 표현하며, 시스템에 영향을 줄 수 있는 행위의 기록을 보관하기 위해 관찰하는 시스템의 감사 기록을 사용한다. 또한, 다중 사용자의 감사 기록 분석을 위해 규칙기반 분석(Rule-based analysis)을 사용하며, 알려진 침입 패턴만을 사용한다. 구성을 필터링하는 작업을 수행하는 전처리기와 프로파일(profile) 기반의 비정상 탐지 모듈과 상태전이 분석 모듈이 침입을 판정한다.

앞에서 보인 바와 같이 침입 탐지 시스템은 시스템 감시 및 보고, 사용자의 활동 추적, 데이터 파일의 변경을 발견 및 보고, 시스템 설정 에러 경고 및 보안 정책을 수립하기 위한 가이드라인(guideline)을 제공을 할 수 있다. 그러나 침입 탐지 시스템이 보안의 모든 것을 해결해주지는 못하고 있다. 침입 탐지 시스템이 해결할 수 없는 문제점은 다음과 같다.

① 취약한 식별/인증 메커니즘을 보완할 수 없다.

② 사람의 관여 없이 침입을 조사할 수 없다.

③ 취약한 네트워크 프로토콜을 보완해주지 않는다.

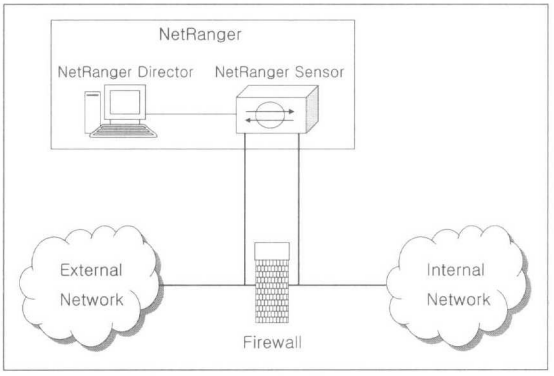
④ 시스템에서 제공하는 정보의 무결성을 보장해주지 않는다.

⑤ 불법 침입에 대한 능동적인 대응 및 차단 기능이 없다.

2.3 상호연동 보안모델

침입 차단 시스템과 침입 탐지 시스템을 상호 연동한 침입 탐지의 대표적인 시스템으로 CISCO에서 제안한 NetRanger 시스템이 있다.

NetRanger 시스템은 비교적 대규모 분산 환경에서 실시간 네트워크 기반의 침입 탐지 시스템으로 하나의 네트워크를 관리하는 NetRanger Sensor와 분산된 NetRanger Sensor를 통합적으로 관리하는 NetRanger Director로 구성된다[17]. NetRanger 시스템은 내외부 침입자의 불법적인 접근들을 모두 탐지하여 제거할 수 있고 관리자가 선택한 보안정책에 맞추어 네트워크를 통한 침입자의 공격으로부터 시스템을 보호한다. 그리고 다양한 침입 형태에 대한 탐지 및 차단의 일관된 정책 구현과 침입 탐지 기술의 확장이라고 볼 수 있다. 그러나 침입 행위를 탐지하기 전까지 시스템 자원의 파괴 및 손상을 허용한다. NetRanger 시스템의 구조는 [그림 1]에서 보인다.



[그림 5] CISCO의 NetRanger 시스템

2.3 허니팟 시스템

허니팟 시스템은 의미에 따라 상용 허니팟 시스템(Productions Honeypot System)과 연구 허니팟 시스템(Research Honeypot System)으로 구성된다[3]. 상

용 허니팟 시스템은 어떤 조직에 있어서 침입에 대한 위험도를 감소시키는 데에 의미가 있고, 허니팟 시스템을 통해 그 허니팟 시스템이 속해 있는 네트워크에 어떤 침입이 이루어지고 있는지 탐지한다. 그리고 침해 사고가 발생했을 경우 허니팟 시스템에 저장된 침입 정보를 이용하여 빠른 대처를 할 수 있도록 한다. 연구 허니팟 시스템은 가능한 많은 정보를 수집함으로써 침입자와 그 침입 방법을 연구하는 데 도움을 준다. 이렇게 수집되고 연구된 정보는 전반적인 정보 보호 기술의 발전에 기여하게 된다.
허니팟 시스템은 허니팟 시스템과 침입자 사이에서 이루어지는 상호작용의 정도에 따라서 세 가지의 단계로 분류되는데, 그 특징은 다음과 같다[18].

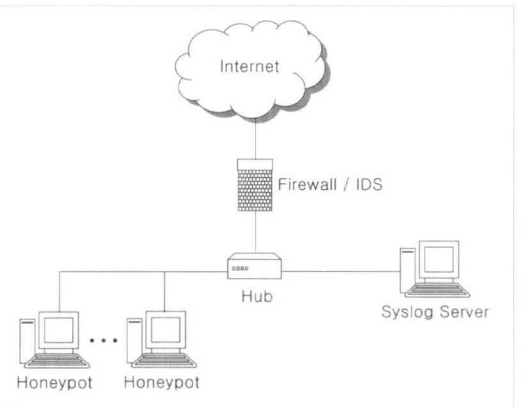
① Low level : 특정 포트에 대해 접속만 허용하고 서비스는 제공하지 않는다.
② Medium level : 실제 서비스를 제공하는 것처럼 행동하는 가상의 서비스를 제공한다.
③ High level : 실질적인 OS와 모든 서비스를 침입자가 이용할 수 있도록 제공한다.

대표적인 허니팟 시스템으로는 허니넷(Honeynet) 시스템과 Bait&Switch 허니팟 시스템이 있다. 허니넷 시스템은 침입에 대한 연구를 목적으로 만들어진 연구 허니팟 시스템이며, 일반적인 서버가 제공하는 서비스를 모두 제공한다[19]. 허니넷 시스템이 기존 허니팟 시스템과 다른 점은 단일 시스템이 아닌, 여러 개의 허니팟 시스템으로 구성되어 네트워크 자체를 허니팟 시스템으로 이용한다는 것이다. 각각의 허니팟 시스템은 Syslog에 의해 침입자가 수행하는 모든 명령어를 로그 파일에 기록하고 원격지에 있는 Syslog 서버에 해당 로그 파일을 전송한다. 이렇게 함으로써 더욱 다양하고 많은 정보를 수집할 수 있다. [그림 2]에서 일반적인 허니넷 시스템의 구조를 보인다.

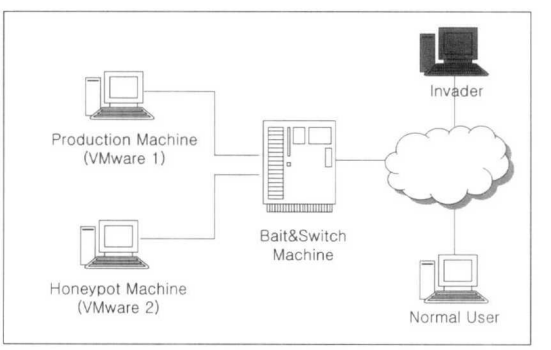
Bait&Switch 허니팟 시스템은 본 논문과 관련된 허니팟 시스템의 한 형태로 허니팟 시스템의 단점을 보완하기 위한 능동적인 시스템이다.

[그림 3]에서 Bait&Switch 허니팟 시스템의 구조를 보인다. Bait&Switch 허니팟 시스템은 기존의 ManTrap 시스템[20]이나 허니넷 시스템과는 달리 침입자가 시스템의 취약성을 파악한 후 침입자가 인식하지 못한 상태에서 허니팟 시스템으로 유인되게 한다. 침입자 유인을 위하여 Snort 시스템의 침입규칙을 이용하여 패킷 경로 재설정(rerouting)을 수행할 송신지 IP를 선별하여 송신지 IP에서 오는 두 번째 패킷부터 경로가 재설정된다[21].

그러나 Bait&Switch 허니팟 시스템은 일반적인 사용자들도 허니팟 시스템으로 유인될 수 있고 침입자가 허니팟 시스템으로부터 다른 시스템을 공격할 수 있다. 특히, 이 시스템은 서비스 거부 공격의 위협성이 내재하여 있다는 문제점이 있다.



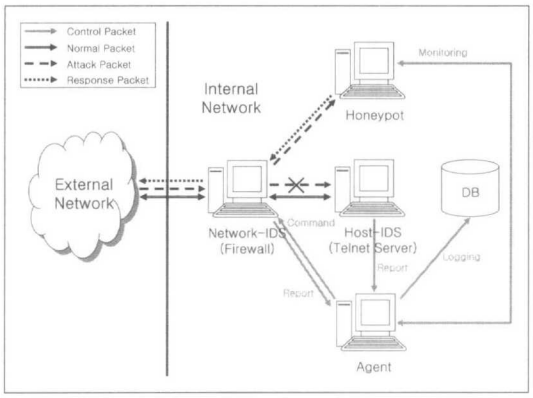
[그림 2] 허니넷 시스템



[그림 3] Bait&Switch 허니팟 시스템

Ⅲ. 제안시스템

본 논문에서 제안하는 시스템은 능동적인 침입자 유인 시스템으로 네트워크 침입 탐지 시스템(Network-IDS), 호스트 침입 탐지 시스템(Host-IDS), 허니팟(Honeypot) 시스템, 에이전트(Agent) 시스템으로 구성된다. 각각의 시스템들은 리눅스 커널 2.4.2 버전을 이용하여 구현한다. [그림 4]에서 제안 시스템의 전체 시스템 구성을 보이며, 각 구성요소들에 대한 자세한 설명은 다음과 같다.



[그림 4] 전체 시스템 구성도

[그림 4]에서 호스트 침입 탐지 시스템은 텔넷(telnet)에 접속한 사용자의 행동을 감시한 다음 파일에 기록한다. 기록된 감사 자료(audit data)를 이용하여 사용자가 실행한 파일, 디렉토리(Directory), 사용자의 접근에 대한 리턴 값 등을 추출하여 불법 행위인지를 조회한다. 조회를 통해 불법 행위를 감지하게 되면 이를 에이전트 시스템에 보고한다.

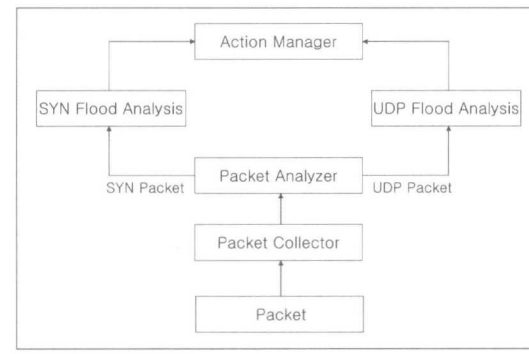
시스템의 감사 자료를 수집하는 감사 데몬(audit daemon)은 현재 리눅스에는 존재하지 않기 때문에 솔라리스의 BSM(Basic Security Module)에 포함되어 있는 감사 패키지(audit package)와 유사하게 구현한다. 감사는 프로세스 수행에 관련된 행동(PC), 파일 읽기, 쓰기에 관련된 행동(FR, FW), 파일 속성의 변화와 관련된 행동(FM), 프로세스간의 IPC(Inter Process Communication)와 관련된 행동(IP), 로그인과 로그아웃에 관련된 행동(LO)의 6개의 클래스로 구분하고 각각의 세부 이벤트(event)는 <표 2>에서 보인다.

<표 2> 각 클래스별 이벤트

Class	Event
PC	EXIT, CHDIR, KILL, CHROOT, SETPGRP, SETUID, ETGID, EXEC, NICE, SETREUID, SETREGID, SETVTX
FR	OPEN, AE_READ_LINK
FW	CREAT, LINK, MKNOD, SYMLINK, RENAME, MKDIR, TRUNCATE, WRITE, RMDIR, UNLINK
FM	CHMOD, CHOWN, FCNTL, FCHOWN, FCHMOD, FLOCK
LO	LOGIN, AT, CRONTAB, LOGOUT, SU, RSHD, PASSWD, RLOGIN, TELNET, FTP
IP	MSGCTL, MSGGET, MSGSND, SHMGET, SHMCTL, SHMAT, SHMDT, SEMCTL, SEMGET

감사 데몬은 백그라운드(background)에서 실행되고, 각 사용자가 생성하는 프로세스가 커널 내의 시스템 호출(System Call)을 하게 되면, 시스템 호출 내부의 audit()라는 함수가 이벤트를 /proc/audit파일에 기록한다. 감사 데몬이 /proc/audit파일을 참조하여 최종적으로 로그 파일에 남기게 된다.

호스트 침입 탐지 시스템에서 사용하는 데이터들은 /etc/security/audit.conf파일에 등록시키면 감사 데몬은 등록된 클래스의 이벤트만 로그 파일에 남긴다. 네트워크 침입 탐지 시스템에서는 패킷 헤더를 분석하고 서비스 거부 공격을 탐지한다. 패킷의 헤더를 분석하여 변조된 부분이 있으면 Agent에 보고한다. 또한, TCP헤더에서 SYN 플래그(flag)가 체크(check)된 패킷과 UDP 패킷의 총 크기가 기준 값보다 큰 경우와 일정 시간 내에 들어온 패킷의 수가 기준 값보다 많을 경우 서비스 거부 공격으로 간주한다. 그림 5에서 네트워크 침입 탐지 시스템의 동작 구조를 보인다.



[그림 5] 네트워크 침입탐지시스템 동작 구조

허니팟 시스템에서는 호스트 침입 탐지 시스템에서 보내온 침입자의 정보를 바탕으로 새로운 텔넷 세션을 만들어 네트워크 침입 탐지 시스템에서 목적지 주소가 변환되어 들어오는 패킷을 받아들일 준비를 한다. 그리고 침입자가 눈치 채지 못하게 호스트 침입 탐지 시스템에서 허니팟 시스템으로 작업 환경을 옮기게 하여 침입자의 정보를 수집한다.

에이전트 시스템에서는 호스트 침입 탐지 시스템의 감사 데몬이 수집한 가공하지 않은 데이터를 수집하고, 수집한 데이터를 분석하여 필요한 데이터를 추출한다. 추출한 데이터를 각각의 침입 탐지 시스템으로 전송하고, 침입 탐지 시스템으로부터 오는 결과에 따라 침입 대응에 관련된 명령을 수행한다.

3.1 침입 탐지 시스템

본 논문에서는 침입 탐지의 한 방식으로 규칙 기반 분석 기법을 사용한다. 그리고 각 규칙들을 모듈화 하여 프로그램 수행 도중에 동적으로 추가할 수 있도록 한다. 규칙의 구성 방식은 if문을 사용하는 것으로 다음과 같이 구현된다.

```
if ( condition ) then ( action )
if{ ( condition A ) and
                ( condition B ). }
                    then ( action )
```

이는 이미 주어진 조건을 만족하는 경우 행동으로 이어지게 되는 단순한 구문이나 판단을 목적으로 하는 전문가 시스템을 구성하는 가장 기본적인 방법이라고 할 수 있다. [그림 6]에서 호스트 침입 탐지 시스템의 파일 읽기, 쓰기에 관련된 행동의 규칙을, [그림 7]에서 이렇게 만들어진 규칙들은 모듈화 하여 동적으로 추가할 수 있는 함수를 보인다.

```
if ( Does the event exist? ) {
  /* check event */
      if (( Did the action succeed? )
         &&
      ((strcmp( comparison event and
         rule ) == 0) &&
      ( event number ) &&
      ( Is the authority different? )))
    {
      send_queue(from, to, type, len,
                 buf);
      /* send to agent */
      return 1; /* intrusion */
    }
    else
      return -1;
}
```

[그림 6] 파일의 읽기, 쓰기 행동 규칙

[그림 6]의 규칙은 사용자에 의해 발생한 이벤트를 조사하여 이벤트에 대한 사용자 행위의 성공 여부, 이벤트와 규칙 비교, 이벤트 번호, 권한이 있는지 등을 검사한다. 이때, 조건에 맞으면 침입으로 판단하고 침입자의 정보를 에이전트에 보고한다.

[그림 7]에 보이는 함수는 탐지 규칙을 추가하기 위한 모듈로, 먼저 탐지 규칙 모듈이 저장되어 있는 디렉토리를 검색한 다음, 추가할 탐지 규칙 모듈이 있으면 그 모듈을 읽고 침입 탐지에 적용한다.

```
void rule_call(.....) {
  /* detection module directory */
  rule_func rules;
  /* module function name */
  .....
  while(dir = readdir(dp) ) {
  /* read directory entry */
    if (dir->d_ino != 0) {
      if (strcmp(r_name, dir->d_name)
          == 0) {
  /* comparison rule and directory name*/
        strcat(host_rule_name, r_name);
        /* concatenate rule name to host
           rule name */
         module = dlopen.(host_rule_name,
                  RTLD_LAZY);
        /* read detection module */
        .....
      }
      rules = dlsym(module,"rule");
      /* loading detection module */
      .....
    }
    (*rules)(data);
     /* execution detection rule */
    dlclose(module);
  }
}
```

[그림 7] 탐지 규칙 모듈 추가 함수

[그림 8]에서 네트워크 침입 탐지 시스템에서 패킷 헤더를 분석하고 서비스 거부 공격을 탐지하는 함수를 보인다. [그림 8]의 syn_judge() 함수는 SYN 패킷이 0.5초 이하의 간격으로 100개 이상 들어오면 SYN Flooding 공격으로 판단한다. 그리고 udp_judge() 함수는 UDP 패킷이 0.5초 이하의 간격으로 100개 이상 들어오고 패킷의 크기가 8192를 넘으면 UDP Flooding 공격으로 판단한다.

[그림 9]는 포트 스캔을 탐지하는 함수로 입력된 패킷이 정해진 규칙에 의해 포트간의 지연(delay)이 3초 이내이고 스캔 되어지는 포트의 수가 5개 인지를 검사한다. 본 논문에서는 스캔 탐지를 위해서 최소한 같은 발신지 주소로부터 스캔 되어지는데 필요한 포트의 숫자(SCAN_COUNT _THRESHOLD)와 포트간의 지연 값(SCAN_DELAY_THRESHOLD)으로 설정한 시간 안에 발생한 공격에 한정된다.

탐지 규칙 모듈에 의해 침입을 탐지한 후 감사 자료를 이용하여 침입자의 정보를 에이전트 시스템에 전송한다. 침입 탐지에 대한 메시지를 수신한 에이전트 시스템은 침입 결과의 중요도에 따라 0, 1, 2단계로 나누고, 침입 대응 행동을 결정한다.

```
int syn_judge(.....)
{
    if(interval < 0.500000 &&
      syn_cnt > 100) {

      rt_val = insert_src_ip(ip);
    /* Flooding attacker's address
       update in database */
       .......
    }
    else if(interval > 0.500000) {
    /* delete SYN packet in the list */

       return (del_first_syn(syn_ip));
    }
    return 0;
}

int udp_judge(.....)
{
    if(interval < 0.500000 &&
      udp_cnt > 100) {

      if(p->prob_net.udp_len > 8192)
      { ..... }
      return 1;
    }
    else if(interval > 0.500000) {
    /* delete UDP packet in the list */
       del_first_syn(udp_ip);
    }
}
```

[그림 8] 서비스 거부 공격 탐지 함수

```
.....
int ScanDect(.....)
{
  .....
  if (now-current->timestamp <=
        SCAN_DELAY_THRESHOLD &&
            now >= current->timestamp)
  {
   /* the inspection whether 5 packets
      came within the 3 seconds */
     current->timestamp = now;

     if (current->count ==
        SCAN_COUNT_THRESHOLD)
       return(TRUE);

     if (current->count ==
        SCAN_COUNT_THRESHOLD - 1) {
       current->count++;
       return(TRUE);
     }
     .....
  }
}
```

[그림 9] 포트 스캔 탐지 함수

① 0단계 : 일반적으로 사용자의 행동에 이상이 없는 경우이다. 아무런 대응을 하지 않는다.

② 1단계(시나리오 B) : 침입 판정에 의하여 침입자의 텔넷 접속을 끊는다.

③ 2단계(시나리오 C) : 침입 판정에 의해 침입자를 텔넷 서버에서 허니팟 시스템으로 이주시킨다.

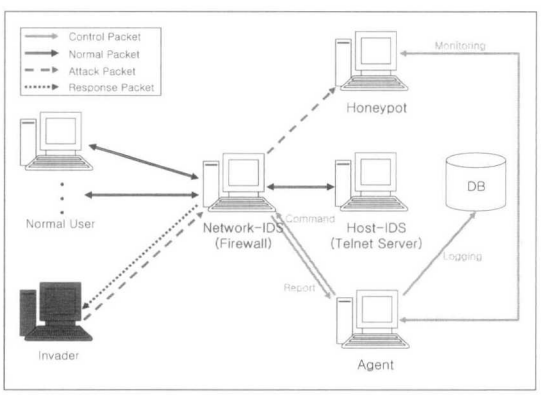
시나리오 B는 침입자의 정보(port number, UID, PID)를 이용하여 침입자에 대한 패킷을 폐기 시키는 명령을 네트워크 침입 탐지 시스템에 전송한다. 그리고 텔넷 서버에 접속한 사용자의 프로세스를 종료한 후 침입자의 로그 정보를 파일에 기록한다. 시나리오 C는 침입자의 정보를 이용하여 허니팟 시스템으로 강제 이동하는 명령을 수행한다. 그리고 침입자의 정보를 로그파일에 기록한다. 이 로그 파일은 침입자의 접속부터 부정사용 행위까지 텔넷 사용에 대한 감사 기록이다. 차후에 기록된 정보를 허니팟 시스템에 기록된 정보와 종합하여 침입에 대한 전체적인 분석을 할 수 있도록 한다.

3.2 침입 대응 절차

이 절에서는 본 논문에서 제안한 침입 대응 절차를 시나리오 A, B, C와 거짓 세션으로 나누어 설명한다.

3.2.1 침입 대응 시나리오

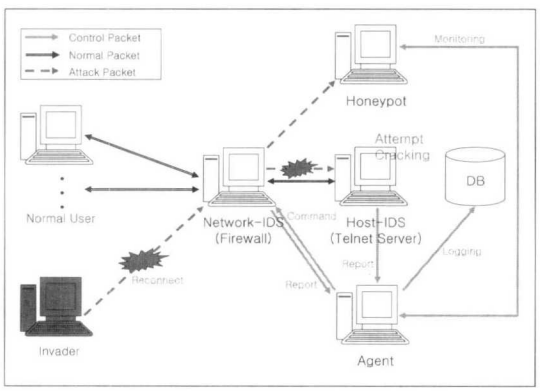
[그림 10]에서 시나리오 A의 동작 과정을 보이고 있으며, 동작 과정은 다음과 같다. 먼저 네트워크 침입 탐지 시스템에서 포트 스캔을 탐지하면 에이전트 시스템에 침입자의 정보를 보고한다. 그리고 닫힌 임의의 포트 중 하나를 선택하여 주소 변환기를 사용하여 허니팟 시스템으로 세션을 유도한다.



[그림 10] 시나리오 A의 동작 개념도

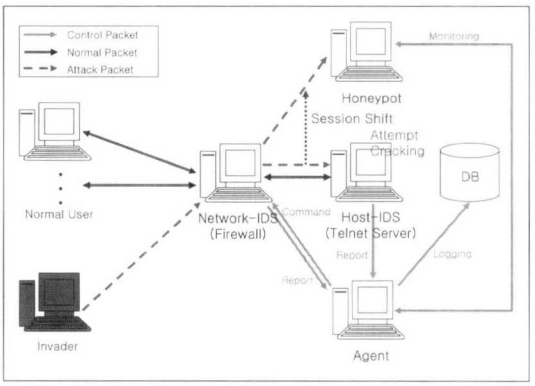
본 논문에서는 임의로 5000번을 지정하여 포트 스캔을 탐지한 후 5000번에 들어오는 패킷에 대해서 허니팟 시스템으로 세션을 유도한다. 두 번째로, 포트 스캔 시 커널이 반응하여 내보내는 패킷을 전송되지 못하게 하고, 임의로 만든 패킷을 전송한다. 전송 패킷의 플래그 비트(flag bit)를 SYN, ACK로 설정하여 닫힌 포트이지만 열려있는 것으로 위장한다.
시나리오 B는 텔넷 서버에 접속 중인 사용자의 행위를 호스트 침입 탐지 시스템에서 감시하며, 침입자로 감지되면 감사 데몬이 에이전트 시스템에 보고하고, 관리자의 정책에 따라 시나리오 B와 C를 선택한다. 시나리오 B로 결정되면 네트워크 침입 탐지 시스템에 침입자에 대한 패킷을 폐기시키는 명령을 전송한다. 그리고 텔넷 서버에 접속한 사용자의 프로세스를 종료한다.

그리고 침입자가 텔넷 서버에 재접속 요구 시 허니팟 시스템으로 연결한다. [그림 11]에서 시나리오 B의 동작 과정을 보인다.



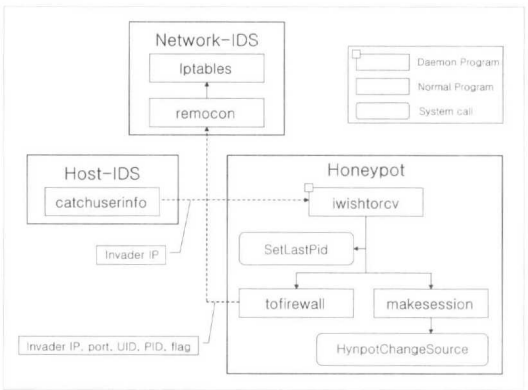
[그림 11] 시나리오 B의 동작 개념도

시나리오 C는 본 논문에서 가장 핵심이 되는 내용으로 침입자가 의심하지 않게 허니팟 시스템으로 이주하게 하는 방법이며, 동작 과정은 [그림 12]에서 보인다. [그림 12]에서 텔넷 서버에 접속 중인 침입자의 접속을 유지한 상태로 허니팟 시스템으로 이주시킨다. 부정사용이 감지되면 감지된 사용자의 정보를 허니팟 시스템 데몬에 보내 침입자를 받을 준비를 한다. 준비가 완료되면 허니팟 시스템의 데몬은 네트워크 침입 탐지 시스템의 Iptables명령을 실행하여 실시간 데이터 패킷의 경로를 바꿔준다.



[그림 12] 시나리오 C의 동작 개념도

침입자를 허니팟 시스템으로 이주하기 위한 텔넷 세션의 이동 과정은 [그림 13]에서 보인다.



[그림 13] 텔넷 세션 이동 과정

[그림 13]에서 호스트 침입 탐지 시스템의 감사 데몬은 침입자의 정보를 알아내기 위한 프로세스(catchuserinfo)를 실행하고, 침입자의 IP와 포트, UID, PID정보를 허니팟 시스템으로 전송한다.
호스트 침입 탐지 시스템에서는 사용자 정보를 포함하는 구조체(UserInfo)를 허니팟 시스템의 접속자 모니터링을 위한 데몬 "iwishtorcv"에 전달한다.
"iwishtorcv" 데몬은 Host-IDS의 catchuserinfo에 의해 전달받은 UserInfo 구조체 정보를 바탕으로 새로 만들어질 침입자의 텔넷 프로세서와 쉘(shell) 프로세서의 PID를 동기화 한다. 그리고 침입자의 패킷이 목적지 주소가 변환되어 허니팟 시스템으로 보내지도록 네트워크 침입 탐지 시스템의 Iptables명령을 실행한다. 또한 make- session에 의해 침입자의 정보를 호스트 침입 탐지 시스템에서의 정보와 동기화 한다.

3.2.2 Fake Session

거짓 세션은 침입자의 포트 스캔을 감시하여 포트 스캔 시 전송 패킷에 대한 부정확한 응답을 함으로써 침입자의 포트 스캔 행위를 쓸모 없는 것으로 만든다. 포트 스캔으로부터 네트워크 침입 탐지 시스템에 지정된 포트로 들어오는 모든 패킷에 대해 커널이 응답 패킷 중에서 플래그 비트가 RST, ACK인 패킷을 모두 폐기시키고 패킷 생성기로 만든 패킷이 전송되도록 한다.

응답 패킷의 플래그 비트를 SYN, ACK로 설정하여 닫힌 포트이지만 열린 것으로 위장하여 전송한다. 그리고 특정 포트를 허니팟 시스템에 연결시켜 시나리오 A에 해당하는 침입 대응 방법을 제공한다. <표 3>에서 Fake TCP헤더의 구성을 보인다. 그리고 <표 4>에서는 네트워크 침입 탐지 시스템에서 Iptables 명령을 보인다.

<표 3>에서 포토가 열려있는 경우 포트 스캔 공격자로부터 네트워크 침입 탐지 시스템의 지정된 포트로 들어오는 모든 패킷에 대해 커널이 보내는 패킷 중 플래그 비트가 SYN, ACK인 패킷을 폐기 시키는 명령을 수행한다.

<표 3> Fake TCP 헤더 구성

RST	SYN

tcp->source = htons(sport);
tcp->dest = htons(dport);
tcp->seq = htonl(seq);
tcp->ask_seq = htonl(ask);
tcp->res1 = 0;
tcp->doff = 5;
tcp->window = htons(0);
tcp->fin = 0;
tcp->syn = 0;
tcp->ask = 1;
tcp->rst = 1;
	tcp->source = htons(sport);
	tcp->dest = htons(dport);
	tcp->seq = htonl(seq);
	tcp->ask_seq = htonl(ask);
	tcp->res1 = 0;
	tcp->doff = TH_OFFSET;
	tcp->window =htons(WIN_SIZE);
	tcp->fin = 0;
	tcp->syn = 1;
	tcp->ask = 1;
	tcp->rst = 0;

포트가 닫힌 경우에는 플래그 비트가 RST, ACK인 패킷을 폐기 시키는 명령을 수행한다. 그리고 호스트 침입 탐지 시스템에서 탐지된 침입자를 허니팟 시스템으로 이주 시킬 경우 침입자의 패킷 경로를 허니팟 시스템으로 변경하는 명령을 수행한다.

<표 4> Iptables 설정 명령

	명 령
포트가 열려있는 경우	/sbin/iptables -A OUTPUT -p tcp --tcp-flags ALL RST, ACK -d $TARGET$ --sport $PORT -j DROP
포트가 닫힌 경우	/sbin/iptables -A OUTPUT -p tcp --tcp-flags ALL SYN, ACK -d $TARGET$ --sport $PORT -j DROP
허니팟 시스템으로 연결	/sbin/iptables -I PREROUTING -t nat -s $TARGET$ -p tcp --dport $PORT$ -j DNAT --to 192.168.0.6:23

Ⅳ. 실험결과 및 분석

본 장에서는 본 논문에서 제안한 능동적 침입 대응 기법으로 시나리오 A, B, C와 거짓 세션의 실험 결과 및 분석을 보인다.

4.1 시나리오 A 실험

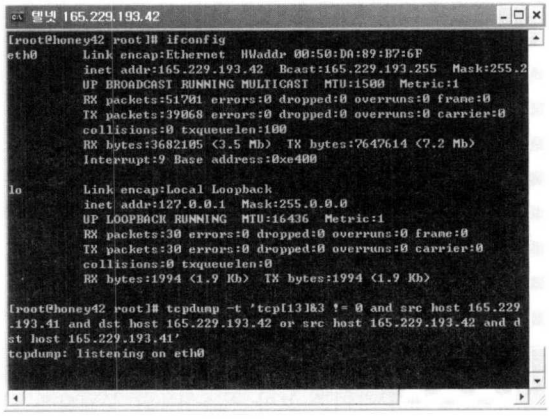
시나리오 A는 네트워크 침입 탐지 시스템에서 서비스 거부 공격 및 스캔 공격에 대응하는 기법이다. [그림 14]는 165.229.193.42의 IP 주소를 갖는 네트워크 침입 탐지 시스템에 SYN Flooding 공격을 TcpDump[22]를 이용해 패킷을 캡처(capture)한 화면이다.

[그림 14] SYN Flooding 공격 화면

네트워크 침입 탐지 시스템에서 SYN Flooding 공격을 탐지하면 대응 기법에 의해 침입자 시스템에서 오는 SYN 패킷을 폐기한다. [그림 15]는 네트워크 침입 탐지 시스템에 SYN Flooding 공격을 한 후, 텔넷 접속을 요청하는 화면이다. [그림 15]에서 보면 침입자 시스템으로부터 오는 SYN 패킷이 폐기되어 텔넷 접속 요청이 받아들여지지 않는 것을 볼 수 있다.

[그림 15] 텔넷 접속 요청 화면

[그림 16]은 침입자 시스템과 네트워크 침입 탐지 시스템 사이의 SYN 패킷을 수집하기 위해 TcpDump를 구동한 화면으로 두 시스템 사이에 SYN 패킷이 없는 것을 볼 수 있다.



[그림 16] SYN 패킷 수집 화면

4.2 시나리오 B 실험

시나리오 B는 텔넷 서버에 접속 중인 사용자가 침입자로 탐지되면 침입자의 접속을 종료하고 재접속 요구 시 허니팟 시스템으로 연결하는 기법이다. 호스트 침입 탐지 시스템에서 침입자로 탐지 되면 감사 데몬이 에이전트 시스템에 침입자의 정보를 전송하고 네트워크 침입 탐지 시스템에 침입자에 대한 패킷을 폐기시키는 명령을 보낸다. 그리고 텔넷 서버에 접속한 사용자의 프로세스를 종료한다. 그리고 침입자가 텔넷 서버에 재접속 요구 시 허니팟 시스템으로 연결한다. [그림 17]에서 감사 데몬 구동 화면을 보인다.

[그림 17] 감사 데몬 실행 화면

[그림 18]은 텔넷 서버에 접속한 사용자가 권한이 없는 행위를 하여 탐지 규칙에 의해 침입자로 탐지되어 텔넷 접속이 종료된 화면을 보이고 있다.

[그림 18] 침입자의 텔넷 접속 종료 화면

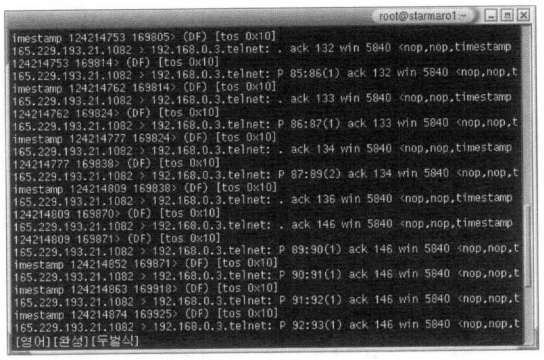
4.3 시나리오 C 실험

시나리오 C는 텔넷 서버의 접속한 사용자가 침입자로 탐지 된 경우, 그 침입자를 의심하지 않게 허

니팟 시스템으로 이주하는 기법이다.

[그림 19]와 [그림 20]은 시나리오 C를 수행할 때 생성되는 패킷을 TcpDump를 이용하여 캡처한 화면이다.

[그림 19]에서 침입자가 허니팟 시스템으로 이주되기 전까지의 침입자와 텔넷 서버간의 패킷을 보여 주고 있다.



[그림 19] 침입자와 텔넷 서버간의 패킷

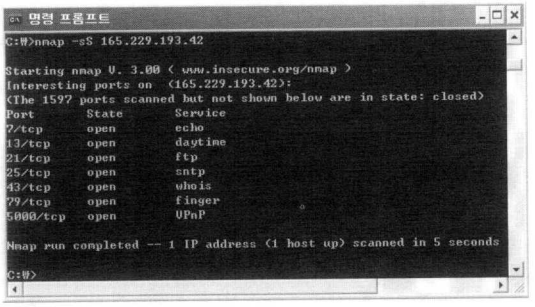
[그림 20]은 허니팟 시스템의 makesession에 의해 새로운 세션이 만들어지고, tofirewall에 의해 패킷의 경로가 텔넷 서버에서 허니팟 시스템으로 재설정되어 침입자와 허니팟 시스템간의 패킷을 보여주고 있다.

[그림 20] 허니팟시스템으로 이주한 후 패킷

4.3 거짓 세션

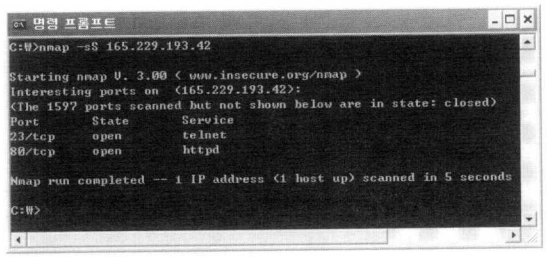
거짓 세션은 포트 스캔을 감시하여 포트 스캔 시 보내는 패킷에 대한 부정확한 응답을 함으로써 침입자의 포트 스캔 행위를 쓸모없는 것으로 만들고 임의의 포트로 침입자의 패킷을 유도하여 허니팟 시스템으로 이주 시키는 기법이다. 본 절의 실험을 위해 포트 스캔 툴인 Nmap[23]을 사용한다.

[그림 21]은 거짓 세션의 정책 없이 네트워크 침입 탐지 시스템으로 포트 스캔한 결과이다. 그림 에서 보면 실제로 서비스 하고 있는 포트 정보가 침입자에게 그대로 노출되는 것을 볼 수 있다.



[그림 21] 거짓 세션 정책 적용 전

[그림 22]는 네트워크 침입 탐지 시스템에 거짓 세션의 정책을 적용한 후 포트 스캔을 한 결과이다. 거짓 세션의 정책에 의해 포트 스캔을 탐지하면 커널이 반응하는 패킷을 폐기 시키고 패킷 생성기에 의해 생성된 패킷을 스캔 공격을 한 침입자 시스템에 전송한다. 그림 22에서 보면 실제 서비스 하고 있지 않지만 서비스를 하고 있는 것처럼 침입자에게 거짓 응답을 보내는 것을 볼 수 있다.



[그림 22] 거짓 세션 정책 적용 후

VI. 결 론

본 논문에서는 기존 보안 시스템들의 문제점을 해결하기 위해 거짓 세션과 허니팟 시스템을 이용한 능동적 침입 대응 기법을 제안하고 구현하였다. 본 논문에서 제안한 거짓 세션은 네트워크상의 수많은 호스트에 대해 무차별적으로 이뤄지는 스캔 공격에 대응하며, 잘못된 거짓 정보를 줌으로써 침입자를 허니팟 시스템으로 유도한다. 그리고 침입자를 허니팟 시스템으로 유도하기 위한 방법으로 세 가지의 시나

리오를 제안하였다. 첫째, 네트워크 침입 탐지 시스템에서 감지된 침입자의 텔넷 접속 요구 시 텔넷 서버 대신 허니팟 시스템으로 연결시킨다. 둘째, 호스트 침입 탐지 시스템에서 감지된 부정 사용자의 연결을 끊고, 재접속 요구 시 허니팟 시스템으로 직접 연결시킨다. 셋째, 둘째와 같은 상황에서 부정 사용자의 연결을 끊지 않고 침입자가 접속 상태를 유지한 채 허니팟 시스템으로 연결시킨다.

본 논문에서 제안된 기법의 구현을 통해 얻을 수 있는 장점으로는 첫째, 사용자 강제 이동 구현을 통해 침입자의 크래킹 내역을 충분히 알아내고 서버의 정보를 안전하게 보호할 수 있다. 둘째, 허니팟 시스템에 이동한 침입자의 행동을 모니터링 함으로써 새로운 공격 유형을 탐지 및 새로운 규칙의 추가가 용이하다. 셋째, 거짓 세션의 구현으로 호스트의 취약점이 쉽게 노출되지 않고, 허니팟 시스템과 연동하지 않을 시에는 방화벽이나 일반 호스트에 대한 보안 솔루션으로 유용하다. 향후 연구 과제로는 텔넷 서버와 허니팟 시스템간의 침입자의 작업 환경을 완벽하게 동기화할 수 있는 방법에 대한 연구가 필요하다.

참 고 문 헌

[1] CERTCC-KR, 한국정보보호진흥원, http://certcc.or.kr, 2003.

[2] Stephen Northcutt, Network instru- sion Detection - Third Edition, New Riders, 2002.

[3] Lance Spitzner, Honeypots - Track- ing Hackers, Addison- Wesley, 2002.

[4] W. Lee and S. J. Stolfo, "Data min- ing approaches for intrusion detection.", In Processing of the 7th USENIX Security Symposium, San Antonio, TX, 1998.

[5] P. A. Porras, "STAT: A State Transition Analysis Tool For Intrusion Detection," M.S. thesis, Univ. of California, Santa Barbara, 1992.

[6] William R. Cheswick, Steven M. Bellovin, "Firewalls and Internet Security", ISBN:0-201-63357-4, 1994.

[7] D. E. Denning, "An Intrusion Detection Model", IEEE Transactions Software Engineering, 1987.

[8] 은유진, 박정호, "침입 탐지 기술 분류 및 기술적 구성요소", KISA Information Security News, Vol.13, 1998.

[9] T. F. Lunt, "A survey of intrusion detection techniques," Computers & Security, vol. 12, no. 4, pp. 405-418, 1993.

[10] Crosbie M, Spafford E, "Applying Genetic Programming to Intrusion Detection", Technical Report, Purdue University, Department of Computer Science, 1995.

[11] Crosbie M, Spafford E, "Defending a Computer System using Autonomous Agents," Technical Report, Purdue University, Department of Computer Science, 1994.

[12] Crosbie M, Spafford E, "Active Defense of a Computer System using Autonomous Agents," Technical Report, Purdue University, Department of Computer Science, 1995.

[13] 이 경하 외, "네트워크 패킷 정보를 기반으로 한 보안 관리", 한국정보과학회 논문지, Vol.25, 1998.

[14] SRI International, http://www.sdl.sri.com/projects/nides/

[15] Eric Bloedorn, Alan D. Christi- ansen, 외 "Data Mining for Network Intrusion Detection: How to Get Started," The MITRE Corporation, In http://www.afcea.org/pastevents/db2001/Bloedorn_ files/frame.htm, 2001.

[16] P.A. Porras and P.G. Neumann, "EMERALD : Event Monitoring Enabling Responses to Anomalous LiveDisturbance," Proceedings of the National Information Sys tems S ecurity Conf erence, pp.353- 365, 1997.

[17] CISCO, "NetRanger Intrusion Detection System", Technical Information, 1999.

[18] Reto Baumann, "White Paper : Honeypots", http://security.rbaumann.net/, 2002.

[19] Honeynet Project Members, "Know Your Enemy : Honeynets", http://www.honeynet.org/papers/honeynet/, 2002.

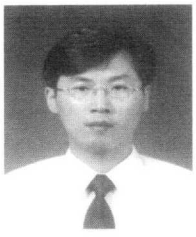
[20] ManTrap : A Secure Deception System, Recuouse Technologies, Inc.2001.

[21] Albert Gonzalez, http://www.violating.us/projects/baitnswitch, Bait & Switch Honeypot

[22] TcpDump, www.tcpdump.org/

[23] Nmap, http://www.insecure.org/n- map/

이 명 섭(Myung Sub Lee) 정회원



e-mail:skydream@cse.yu.ac.kr
1998년 2월 : 경일대학교 컴퓨터공학 졸업 (공학사)
2000년 2월 : 영남대학교 대학원 컴퓨터공학과 졸업 (공학석사)
2003년 8월 : 영남대학교대학원 컴퓨터공학과 졸업(공학박사)
2003년 3월 - 현재 : 영남대학교 전자정보공학부 객원교수

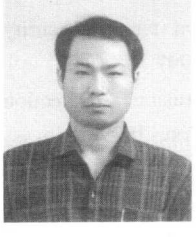
<주관심 분야> 망관리 시스템, 데이타 마인닝, 에이전트, QoS(Quality of Service)

신 경 철(Kyung Chul Shin) 학생회원

e-mail:starmaro@hotmail.com
2000년 2월 : 영남대학교 컴퓨터공학 졸업 (공학사)
2004년 2월 : 영남대학교 대학원 컴퓨터공학과 졸업(공학석사)

<주관심 분야> 지능형 망관리 시스템, 에이전트, IDS, QoS(Quality of Service)

박 창 현(Chang Hyeon Park) 정회원



e-mil: park@cse.yu.ac.kr
1986년 : 경북대학교 전자공학과 졸업 (공학사)
1988년 : 서울대학교 계산통계학과 전산학전공 (이학석사)
1992년 : 서울대학교 계산통계학과 전산학전공(이학박사)
1992년 - 1993년 : 서울대학교 컴퓨터신기술공동연구소 특별연구원
1998년 - 1999년 : University of Maryland, Institute of Advanced Computer Systems, Visiting Researcher
1993-현재 : 영남대학교 컴퓨터공학과 부교수

<주관심 분야> 인공지능, 데이타 마인닝, 에이전트, 지능형 망관리 시스템