# 취약성 데이터베이스에 기반한 분산 웜 탐지 시스템 설계

임정묵*, 한영주*, 정태명**
*성균관대학교 컴퓨터공학과
**성균관대학교 정보통신공학부
e-mail : {izeye, yjhan}@imtl.skku.ac.kr, tmchung@ece.skku.ac.kr

# An Architecture for Vulnerability Database based and Distributed Worm Detection System

Jung-Muk Lim*, Young-Ju Han* and Tai-Myung Chung**
*Dept. of Computer Engineering, SungKyunKwan University
**School of Information Communication Engineering, SungKyunKwan University

**요 약**

인터넷이 생활과 밀접하게 연결되면서 인터넷에 대한 공격이 엄청난 피해를 야기시킬 수 있게 되었다. 웜은 스스로를 복제하여 인터넷 상의 취약한 호스트들을 공격하여 여러가지 피해를 야기시키고 있다. 웜을 탐지하고 방어하기 위해 inbound/outbound 스캔률 검사, 웜 시그니처 검사와 같은 네트워크 기반 침입탐지 방법과 웜 생성 파일 검사, 파일 무결성 검사와 같은 호스트 기반 침입탐지 방법이 제안되었다. 하지만 단일 시스템에서의 웜 탐지는 한계가 있을 뿐만 아니라 대응에 있어서도 더딜 수 밖에 없다. 본 논문에서는 웜 탐지 시스템을 분산 배치시킴으로써 탐지의 정확성을 확보하였고 웜 경보를 모든 웜 탐지 시스템에 송신함으로써 대응에 있어 신속성을 제공해준다. 뿐만 아니라 취약성 데이터베이스를 통해 최신으로 갱신만되어 있다면 제로데이 공격에도 대응할 수 있는 메커니즘을 제공한다.

## 1. 서론

인터넷이 일반화되어 생활과 산업에 밀착되어 발전되고 있을 뿐만 아니라, 정부의 공문서 처리와 발행까지도 인터넷을 통해 가능하게 되었다. 따라서 인터넷이 공격을 받아 서비스를 할 수 없게 된다면 모든 분야에 걸쳐 엄청난 피해를 야기시킬 것이다. 인터넷을 위협하는 공격 중에 가장 광범위하게 영향을 미칠 수 있는 공격은 웜에 의한 공격이다. 웜은 1988 년 모리스 웜을 시작으로 계속적으로 발전하여 최근에 코드레드 웜나 님다 웜과 같이 대규모의 피해를 야기시키는 웜이 등장하고 있고 강력한 피해를 줄 수 있는 웜들의 가능성이 확인되고 있다[1][2][3]. 그에 따라 산업계와 학계 모두가 이를 탐지하고 방어하기 위한 다양한 방법을 모색하고 있다.

웜을 탐지하기 위하여 호스트 기반 침입탐지시스템과 네트워크 기반 침입탐지시스템이 이용된다. 호스트 기반 침입탐지시스템은 주로 웜의 특정 파일 생성, 수정, 삭제와 같은 행위를 모니터링함으로써 탐지하고, 네트워크 기반 침입탐지시스템은 주로 과도한 스캔 행위를 모니터링함으로써 탐지하거나 취약성 공격 코드를 시그니처 기반으로 탐지한다.

하지만 호스트 기반 침입탐지시스템은 메모리 상주형 웜과 같이 파일 시스템에 흔적을 남기지 않는 경우 탐지가 용이하지 않고, 네트워크 기반 침입탐지시스템은 웜이 스캔 속도를 조절하거나 새로운 시그니처를 사용하는 경우 시그니처가 데이터베이스 상에 존재하지 않기 때문에 탐지가 불가능하거나 어렵다.

본 논문에서는 오탐률을 줄이기 위해 기존의 웜 탐지 시스템과 결합하여 취약성 데이터베이스에 기반한 분산 웜 탐지 시스템을 제시한다. 웜의 제로데이가 가까워지고 있다고는 하지만 지금까지의 웜이 이미 발표된 취약성을 이용한다는 사실에 기반하여 웜 탐지

결과와 취약성 정보와의 상관분석을 통해 탐지하는 방식이다. 이 시스템은 최신 취약성 데이터베이스 서버와 이 데이터베이스를 참조하는 분산 배치되어 있는 웜 탐지 시스템들로 구성된다.

논문의 구성은 다음과 같다. 2 장에서는 논문 전체의 배경지식을 제공하기 위하여 웜에 대한 기술적인 기초를 제공하고, 기존의 탐지 방법 및 각 방법의 장단점을 기술한다. 3 장에서는 본 시스템에서 사용될 취약성 데이터베이스에 대해 설명한다. 4 장에서는 취약성 데이터베이스 서버와 웜 탐지 시스템들로 구성된 전체 시스템에 대해 설명한다. 5 장에서는 웜 탐지를 위해 사용되는 알고리즘에 대해 설명한다. 6 장에서는 제시한 시스템의 한계점을 분석한다. 7 장에서는 본 논문의 결론과 함께 앞으로의 연구 방향을 제시한다.

## 2. 관련 연구

### 2.1 웜 개요

웜은 숙주 파일없이도 한 시스템에서 다른 시스템으로 자신을 복제하는 프로그램을 일컫는다[4]. 웜은 숙주 파일을 요구하지 않는다는 점에서 바이러스와 구분되고 자기복제가 가능하다는 점에서 트로이 목마와 구분된다. 하지만 이는 용어적인 구분일 뿐이고, 실제 웜이 전개되어 나가는 과정에서 이러한 구분이 불분명해지는 경우가 많다.

웜의 동작은 크게 목표선정, 취약성공격, 감염, 전파의 네 단계로 구분한다[5][6]. 다음 [그림 1]은 웜의 동작을 단계별로 보여주고 있다.


[그림 1] 웜의 동작 4 단계

목표선정 단계는 감염 호스트가 목표 호스트를 찾는 단계로 일반적으로 특정 서비스의 존재나 취약 유무를 인터넷이나 인트라넷을 대상으로 수행한다. 취약성공격 단계는 특정 서비스의 취약성을 이용하여 해당 호스트의 권한을 획득하는 과정이다. 감염 단계는 웜의 본체 전송을 비롯하여 백도어 설치, 시스템 파일 수정, 시스템 유틸리티를 트로이 목마로 교체함으로써 자신을 은닉하는 등의 부수적인 작업을 수행한다. 전파 단계는 동작의 관점에서 목표선정 단계와 동일하지만, 감염 호스트의 관점에서는 목표선정 단계가 스캔을 당하는 단계인 반면에 전파 단계는 스캔을 수행하는 단계로 서로 판이하게 다르다.

### 2.2 기존의 웜 탐지 방법

웜 탐지는 앞에서 설명한 각 동작 단계별로 다르게 이루어진다. 목표선정 단계에서 웜을 탐지하기 위해서는 네트워크 기반 침입탐지시스템을 설치하여 inbound 트래픽을 모니터링해야만 한다. 이는 목표선정 단계가 취약성을 갖는 서비스를 찾는 단계로서 특정 서비스 포트를 고정시키고 네트워크 내에 다수 개의 호스트를 대상으로 스캔하기 때문에 이러한 행위에 기반하여 웜을 탐지할 수 있다. 하지만 웜이 순차으로 스캔하지 않고 랜덤하게 스캔한다면 이와 같은 방식으로 웜을 탐지할 수 없다. 랜덤하게 들어온 한 개의 스캔만으로도 네트워크 전체를 감염시킬 수 있기 때문에 inbound 트래픽을 모니터링하는 방식만으로는 한계가 있다. 이 방식은 특정 시간 동안 특정 이벤트의 개수에 기반하여 탐지하기 때문에 웜이 스캔 속도를 조절할 경우 탐지는 더욱 어려워진다.

취약성공격 단계에서 웜을 탐지하기 위해서는 네트워크 기반 침입탐지시스템을 설치하여 웜의 시그니처와 비교를 수행함으로써 탐지할 수 있다. 하지만 웜 작성자가 시그니처 기반 탐지를 우회하도록 웜을 작성하거나 새로운 취약성을 이용하는 경우 일반적인 시그니처 기반 탐지는 불가능하다.

감염 단계에서 웜을 탐지하기 위해서는 호스트 기반 침입탐지시스템을 설치하여 웜이 생성하는 특정 파일을 모니터링하거나, Tripwire[7]와 같은 파일 시스템 무결성 검사 도구를 이용하여 웜의 파일 변조를 모니터링하거나, 웜이 설치한 백도어를 탐지하기 위해 netstat 와 같은 시스템 도구를 활용하여 포트를 모니터링할 수 있다. 하지만 메모리 상주형 웜이나 전파 이외의 다른 공격을 하지 않는 웜의 경우에 탐지하기 어렵다.

전파 단계에서 웜을 탐지하기 위해서는 네트워크 기반 침입탐지시스템을 설치하여 outbound 트래픽을 모니터링해야만 한다. 이 경우 출발지 주소와 목적지 포트 번호가 동일한 트래픽이 여러 목적지 주소를 대상으로 하여 발생하는 경우에 웜으로 간주할 수 있다. 이 방식은 inbound 트래픽을 모니터링하는 목표선정 단계에서의 방식보다는 탐지가 용이하지만, 웜이 스캔 속도를 조절할 경우 목표선정 단계에서의 방식과 마찬가지로 특정 시간 동안 특정 이벤트의 개수에 기반하여 탐지하기 때문에 탐지가 어렵다.

## 3. 취약성 데이터베이스

취약성 정보를 활용하기 위해 CVE (Common Vulnerabilities and Exposures)[8] 목록으로부터 원격 공격이 가능한 취약성 목록을 추출한다. 추출된 목록으로부터 취약한 서비스 포트 번호와 추가적으로 공격 코드의 시그니처를 구할 수 있다면 둘 다 데이터베이스화한다.

공격 코드의 시그니처는 웜이 지금까지 사용한 적이 없다고 하더라도 차후에 충분히 사용되어질 수 있으므로 필요하다. 이상징후에 기반한 탐지 방식과는 달리, 공격 코드의 시그니처와 일치한다면 오탐률은 0 이라고 말할 수 있다.

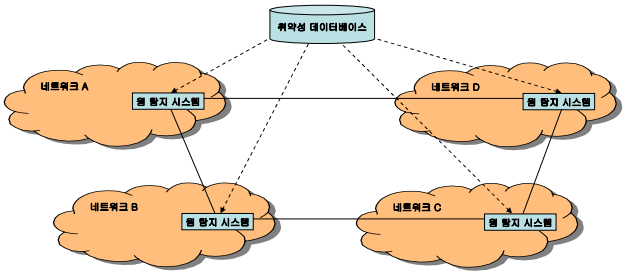
공격 코드의 시그니처뿐만 아니라 취약한 서비스 포트 번호도 필요하다. 공격 코드의 시그니처는 이미 알려진 공격에 한해서만 가능한 해결책이기 때문에 새로운 공격에 대비하기 위해서는 취약한 서비스 포트 번호도 데이터베이스화할 필요가 있다.

이와 같은 작업을 자동화하기 위하여 Nessus[9]와 같은 취약성 검사 도구를 활용할 수 있다. 본 논문에서는 설계에 목적을 두고 있기 때문에 Nessus 의 활용 방안과 같은 특정 도구나 구현을 자세히 언급하지는 않는다.

## 4. 시스템 구성

취약성 데이터베이스 기반 분산 웜 탐지 시스템은 중앙 취약성 데이터베이스와 네트워크마다 설치된 웜 탐지 시스템으로 구성된다. 인터넷 상에 웜 탐지 시스템을 탑재한 네트워크가 많을수록 웜 탐지 정확도가 향상되고 빠른 대응을 할 수 있다.

다음 [그림 2]는 취약성 데이터베이스 기반 분산 웜 탐지 시스템의 구성도를 나타낸다.




[그림 2] 취약성 데이터베이스 기반 분산 웜 탐지 시스템의 구성도

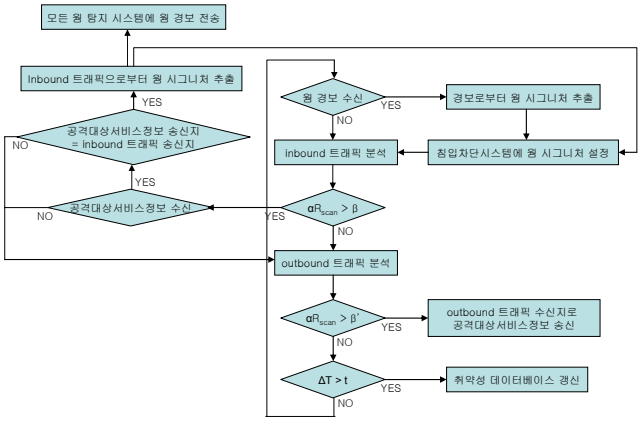
점선으로 취약성 데이터베이스로부터 각 웜 시스템으로 향하는 실선은 취약성 정보이다. 각 웜 탐지 시스템은 최신 취약성 정보를 취약성 데이터베이스로부터 주기적으로 받아 유지하고 있다. 실선으로 된 웜 탐지 시스템 간의 연결은 웜 탐지 시스템 간의 웜 탐지 정보 교환을 의미한다.

웜 탐지 시스템은 웜을 탐지하게 되면 공격 대상 서비스 정보를 웜의 진행 방향에 있는 웜 탐지 시스템으로 전파시킨다. 웜의 진행 방향에 있던 웜 탐지 시스템은 공격 대상 서비스 정보가 자신의 네트워크에서 감지되고 있는 이상징후와 일치한다면 웜 경보를 이상징후를 일으키는 웜 시그니처와 함께 모든 웜 탐지 시스템으로 전송한다. 이를 수신한 웜 탐지 시스템은 침입차단시스템과 연계하여 해당 시그니처를 포함한 패킷을 모두 폐기한다.

## 5. 탐지 알고리즘

탐지 알고리즘을 설계함에 있어 웜 시그니처에 기반한 탐지는 기본적인 부분으로 여기에서는 생략하였다. 본 논문에서 제시하는 탐지 알고리즘은 네트워크 기반 탐지 방법을 제시하고 있지만 호스트 기반 탐지 결과와 상관분석을 통해 더 높은 정확성을 확보할 수 있다.

각 웜 탐지 시스템은 다음 [그림 3]과 같은 탐지 알고리즘을 기반으로 동작한다.




[그림 3] 웜 탐지 시스템의 탐지 알고리즘

웜 탐지 시스템은 웜 경보 수신을 하게 되면 웜 경보로부터 웜 시그니처를 추출하여 침입차단시스템에서 이를 차단하도록 설정한다.

수신된 웜 경보가 없다면 inbound 트래픽을 분석하여 스캔률 ($R_{scan}$)에 취약성 가중치 ($\alpha$)를 곱하여 inbound 스캔 경고 임계치 ($\beta$)보다 크면 공격대상서비스정보 수신여부를 확인한다.

공격대상서비스정보는 웜으로 의심되는 공격에 대한 취약 서비스 정보를 웜에게 공격당하고 있는 곳으로 의심되는 네트워크로 보내는 정보이다. 이 정보를 수신한 네트워크에서 동일한 취약 서비스를 공격받고 있다고 판단되면 모든 웜 탐지 시스템으로 웜 시그니처와 함께 웜 경보를 보내주는 방식으로 동작한다.

공격대상서비스정보 수신여부 확인 결과, 수신하였다면 공격대상서비스정보 송신지와 inbound 트래픽 송신지가 같은지를 확인하여 같다면 inbound 트래픽으로부터 웜 시그니처를 추출하여 이를 포함한 웜 경보를 모든 웜 탐지 시스템에 전송하고 자신의 네트워크를 외부 네트워크로부터 보호하는 침입차단시스템이 웜 시그니처를 차단하도록 설정한다.

inbound 트래픽 분석이 완료되면 outbound 트래픽 분석을 시작한다. outbound 트래픽을 분석하여 스캔률 ($R_{scan}$)에 취약성 가중치 ($\alpha$)를 곱하여 outbound 스캔 경고 임계치 ($\beta$')보다 크면 outbound 트래픽 수신지로 공격대상서비스정보를 송신한다.

취약성 데이터베이스 갱신 주기를 확인하여 필요하다면 취약성 데이터베이스로부터 최신 취약성 정보로 자신의 데이터베이스를 갱신한다.

이후 과정은 처음부터 순환된다.

## 6. 한계점

이 시스템은 많은 네트워크에서 웜 탐지 시스템을 탑재해야만 제대로 된 성능을 발휘한다. 웜의 동작 특성상 단일 네트워크 내에서의 탐지보다는 분산된 탐지 시스템을 사용해야만 탐지의 효율성과 정확성을 향상시킬 수 있다.

이 시스템은 취약성 데이터베이스에 기반하고 있기 때문에 알려지지 않은 취약성을 사용해서 공격하는 경우 탐지에 어려움이 있다. 하지만 제로데이 공격과 같이 취약성 발표와 동시에 이루어지는 공격에 대해

서는 취약성 데이터베이스의 갱신만 최신으로 해준다면 충분히 대응할 수 있다.

## 7. 결론 및 향후 연구

취약성 데이터베이스를 기반으로 하여 분산된 웜 탐지 시스템을 구축함으로써 웜 탐지의 효율성과 정확성을 향상시키고자 하였다. 취약성 데이터베이스를 통해 스캔 위협에 가중치를 부여하였고 분산된 웜 탐지 시스템을 통해 웜 탐지를 정확하게 하여 신속하게 웜 경보를 웜 시그니처와 함께 웜 탐지 시스템을 탑재한 모든 네트워크에 전파하여 방어할 수 있도록 하는 시스템을 제안하였다.

향후에는 본 논문에서 제시한 시스템의 성능을 ns2, SSFNet, GTNetS 등과 같은 네트워크 시뮬레이션 도구를 사용하여 평가해보고자 한다.

## 참고문헌

[1] Darrell M. Kienzle, Matthew C. Elder, “Recent Worms: A Survey and Trends”, WORM’03, October 27, 2003.

[2] Stuart Staniford, Vern Paxson, Nicholas Weaver, “How to Own the Internet in Your Spare Time”, Proceedings of the 11th USENIX Security Symposium (San Francisco CA, August 2002).

[3] Stuart E. Schechter, Michael D. Smith, “Access For Sale - A New Class of Worm”, WORM’03, October 27, 2003.

[4] Symantec, “What is the difference between viruses, worms, and Trojans?”, http://service1.symantec.com/SUPPORT/nav.nsf/pfdocs/1999041209131106, March 30, 2005.

[5] CERIAS Intrusion Detection Research Group, "Digging For Worms, Fishing For Answers", 18th Annual Computer Security Applications Conference (Las Vegas, Nevada), December 9-13, 2002.

[6] Nicholas Weaver, Vern Paxson, Stuart Staniford, Robert Cunningham, “A Taxonomy of Computer Worms”, WORM’03, October 27, 2003.

[7] Gene H. Kim, Eugene H. Spafford, “The Design and Implementation of Tripwire: A File System Checker”, In ACM Conference on Computer and Communications Security, pages 18-29, 1994.

[8] CVE, “CVE (version 20040901)”, http://cve.mitre.org/cve/downloads/full-cve.html, Semptember 1, 2004.

[9] Nessus, “Nessus Open Source Vulnerability Scanner Project”, http://www.nessus.org.