유출트래픽 분석기반의 침입탐지시스템 설계 및 구현

Design and Implementation of an Intrusion Detection System based on Outflow Traffic Analysis

신동진*, 양해술**
서울벤처정보대학원대학교 컴퓨터응용기술학과*, 호서대학교 벤처전문대학원**

Dong-Jin Shin(djshin@hoseo.or.kr)*, Hae-Sool Yang(hsyang@hoseo.edu)**

요약

현재 일반화되어 있는 침입탐지 시스템의 경우 중요한 서버의 보안에 유용한 호스트기반 IDS는 합법적인 사용자의 불법행위를 모니터링 가능하고 운영체계와 밀접히 결합하여 보다 정교한 모니터링, 네트워크 환경과 상관없이 사용가능하다는 장점이 있지만 비용의 증가와 침입탐지를 위한 처리에 해당 시스템 자원 소모, 네트워크 기반의 공격에 취약하며 IDS오류 시 해당 호스트의 기능이 마비될 수 있다. 네트워크기반 IDS는 네트워크 엑세스 지점에만 설치하여 비용절감 및 네트워크 자원에 대한 오버헤드감소, 공격에 노출될 가능성이 낮으며 네트워크 환경에 관계없이 사용가능하지만 대용량의 트래픽 처리에 어려움과 제한된 탐지능력, 알려지지 않은 악성코드나 프로그램에 대처능력이 떨어지는 한계를 가지고 있다. 본 논문에서는 이러한 보안 솔루션들 중에서 개인용 방화벽을 활용하는 데스크톱 보안과 함께 적용하여 개인용 컴퓨터의 보안능력을 향상 시키는 유출 트래픽 분석기반 침입탐지시스템의 설계 및 구현을 목적으로 한다. 침입이 발생하고 새로운 패턴의 악성 프로그램이 정보의 유출을 시도하는 행위를 탐지하여 차단함으로써 컴퓨터나 네트워크의 심각한 손실을 감소시킬 수 있다.

■ 중심어 : | 개인용 컴퓨터 보안 | 침입탐지시스템 | 악성코드 탐지 |

Abstract

An increasing variety of malware, such as worms, spyware and adware, threatens both personal and business computing. Remotely controlled bot networks of compromised systems are growing quickly. This paper proposes an intrusion detection system based outflow traffic analysis. Many research efforts and commercial products have focused on preventing intrusion by filtering known exploits or unknown ones exploiting known vulnerabilities. Complementary to these solutions, the proposed IDS can detect intrusion of unknown new malware before their signatures are widely distributed. The proposed IDS is consists of a outflow detector, user monitor, process monitor and network monitor. To infer user intent, the proposed IDS correlates outbound connections with user-driven input at the process level under the assumption that user intent is implied by user-driven input. As a complement to existing prevention system, proposed IDS decreases the danger of information leak and protects computers and networks from more severe damage.

■ keyword : | Desktop Computer Security | IDS(Instrusion Detection System) | Malware Detection |

* 본 연구는 지식경제부와 정보통신연구진흥원의 대학IT연구센터 지원사업의 연구결과로 수행되었음.
(IITA-2009-(C1090-0902-0032))
접수번호 : #090316-002
접수일자 : 2009년 03월 16일
심사완료일 : 2009년 04월 15일
교신저자 : 신동진, e-mail : djshin@hoseo.or.kr

I. 서 론

오늘날 정보통신기술의 급속한 발전과 초고속 인터넷 보급이 확대되어 전자상거래, 행정, 금융 등 사회 각 분야에 인터넷의 활용이 일반화됨에 따라 이를 위협하는 애드웨어(adware), 스파이웨어(spyware), 웜(worm)과 같은 다양한 악성코드(malware)의 증가로 정보범죄와 정보화에 따르는 각종 역기능 현상을 유발시킴으로써 개인용 컴퓨터를 포함한 비즈니스 컴퓨팅 환경에 위협을 주고 있다[1][13].

빠르게 발전하는 악성 프로그램으로부터 컴퓨터 시스템을 보호하기 위하여 많은 상용화 제품들은 새롭게 발견된 악성 코드의 시그니처(signature)를 데이터베이스화하고 분산된 컴퓨터 시스템들이 감염되거나 침해를 당하지 않도록 시그니처를 적용시키는 방법을 채택하고 있다. 그러나 이러한 방법은 새로운 악의적인 소프트웨어에 대하여 일시적으로 그 기능을 발휘하지 못하는 무방비 상태에 놓일 수 있다[2][5].

특히, 일반화되어 있는 침입탐지 시스템의 경우 중요한 서버의 보안에 유용한 호스트기반 IDS는 합법적인 사용자의 불법행위를 감시하고 네트워크 환경과 상관없이 사용가능하다는 장점이 있지만 비용의 증가와 침입탐지를 위한 처리에 해당 시스템 자원소모, 네트워크 기반의 공격에 취약하며 IDS오류 시 해당 호스트의 기능이 마비될 수 있다. 네트워크기반 IDS는 네트워크 엑세스 지점에만 설치하여 비용점감 및 네트워크 자원에 대한 오버헤드감소, 공격에 노출될 가능성이 낮으며 네트워크 환경에 관계없이 사용가능하지만 대용량의 트래픽 처리에 어려움과 제한된 탐지능력, 알려지지 않은 악성코드에 대처능력이 떨어지는 한계를 가지고 있다[10].

본 논문에서는 이러한 보안 솔루션들 중에서 개인용 방화벽을 활용하는 데스크톱 보안과 함께 적용하여 개인용 컴퓨터의 보안능력을 향상 시키는 유출 트래픽 분석기반 침입탐지시스템의 설계 및 구현을 목적으로 한다. 침입이 발생하고 새로운 패턴의 악성 프로그램이 정보의 유출을 시도하는 행위를 탐지하여 차단함으로써 컴퓨터나 네트워크의 심각한 손실을 감소시킬 수 있다.

II. 악성 프로그램의 유형 분석

개인용 컴퓨터에서의 악성코드는 사용자의 컴퓨터 내에 있는 정보를 유출하고 시스템 자원을 과도하게 소비하며 악성 네트워크 트래픽을 유발하여 전체 네트워크의 성능을 저하시킨다.

최근 5년간 침해사고의 유형은 [그림 1]과 같다[1].

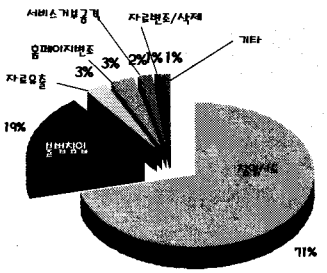

그림 1. 정보보안 침해사고의 유형

Downloader, Xema와 같은 악성 코드는 자체 확산기능은 없이 주로 웹사이트를 통하여 감염되어 확산되며 Virut의 경우 단순히 실행파일을 감염시켜 컴퓨터 사용자에게 불편을 주는 형식이 아닌 악성 코드를 만들어 배포한 공격자의 명령을 전달받아 분산서비스거부(DDoS) 공격을 유발하게 된다. 또한 이러한 악성 코드들은 바이러스와 악성 Bot의 기능이 결합된 형식으로 설계되어 피해를 확산시킨다. 이러한 악성 코드들은 자신의 복제 여부와 존재 유형에 따라 크게 [그림 2]와 같이 분류하며 동작원리를 유형별로 정리하면 다음과 같다[12].

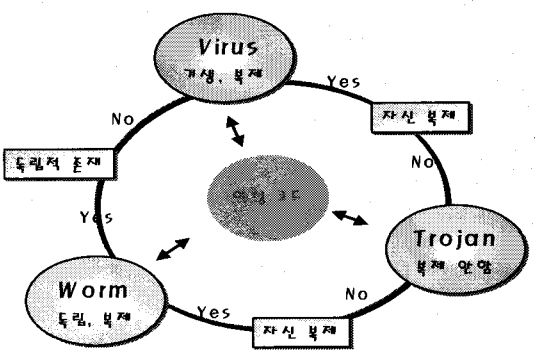

그림 2. 악성코드의 분류

Ⅲ. 악성코드의 탐지기법

악성코드는 파일 감염을 목적으로 하는 바이러스(virus)로부터 네트워크를 통한 급속한 확산을 시도하는 웜(worm) 그리고 데이터 유출을 위한 트로이목마(trojan horse)에 이르기까지 다양한 모습으로 발전하였다. 이러한 악성 코드의 위협은 해가 갈수록 증가 추세를 이루고 있으며 기술적인 면에서도 고지능성을 더해 가고 있어 컴퓨터 사용자들에게 많은 부작용을 유발시키는 원인을 제공하고 있다. 그러나 이렇게 증가하는 악성 코드의 위협에 대응하는 안티 바이러스에 대한 연구 역시 새로운 악성 코드의 위협들로부터 컴퓨터 시스템을 보호하기 위해 다양한 대응 방안들이 활발하게 연구되고 있다[10][12].

악성코드의 침입탐지 기법은 크게 2가지의 모델 연구를 중심으로 하는데 첫째는 침입에 관한 축적된 지식을 사용하여 어떤 침입 방법을 사용하고 있다는 증거에 중점을 두는 모델과, 두 번째는 모니터링 중인 시스템의 정상행위에 관한 참조모델을 생성한 후 정상행위에서 벗어나는 경우를 찾는 방식이다.

표 1. 침입탐지 모델에 따른 연구모델

침입탐지 모델	연구모델
지식기반	∴ 전문가시스템(expert system) ∴ 시그니처 분석(signature analysis) ∴ 페트리넷(Petri-net) ∴ 상태전이분석(state transition analysis) ∴ 신경망(neural network) ∴ 유전 알고리즘(genetic algorithm)
행위기반	∴ 통계적(statistical) 방법 ∴ 전문가시스템(expert system) ∴ 신경망(neural network) ∴ 컴퓨터 면역학(computer immunology) ∴ 데이터마이닝(data mining) ∴ HMM(Hidden Markov Model) ∴ 기계학습(machine learning)

지식기반 침입탐지(오용탐지) 방법은 알려진 침입행위를 이용하여 침입을 탐지하고, 정해진 모델과 일치하는 경우를 침입으로 간주하며, 행위기반 침입탐지(비정상행위 탐지) 방법은 사용자의 패턴을 분석한 후, 입력패턴과 비교하여 침입을 탐지하는 방법이다.

[표 1]에 침입탐지 모델에 따른 각각의 연구모델 현황 표시하였다[1][13].

이러한 다양한 대응 방안들의 기본 전제는 급속하게 확산되는 알려지지 않은 새로운 악성 코드에 대한 침입탐지와 침입차단에 대한 대응체제의 구축이며 새로운 악성 코드를 어떻게 효과적으로 탐지할 것인가에 중점을 두고 있다.

Ⅳ. 유출트래픽 분석기반의 침입탐지시스템 설계

악성코드의 침입에 대한 대응 형태는 수동적 대응과 능동적 대응으로 구분된다. 대부분의 침입탐지 알고리즘은 수동적으로 공격이 발견되면 경보를 발생하거나 대응 소프트웨어가 개발되어 배포되는 체제를 유지하고 있다[4][5].

그러나 [그림 3]에서 보는바와 같이 악성코드에 대한 대응체계를 아무리 신속하게 결정해도 수동적 대응체계에서는 악성 코드에 감염되는 피해를 반드시 수반하게 된다.

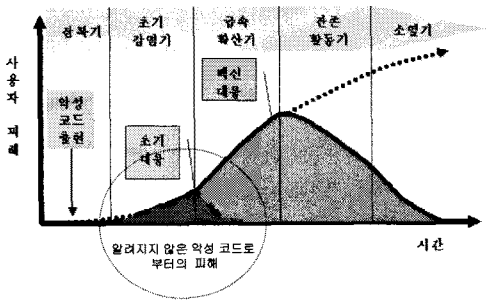

그림 3. 대응시점별 악성코드에 의한 피해

빠르게 발전하는 악성 코드로부터 컴퓨터 시스템을 보호하기 위한 침입탐지 알고리즘은 공통적으로 2가지의 해결방법을 요구한다.

첫째, 새로운 악성코드가 발견되면 그 특징을 기반으로 시그니처 데이터베이스를 생성한다. 둘째, 분산된 컴퓨터 시스템들은 이러한 시그니처를 침입에 앞서 로컬(local) 데이터베이스에 다운로드하여 새로운 악성코드에 대응하는 침입탐지 환경으로 수정 적용하여야 한다.

그러나 이러한 과정을 철저하게 수행하더라도 새로

운 악성코드에 대해서는 일시적으로 무방비 상태에 놓일 수 있다. 특히, 웜(worm)의 경우에는 시그니처를 생성하고 분배하는 것 보다 더욱 빠르게 증식되기 때문이다[6].

따라서 알려지지 않은 악성코드에 대응하는 침입탐지 알고리즘은 침입이 발생한 후에 악성코드의 침입을 신속하게 자동적으로 탐지하는 구조에 중점을 두고 컴퓨터 시스템상의 손실을 완화시키는 구현방식의 실현이 필요하며 본 논문에서는 개인용 컴퓨터에 침입이 발생한 후에 새로운 악성코드의 침입을 신속하게 자동적으로 탐지하는 구조에 중점을 둔 유출트래픽 분석기반의 침입탐지 시스템을 제안하고 설계한다.

제안된 시스템은 침입을 탐지하기 위하여 내부 네트워크로 유입되는 트래픽을 모니터링하지 않는다. 내부 네트워크로 유입되는 트래픽에 대한 침입탐지는 방화벽(firewall), 호스트 기반이나 네트워크 기반의 침입방지시스템(IPS)이나 침입탐지시스템(IDS)에 의하여 실현된다고 전제하며, 악성 코드는 침입 후 필연적으로 컴퓨터 사용자의 정보를 탈취하기 위하여 사용자의 의도와는 관계없는 프로세스를 생성하거나 악성 코드가 침입에 앞서 설정한 외부 목적지를 향한 네트워크 트래픽을 발생시키게 된다. 따라서 제안된 시스템은 이러한 악성 코드의 특성을 컴퓨터 사용자의 정상적인 시스템 사용 행위와 비교하여 악성 코드인지를 판단하게 된다.

사용자 시스템에 악성코드의 침입이 발생한 후 사용자가 의도하지 않은 정보 유출(outflow)을 탐지하는 것으로 침입탐지를 실현하는 제안 시스템은 [그림 4]와 같은 구조를 갖는다.

제안 시스템은 사용자 레벨(user level)과 커널 레벨(kernel level)의 구조로 설계되었으며 침입탐지 엔진(IDS Engin)으로 사용자 모니터(user monitor), 프로세스 모니터(process monitor), 네트워크 모니터(network monitor) 그리고 유출 탐지 모듈(outflow detector)로 구성된다. 사용자 허가 목록(UPL : User Permission List)은 외부 네트워크로 접속이 허용된 시스템이나 응용 프로그램에 대한 목록을 관리하는 데이터베이스로 사용자에 의해 관리되는 구조를 갖는다.

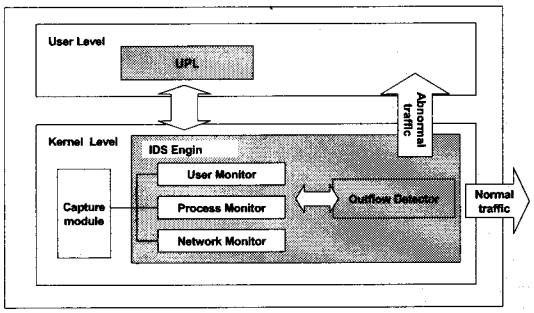

그림 4. 제안된 침입탐지 시스템의 구조

제안된 시스템은 개인용 컴퓨터의 새로운 알려지지 않은 악성 코드의 침입을 탐지하고 효율성과 타당성을 제고하기 위하여 악성 코드에 대한 시그니처 데이터베이스를 갖지 않으며 악성 코드의 유형과 알려져 있는 여부에 관계없이 동작되도록 설계한다.

V. 기능 블록별 알고리즘 설계

[그림 5]는 제안 시스템이 사용자가 의도하지 않은 정보 유출(outflow) 행위를 탐지하여 침입에 대한 판단을 결정하는 과정을 표현한 것이다.

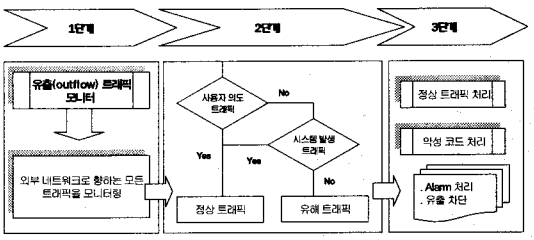

그림 5. 정보 유출 행위기반 침입탐지 절차

제안 시스템은 1단계에서 외부 네트워크로의 접속을 시도하는 트래픽을 감시하고 2단계에서 외부 네트워크로의 접속이 사용자가 의도한 능동적인 접속인지, 사용자가 의도하지는 않았지만 시스템의 사용이나 능동적인 접속 후에 발생하는 간접적인 접속인지, 악성 코드에 의한 사용자 비의도적 접속인지를 판단한다. 3단계에서는 정상적인 행위로 판단된 트래픽은 정상 트래픽 처리를 수행하고, 악성 코드로 판단된 트래픽은 경보를

발생하고 유출을 차단하는 처리과정을 수행한다.

침입을 탐지하기 위한 사용자 의도 접속유형을 정의하면 [표 2]와 같이 3가지 유형으로 정의된다.

표 2. 접속유형 모델링

접속유형 정의	외부 네트워크로의 접속이 발생하는 경우
사용자 행위	의도한 사용자의 입력
프로세스 행위	프로세스의 시작이나 종료
네트워크 행위	접속요청, 데이터의 수신, 네트워크 운영 (DNSlookup 등)

∴ 사용자의 입력으로 인한 요청한 데이터의 수신이 발생하면 동일한 프로세스내에서 외부 네트워크로의 새로운 연결이 생성되는 것은 정상적인 트래픽으로 처리된다.

∴ 대부분의 응용 프로그램에서 외부 네트워크로 접속을 시도하는 경우 사용자가 의도하는 접속은 [그림 6]과 같은 방식으로 실행된다.

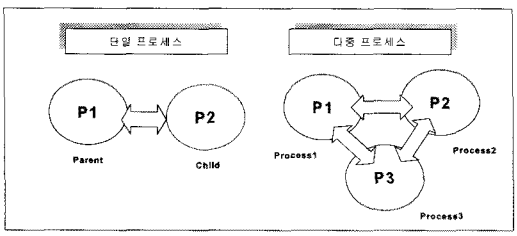

그림 6. 프로세스 환경에 의한 외부 네트워크 접속

단일 프로세스의 경우 외부 네트워크 접속시도는 동일한 프로세스의 연결 또는 부모(parent) 프로세스로부터 역할을 위임받은 자식(child) 프로세스의 기능(데이터의 수신, 통신 포트(port)의 감시, 사용자 입력처리 등)실행이 실행되므로 프로세스의 관계를 확인하여 정상적인 트래픽으로 처리한다.

다중 프로세스 경우는 프로세스 각각이 독립적으로 실행되기 때문에 정상적인 트래픽의 여부를 판단하기 위해서는 개별 프로세스 전체에 대한 모든 가능한 네트워크 동작을 감시해야 한다. 본 논문에서는 단일 프로세스(parent-child) 환경하의 통신만 고려하여 설계한다.

유출 탐지 설계는 [그림 5]의 2단계와 같이 외부 네트워크로의 접속이 사용자가 의도한 것인지 또는 시스템 사용자에 의한 허가된 접속인지를 결정한다. 시스템 또는 사용자가 허가한 접속은 각각의 운영체제와 사용자 허가 목록(UPL:User Permission List)의 정의로 구현한다. 이러한 사용자 허가 목록은 [표 3]과 같이 3가지 유형으로 정의된다.

표 3. 사용자 허가 목록 유형정의

허가목록 유형	사 용 예
시스템 데몬 (system demon)	운영체제의 네트워크 운영(관리) 프로그램
응용 프로그램 (application program)	자동적으로 수정(update)을 점검하는 프로그램
네트워크 응용 프로그램 (network application program)	자동적으로 시작되는 네트워크 응용 프로그램

본 논문의 실험평가를 위한 윈도우즈 환경하의 시스템 데몬, 업데이트 응용 프로그램, 네트워크 응용프로그램에 대한 사용자 허가목록은 [표 4]와 같이 정의한다.

표 4. 사용자 허가 목록에 대한 프로그램 정의

허가목록 유형	이미지(image) 이름	사용자(user) 이름
시스템 데몬	System	SYSTEM
	spoolsv.exe	SYSTEM
	svchost.exe	NETWORK SERVICE
	services.exe	NETWORK SERVICE
응용프로그램	프로세스 또는 이미지 이름	
	Microsoft windows & office update	
	Java update	
	Real Player	
네트워크 응용 프로그램	Windows Messenger	
	myLinker	

사용자 허가목록에 정의된 유형에 대한 웹사이트 및 외부 네트워크로의 접속은 사용자 및 시스템 발생 트래픽으로 판단하여 항상 접속을 허용하는 정상 트래픽으로 처리한다.

[그림 7]은 유출탐지의 과정을 DFD(Data Flow Diagram)로 표현한 것이다.

데이터 패킷의 도착이 발생하면 패킷을 분석하여

TCP와 UDP 각각의 트랜잭션 패킷으로 분리하여 TCP와 UDP 헤더(header)를 참조하여 해당 IP패킷의 출발지(source)를 판단하기 위한 리어셈블(reassemble)과정을 거친 후 유출탐지를 판단하기 위한 처리과정을 거친다. 유출탐지처리는 사용자 허가목록(UPL)을 참조하여 유출 트래픽 처리시에 수록한 사용자 허가목록을 포함한 사용자 의도접속 여부를 결정한다.

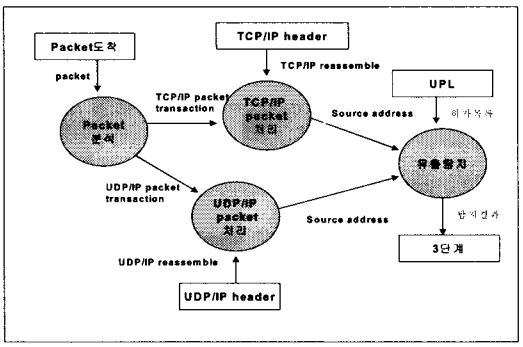

그림 7. 유출탐지의 자료흐름도(DFD)

웜은 감염대상에 따라 이메일 웜과 같이 사용자가 활성화의 매체가 되던지 스스로 활성화되는 유형으로 분류할 수 있다. 스스로 활성화되는 경우라도 개인용 컴퓨터를 감염시키기 위해서는 사용자의 입력을 요구하게 된다. 활성화된 악성코드는 백그라운드(background) 프로세스로 실행되어 어떠한 사용자의 입력도 받지 않고 해당 컴퓨터 시스템의 정보자산을 유출하게 된다. 따라서 이러한 악성코드의 특성을 이용하여 사용자의 입력을 요구하지 않는 악성코드에 대한 침입탐지를 위해서 [표 5]와 같이 정상적인 프로세스와의 구분을 정의하고 해당 알고리즘을 적용한다.

표 5. 백그라운드 프로세스의 특성

비교항목	정상적인 프로세스	악성코드 프로세스
유 형	백그라운드 프로세스	백그라운드 프로세스
사용자 입력(UI)	요구	요구하지 않음
프로세스간 연결지연(PD)	발생	발생하지 않음

[표 5]와 같이 정상적인 프로세스는 사용자의 접속요구가 반드시 필요하며 악성코드는 사용자의 입력을 요구하지 않는다. 또한 접속유형의 모델링에서 정의한 바와 같이 정상적인 프로세스의 활동에서는 마지막 사용자의 입력이나 데이터의 수신시 프로세스간의 연결지연이 발생하게 되는데 악성코드는 이러한 연결지연이 발생하지 않는다. 따라서 프로세스의 특성에 따라 연결지연을 점검하여 침입을 탐지한다. 프로세스의 연결지연에 대한 유형정의는 [표 6]과 같다.

표 6. 프로세스의 연결지연에 대한 유형 정의

프로세스 유형	유 형 정 의
PDa	PDa 유형의 프로세스는 부모(parent)-자식(child)간의 관계를 갖는 다중 프로세스 환경하의 구조로 자식 프로세스가 생성되기 전에 해당 자식 프로세스의 부모 프로세스에 의해 요청된 데이터를 수신하는데 발생하는 지연시간으로 보통 프로세스가 실행되는데 필요한 적재(load)시간을 포함하며 소요지연 시간은 컴퓨터의 성능에 따라 차이가 있지만 사용자가 의도한 접속에 대하여 초(seconds)단위의 지연시간을 소요한다.
PDb	PDb 유형의 프로세스는 단일 프로세스 환경으로 해당 프로세스에 의해 요청된 데이터를 수신하는데 발생하는 지연시간으로 프로세스의 반응시간이며 PDa 유형의 프로세스와 마찬가지로 사용자가 의도한 접속에 대하여 초(seconds)단위의 지연시간을 소요한다.
PDc	PDc 유형의 프로세스는 클라이언트-서버 환경의 이메일 수신과 같이 동일한 IP또는 호스트 주소로의 마지막 접속요청 후 현재까지 데이터를 수신하는데 발생하는 지연시간으로 사용자가 의도한 접속에 대하여 분(minutes) 단위의 지연시간을 소요한다[6][9]. 따라서 접속요청에 의한 사용자 입력이나 프로세스간 연결지연이 발생하지 않는 프로세스는 악성코드로 판단하여 침입을 탐지한다.

[그림 8]은 발생한 접속요청에 대하여 악성코드에 대한 탐지절차 흐름도이다.

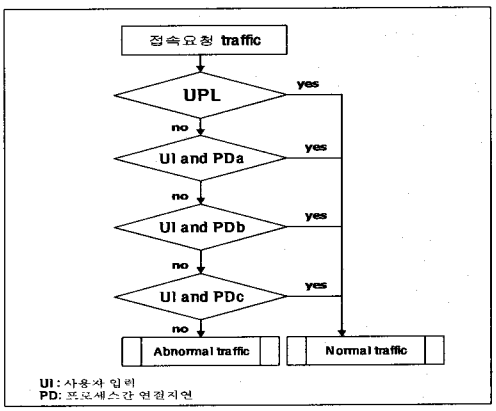

그림 8. 침입탐지 절차 흐름도

Ⅵ. 침입탐지 기능 블록 설계

유출탐지로 침입을 탐지하는 유출탐지(outflow detector) 기능 블록을 포함하여 [그림 4]의 침입탐지(IDS) 엔진은 사용자 모니터(user monitor), 프로세스 모니터(process monitor), 네트워크 모니터(network monitor)의 기능 블록으로 구성된다. 각각의 기능 블록들은 독립적으로 구성되며 유출탐지를 위해 유출탐지(outflow detector) 모듈로 실시간 정보를 제공하기 때문에 상호 교차적으로 설계한다.

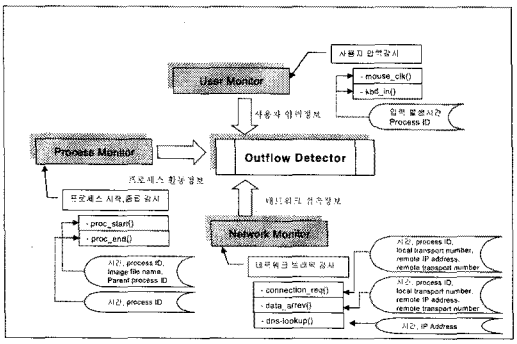

그림 9. 침입탐지 기능블록 설계

[그림 9]는 [그림 4]의 침입탐지(IDS) 엔진을 기능 블록별로 설계한 것이다.

* 사용자 모니터(user monitor)

사용자 입력을 감시하는 기능 블록으로 유출 탐지기(outflow detector)에 사용자가 마우스(mouse)나 키보드로 입력하는 접속요청을 mouse_clk()함수와 kbd_in() 함수로 모니터링하여 접속요청이 발생한 시간과 입력을 받은 프로세스 ID(IDentification)를 전달한다. 유출탐지기는 어떤 프로세스가 사용자 의도접속을 요청했는지에 대한 판단을 하고 접속요청으로부터 데이터를 수신할 때까지 지연시간을 계산하는 인자(argument)로 사용한다. 사용자 입력과 프로세스간의 이러한 연관은 운영체제에서 제공하는 프로세스관리 기능을 이용하여 프로세스간의 복잡도를 증가시키지 않는다.

* 프로세스 모니터(process monitor)

프로세스의 시작과 종료를 감시하는 기능블록으로 유출 탐지기에 프로세스 활동정보를 전달한다. 프로세스 활동정보는 프로세스의 시작을 모니터링하는 함수(proc_start())는 프로세스의 시작시간, 프로세스 자신의 ID 그리고 이미지(image) 파일이름(프로그램이 주기억장치에 적재(load)되어 중앙처리장치(cpu)가 실행하게 되면 프로그램 파일이름과 프로세스 이름이 구분), 부모(parent) 프로세스의 ID를 전달하며, 종료를 모니터링하는 함수(proc_end())는 프로세스의 종료시간과 해당 프로세스의 ID를 유출 탐지기에 전달한다.

* 네트워크 모니터(network monitor)

네트워크 트래픽을 감시하는 기능블록으로 유출 탐지기에 유출탐지를 위하여 네트워크 트래픽 정보를 전달한다. 네트워크 트래픽 정보는 접속요청(connection_req()), 데이터의 수신(data_arrev()), 도메인(domain) 이름 검색(dns_lookup()) 함수로 구성된다. 대부분의 응용 프로그램들은 TCP와 UDP 프로토콜을 사용한다. TCP를 통하여 데이터를 교환하기 위해서는 연결을 초기화하는데 TCP 플래그(flag)를 이용한다. 신뢰성을 보장하는 연결설정을 위해서 제공되는 TCP Sync 패킷은 외부로의 네트워크 접속시에 TCP 접속을 위한 3단계 교환(3-way handshaking) 중 1단계에 해당하기 때문에 모든 신뢰성이 제공되는 접속을 네트워크 모니터는 검사하게 된다. 접속요청 함수는 접속요구 시간, 프로세스 ID, 지역전송 포트번호(local transport number), 원격 IP주소(remote IP address), 원격전송 포트번호(remote transport number)와 같은 구성요소를 유출 탐지기에 전달한다. 이때 접속요구 시간은 자신에 대한 연결에 관계없이 그 요청에 대한 DNS 검색에 대한 시작 시간이 된다.

데이터의 수신(data_arrev()) 함수는 정상적인 외부 접속으로부터 수신한 TCP나 UDP 패킷의 비어있지 않은 적재(payload)상태를 확인하는 것으로 점검하여 유출 탐지기에 전달하며, 접속방향은 첫 번째 TCP Sync 패킷과 UDP 패킷의 방향에 의해 결정된다.

데이터의 수신(data_arrev()) 함수는 데이터 패킷이

수신되었을 때 시간이라는 점을 제외하면 데이터 패킷 수신시간, 프로세스 ID, 지역전송 포트번호(local transport number), 원격 IP주소(remote IP address), 원격전송 포트번호(remote transport number)와 같은 구성요소를 가지며 유출 탐지기에 전달한다. 도메인(domain) 이름 검색(dns_lookup()) 함수는 DNS lookup 패킷을 구문분석(parse)하고, DNS 검색 다음에 뒤따르는 접속요구에 대한 성공적인 DNS 검색은 도메인(domain) 이름에 해당하는 동일한 원격 IP 주소들 돌려받는다.

dns_lookup() 함수는 이러한 성공적인 DNS 검색에 따른 검색시간과 도메인 이름에 대한 IP 주소를 매핑(mapping)시키는 테이블을 구성하여 유출 탐지기에 전달한다. DNS 검색은 사용자 입력과 그에 따른 접속요구 사이에 초(seconds)단위의 지연시간을 갖기 때문에 접속요청에 의한 사용자 입력이나 프로세스간 연결지연이 발생하는 문제점은 유출탐지 상에서 예외처리(exception handling)하지 않아도 유출탐지 과정에서 부작용(side-effect)이 발생하지 않는다. dns_lookup() 함수는 도메인 이름 검색시간, IP주소목록과 같은 구성요소를 가지며 유출 탐지기에 전달한다.

Ⅶ. 성능실험 및 결과

본 논문에서 제안한 유출 트래픽 분석기반 침입탐지 시스템은 Windows XP상에서 기반 프레임워크로 뛰어난 객체지향 프레임워크인 비주얼 컴포넌트 라이브러리(VCL)를 사용하는 델파이(Delphi)로 구현하였으며, 알려지지 않은 악성 코드에 대한 유출탐지 기능에 대한 성능 비교 실험을 위해 악성 코드의 유형별 동작특성을 감안하여 대표적인 악성 코드를 분류하고 기존 상용화된 소프트웨어(안티바이러스 소프트웨어)와의 기능을 비교하였다. [표 7]과 같이 실험환경을 1,2단계로 구분한 것은 시그니처 방식의 상용화된 안티바이러스 소프트웨어를 해당 악성코드가 발표된 연도보다 이전 버전으로 구성함으로 새로운 악성코드에 대응하지 못하는 것을 실험하였다.

표 7. 성능실험 구성 및 기준

단계	실험환경 구성	실험 기준
1	악성코드의 출연 년도 < 안티바이러스 소프트웨어 제작(업데이트) 년도	알려진 악성 코드 (K/M : Known Malware)
2	악성코드의 출연 년도 > 안티바이러스 소프트웨어 제작(업데이트) 년도	알려지지 않은 악성 코드 (U/M : Unknown Malware)

많은 상용화 제품들은 분산된 컴퓨터 시스템들이 감염되거나 침해를 당하지 않도록 시그니처를 적용시키는 방법을 채택하고 있기 때문에 새로운 악성 코드의 출연은 기존의 안티바이러스 소프트웨어로 탐지할 수 없는 실험의 타당성을 위하여 실험환경을 구분하였다.

표 8. 실험평가를 위한 악성 코드의 유형

악성 코드 분류	악성 코드 유형	대표적인 악성 코드
A group	이메일 유형	Win32.Nimda I-Worm.Win32.Bagle I-Worm.Win32.Netsky
B group	P2P 유형	I-Worm.Win32.Mydoom I-Worm.Win32.GnuMan Worm.Win32.Deadhat.55808
C group	IRC 유형	I-Worm.Win32.Swen I-Worm.Win32.Merkur Backdoor.Win32.SdBot
D group	MS network	I-Worm.Win32.Agobot I-Worm.Win32.Lovgate Backdoor.Win32.SdBot

[표 8]은 제안된 시스템의 실험평가를 위해 악성 코드의 유형별로 대표적인 악성 코드를 3가지씩 지정하여 분류한 결과이며, 실험을 위해 대상 컴퓨터(Intel Core2 Duo 2.4GHz Processor, 4GB 메모리, Windows XP Professional Service Pack 3)에 악성 코드를 감염시킨다. 실험을 위한 상용화된 안티바이러스 소프트웨어에 의한 악성 코드의 감염 결과는 [그림 10]과 같다.

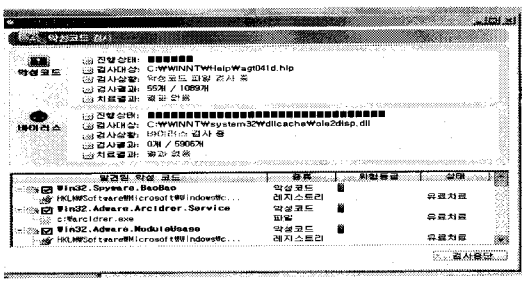

그림 10. 악성 코드의 감염실험

[표 8]의 악성 코드에 의한 사용자 비의도 접속을 탐지한 결과로 제안시스템(ODS)의 UI 화면은 [그림 12]와 같이 발생일시, 프로토콜, 해당 프로세스에 대한 정보로 구성되며, 사용자 비의도 접속에 대한 탐지를 위해 기본적으로 시스템에서 사용하는 프로세스들을 초기 UPL 목록에 등록하는 절차는 UPL Update 메뉴에 의하여 [그림 11]과 같이 실행하였다.

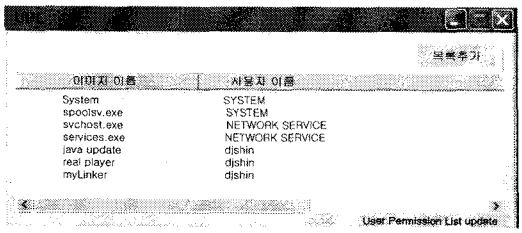

그림 11. 제안 시스템의 UPL목록 등록

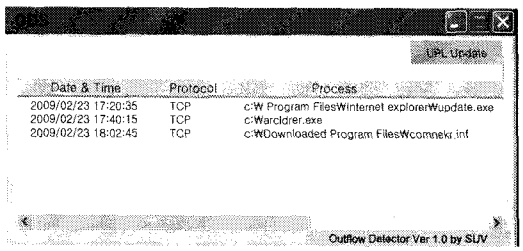

그림 12. 제안 시스템의 사용자 비의도 접속탐지

[표 7]의 단계와 절차 [표 8]의 악성 코드를 대상으로 한 안티바이러스 소프트웨어와 제안시스템의 실험결과를 정리하면 [표 9]와 같다. 제안 시스템은 알려지지 않은 악성 코드에 의해 사용자의 시스템이 감염되었다는 전제하에 사용자 비의도 접속을 식별하는 것으로 침입에 대한 탐지를 실현하기 때문에 기존의 안티 바이러스 제품과 비교하는 직접적인 성능 실험에 부가하여 제안 시스템의 특성을 중심으로 결과를 평가하였다.

표 9. 실험평가 결과

실험 대상	알려진 악성코드(K/M)		알려지지 않은 악성코드(U/M)	
	결과	감염 후 동작	결과	감염 후 동작
제안시스템	▣	비의도 접속 탐지	▣	비의도 접속 탐지
제품 A	○	탐지 후 삭제	▣	탐지 지속
제품 B	○	탐지 후 삭제	▣	탐지 지속
제품 C	○	탐지 후 삭제	▣	탐지 지속
제품 D	○	탐지 후 삭제	▣	탐지 지속
제품 E	○	탐지 후 삭제	△	탐지 후 삭제
○ : 탐지, △ 부분탐지, ▣ 미탐지				

실험결과와 같이 기존의 안티 바이러스 제품은 알려진(악성코드의 출연 년도 < 안티바이러스 소프트웨어 제작(업데이트) 년도) 악성코드에 대하여 탐지 후 사용자의 확인과정 후 삭제를 실행하였으며, 알려지지 않은(악성코드의 출연 년도 > 안티바이러스 소프트웨어 제작(업데이트) 년도) 악성코드에 대해서는 프로세스 실행 상태로 탐지를 지속하였다.(제품 E의 경우 알려지지 않은 악성 코드의 일부에 탐지 후 삭제를 실행하였는데 이는 악성 코드의 유형이 같은 그룹에 제한적으로 적용된 결과를 보여 주었다.) 제안 시스템은 설계 목표와 같이 감염 후 유출 트래픽을 기준으로 악성 코드의 알려져 있는 상태 유무에 관계없이 사용자 비의도 접속 탐지를 수행하였다.

Ⅷ. 결 론

본 논문은 보안 솔루션들 중에서 개인용 방화벽을 활용하는 데스크톱 보안과 함께 적용하여 개인용 컴퓨터의 보안능력을 향상 시키는 유출 트래픽 분석기반 침입탐지시스템의 설계 및 구현을 목표로 제안되었다. 이를 위해 기반 기술의 소개와 전체 시스템 설계 및 각 구성 모듈의 기능과 역할에 대해서 정의하였고, 유출탐지와 침입탐지를 위한 기능 블록별 알고리즘을 정의하였으며, 설계를 바탕으로 유출 트래픽 분석기반 침입탐지시스템을 구현하였다.

유출 트래픽의 사용자 의도 접속에 대한 효율적인 관리를 위해 사용자 레벨(level)의 UPL(User Permission List)관리와 침입탐지에 대한 과부하(overload)를 최소화하기 위하여 커널(kernel) 레벨로 분리 설계하였다. 유출 트래픽 분석기반 침입탐지 시스템이 알려지지 않은 새로운 악성코드에 대한 탐지가 가능해야 하기 때문에 기존의 유형별 대표적인 악성코드를 대상으로 악성코드가 출연하기 전후의 안티 바이러스 소프트웨어 버

전(version)을 기준으로 실험을 진행하였다.

제안된 시스템은 알려지지 않은 악성코드에 감염된 후 사용자가 의도하지 않은 접속을 탐지하는 것으로 침입을 탐지한다. 특히 기존의 시그니처 방식은 관리해야 할 패턴(n개 기준)이 증가하면 최소 ($\log_2$ n)번에서 최대 n번의 검색시간이 소요되지만 제안 시스템은 외부 네트워크로 접속을 요청하는 트래픽의 수에 고정된 검색시간으로 침입을 탐지한다.

향후 연구 과제로는 제안 시스템을 기반으로 유출 트래픽 분석의 부하(load)를 줄이고 다양한 다중 프로세스 환경하에서도 유출분석을 통하여 침입탐지가 가능한 시스템을 설계하는 연구가 필요하다.

참 고 문 헌

[1] 한국정보보호진흥원, *2007-정보보호 실태조사*, 한국정보보호진흥원, 1997.

[2] E. Carl, S. Eugene, and M. Jim, *Intrusion Detection & Prevention*, McGraw -Hill, 2004.

[3] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format," IETF Internet Draft, draft-ietf-idwg-idmef-xm, pp.1-14, 2005.

[4] K. McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets : MIB-II," RFC1213, 1991.

[5] C. Frederic and M. Alexander, "Alert Correlation in a Cooperative Intrusion Detection Framework," IEEE Symposium on Security and Privacy, 2002.

[6] W. Lee and S. Stolfo, *A framework for constructing features and models for intrusion detection systems*, ACM Transactions on Information and System Security, 2000.

[7] http://www.microsoft.com/technet/security/prodtech/win2000/secwin2k/09detect.mspx

[8] H. Gilbert, *LAN Management with SNMP and RMON*, John Wiley & Sons, 1996.

[9] J. Anthony, *Network Programming for Microsoft Windows - 2nd Edition*, 정보문화사, 2002.

[10] http://www.symantec.com

[11] IETF, "A Simple Network Management Protocol," RFC 1157.

[12] http://kr.ahnlab.com/b2b/securityinfo/html/renew_asec_report/200903.jsp?site=b2c

[13] http://www.kisa.or.kr/index.jsp

[14] http://www.ad-spider.com/spyware

[15] http://www.nprotect.com/v6/service/index.php? mode=service_avs

저 자 소 개

신 동 진(Dong-Jin Shin) 정회원

- 1989년 : 서울산업대학교 전자계산학과(학사)
- 1991년 : 한양대학교 산업대학원 전자계산학과(석사)
- 2006년 ~ 2008년 : 서울벤처정보대학원대학교 컴퓨터응용기술학과 박사과정 수료
- 1991년 ~ 1997년 : (주)정원엔시스템 기술연구소(연구원)
- 1997년 ~ 현재 : 서울호서전문학교 사이버해킹보안과 교수

<관심분야> : 정보보호, 해킹 및 바이러스, 침입차단시스템(Firewall), 침입탐지시스템(IDS)

양 해 술(Hae-Sool Yang) 정회원

- 1975년 : 홍익대학교 전기공학과 졸업(학사)
- 1878년 : 성균관대학교 정보처리학과(석사)
- 1991년 : 日本 오사카대학 정보공학과 S/W공학전공(공학박사)
- 1975년 ~ 1979년 : 육군중앙경리단 전산장교
- 1980년 ~ 1995년 : 강원대학교 전자계산학과 교수
- 1986년 ~ 1987년 : 日本 오사카대학교 객원연구원
- 1995년 ~ 2002년 : 한국S/W품질연구소 소장
- 1999년 ~ 현재 : 호서대학교 벤처전문대학원 교수

<관심분야> : 소프트웨어공학(특히, S/W 품질보증과 평가, 품질감리와 컨설팅, 프로젝트관리, CBD기반기술, IT품질경영