한국산학기술학회논문지
Vol. 7, No. 6, pp. 1206-1213, 2006

DHT 기반 네트워크의 웜 시그니쳐 자동 생성기의 구현

김지헌[1*], 이유리[1], 박동규[1], 오진태[2], 장종수[2], 민병준[3]

Implementation of Automatic Worm Signature Generator in DHT Network

Ji-Hun Kim[1*], You-Ri Lee[2], Dong-Gue Park[3], Jin-Te Oh[4], Jong-Soo Jang[5] and Byeong-Jun Min[6]

요 약 자기 전파하는 웜들의 속도가 사람이 대응하는 속도 보다 매우 빠르기 때문에 zero-day 웜들을 봉쇄하기 위해서는 웜 시그니쳐의 빠른 검출과 자동생성이 필수적이다. 본 논문에서는 웜 공격에 대응 할 수 있는 시그니쳐 자동생성 방법을 제안하고, 제안한 방법의 프로토타입을 구현하여 DHT 기반 네트워크에 적용하여 웜 시그니쳐를 생성함으로써 제안한 방법의 유효성을 보이도록 한다.

Abstract Fast detection and automatic generation of worm signatures are essential to contain zero-day worms because the speed of self-propagating worms is too fast for humans to respond. In this paper, we propose an automatic signature generation method against worm's attack, and show the effectiveness of the proposed method by implementing it and appling it to the DHT based network and generating the worm signatures for it.

Key words : Internet Worm, Signature, DHT, Zero-day Worm

1. 서 론

정보기술의 발달로 누구나 네트워크를 통하여 텍스트, 이미지, 사운드, 동영상 등의 멀티미디어 데이터를 접할 수 있게 되었으며, 이를 통한 원격 회의, 가상 학습, 원격 진료 등의 네트워크를 이용한 산업 서비스가 가능하게 되었다. 그러나 이러한 통신/네트워크 기술의 발달과 함께 네트워크를 통한 인터넷 웜의 공격 형태 역시 Email, 파일 공유, 메신저 등을 이용한 다양한 기법이 사용되고 있으며, 전파속도가 단축되면서 더욱 치명적인 형태로 진화하고 있다.

네트워크 침해 유형이 기존에는 특정 호스트나 서버를 목표로 하여 개별 시스템이나 해당 시스템이 제공하는 서비스에 대한 공격이 주를 이루었으나 인터넷 웜은 광역 네트워크 자체에 대한 공격이 이루어지거나, 광역 네트워크의 서비스 제공을 방해하는 형태로 변화하고 있다. 현재 인터넷 웜의 공격 특징은 피해의 전파 속도가 빠르고, Zero-Day 공격이 증가되고 있으며, 공격 범위가 광역화 및 분산화 되고 있다는 것을 들 수 있다.

이를 방지하기 위하여 침입탐지 및 방지 기술이 연구되고 있으나, 현재의 침입탐지 및 방지 기술은 오탐율이 높고, 알려지지 않은 공격(Zero-day attack)에 대하여 신속하게 대응하지 못하는 문제점을 가지고 있는 상황이다. 그러므로 이러한 공격을 실시간으로 탐지하고 대응하여 네트워크 인프라를 보호하기 위한 기술 개발이 절실히 요구되고 있는 상황이다.

따라서 본 논문에서는 최근 인터넷 웜 공격에 대응할 수 있는 실시간 웜 시그니쳐 자동 생성 방법에 대하여 연구하고 이를 네트워크 분산 시스템인 DHT(Distributed Hash Table) 기반의 네트워크에 적용하여 시그니쳐 생성 시스템 간에 협력(cooperation)을 통해 인터넷 웜 공격에 대한 오탐율을 낮추고 탐지 속도의 향상을 위한 시그니쳐 생성기의 구현을 제안한다.

[1]순천향대학교 정보통신학과
[2]한국전자통신연구소 정보보호연구단
[3]인천대학교 컴퓨터공학과
*교신저자: 김지헌(zippykim@sch.ac.kr)

2. 관련 연구

웜 시그니쳐 생성 시스템은 Honeycomb 시스템을 시작으로 자동 웜 시그니쳐 생성 기술이 발전하게 되었다. 기존에 연구된 자동 웜 시그니쳐 생성 알고리즘 중에 대표적인 알고리즘들로는 Honeycomb[1], Aurograph[2], Earlybird[3], Pol,ygraph[4], Worm shield[5]등이 있다.

2.1 Honeycomb

Honeycomb[1]은 허니팟을 통해 들어오고 나가는 모든 트래픽이 시그니쳐 생성의 입력으로 사용되며 패턴 탐지 기술에 기반을 둔다. 입력으로 받은 패킷에 대하여 프로토콜 분석을 수행한 후, 네트워크 IP, TCP 시퀀스 초기 넘버 등이 일치하는 헤더들이 있는지를 비교하고, 만약 같은 목적지 포트들을 가지고 있는 연결들이 있다면, 메시지에 LCS(Longest-common substring)[7] 알고리즘을 적용하는 수평 탐지와 수직 탐지 방법을 사용하여 시그니쳐 풀에 더해지게 된다.

2.2 Autograph

Autograph[2]는 포트 스캐너 탐지 메커니즘을 사용하여 유입되는 트래픽에서 의심되는 트래픽을 구별하여 의심되는 트래픽에서 가장 자주 나타나는 바이트 시퀀스를 시그니쳐 후보로 사용하여 페이로드 분배(COPP : Content Based Payload Partitioning)를 사용하여 컨텐츠 블럭을 핑거프린팅하여 가장 많이 나타난 컨텐츠 블럭을 시그니쳐로 생성한다.

2.3 Earlybird

Earlybird[3]는 이전에 알려지지 않은 웜을 자동으로 빠르게 탐지하기 위해 트래픽부터 웜 탐지까지의 시그니쳐 추출이 자동화되어있는 시그니쳐 생성 방법으로 Content invariance, Content prevalence, Address dispersion의 세 가지 가정에 기반하고 있다. Earlybird는 네트워크에 입력되는 모든 패킷을 일정한 길이의 부분 문자열로 나누고 나누어진 컨텐츠의 라빈 핑거프린트(Rabin fingerprints)[6]를 구한 다음, 각 계산된 핑거프린트(fingerprints)를 목적지 포트와 프로토콜과 함께 해쉬값을 구하고, 이 해쉬 값들을 content prevalence table의 저장 카운팅하여 한계치를 넘으면 컨텐츠를 이동하여 address dispersion table 엔트리를 생성하여 저장하게 된다. 이 테이블에서도 빈도수가 체크되며 체크된 빈도수가 한계치를 넘으면 웜으로 간주하게 된다.

2.4 Polygraph

Polygraph[4]는 polymorphic 웜을 위한 시그니쳐 생성 방법으로 흐름 분류기를 사용하여 입력되는 모든 네트워크 트래픽을 의심스러운 트래픽과 무해한 트래픽으로 분류하고, 의심스러운 트래픽에서 토큰을 추출하여 웜을 탐지하기 위한 세 가지 시그니쳐 타입을 생성한다. 세 가지 시그니쳐 타입은 Conjunction 시그니쳐, Token-subsequence 시그니쳐, Bayes 시그니쳐이다. 다중 웜의 경우에도 시그니쳐 생성방법을 다루고 있으며, 세가지 시그니쳐 생성 방법들 중 Bayes 시그니쳐 생성 방법은 다중 웜의 경우에도 적용이 가능하며, Conjunction 시그니쳐 생성 방법과 Token-subseqence 시그니쳐 생성 방법에 대하여는 계층 클러스터링 방법을 제안하고 있다.

2.5 WormShield

WormShield[5]는 DHT 기반 네트워크에서 웜 시그니쳐를 빠르게 자동으로 생성하는 방법으로 로컬과 글로벌 주소분산 테이블을 이용하여 웜 관련 정보를 빠르게 전파하는 방식이다. DHT 기반의 네트워크의 각 모니터들은 입력으로 들어오는 트래픽을 라빈 핑거프린팅하여 컨텐츠 블럭을 생성한다. 각 모니터들은 블럭의 발생수를 확인하여 정해놓은 한계치를 넘어서면 주소 분산을 확인하고 한계치를 넘어서면 root 모니터에 있는 global table에 전송하고 global table에 발생한 수를 세어 global table의 한계치를 넘으면 웜으로 간주하여 시그니쳐를 생성하며 한계치를 넘지 못하면 global table 엔트리에 업데이트 된다. 생성된 시그니쳐는 WormShield 네트워크의 모든 모니터에 퍼지게 된다.

3. DHT 네트워크 기반의 시그니쳐 생성 알고리즘

3.1 DHT 방식의 네트워크

1999년 5월 파일 공유를 위한 Napster 소개로 시작된 Peer-to-Peer(P2P) 네트워킹 기술은 분산 컴퓨팅, 인터넷 전화에 성공적으로 적용됨에 따라, 현재 가장 관심 있는 인터넷상의 새로운 통신 방식으로 떠오르고 있다. 집중화 구조 없이 분산 시스템의 자율

구성을 목표로 하는 P2P 시스템에서 원하는 데이터를 쉽게 검색하여 찾고 관리하기 위한 문제를 해결하기 위하여 DHT(Distributed Hash Table) 기반의 분산 시스템 방식이 출현하였다.

DHT는 크게 keyspace partitioning, overlay network 두 부분으로 나뉘는데 해쉬의 키 집합들을 DHT시스템 내부의 각 노드에 분산(keyspace partitioning)시키고 DHT의 엔트리 포인트에 상관없이 키의 위치를 찾아갈 수 있도록 라우팅하는 알고리즘을 이용해서 DHT시스템 내부의 네트워크(overlay network)를 이동하며 목적지를 찾아가게 된다. 이러한 목적지 접근 방식을 structured key based routing이라 하며, 해쉬 테이블 내부의 버킷(bucket)을 그룹으로 묶고 그룹 사이를 링크로 연결하여 자기가 원하는 정보가 있는 그룹을 찾아갈 수 있도록 해주는 것이다. 외부에서 보면 해쉬처럼 보이지만 내부에서 원하는 버킷을 찾아가는 방식은 기존의 해쉬와 다르다.[9][10]

3.2 DHT 네트워크 기반의 시그니쳐 생성 알고리즘

인터넷 웜의 공격이 발생하면 네트워크에 많은 트래픽이 발생하게 된다. 이것은 인터넷 웜의 공격 특징 중에 하나인 자신을 복제 하여 전파하는 행위가 이루어지게 되면서 네트워크에 형성되어 있는 다른 컴퓨터를 찾기 위해 스캐닝을 실시하거나 자신이 복제한 파일을 다른 컴퓨터에 전송을 시작하기 때문이다. 하지만 트래픽의 증가는 인터넷 웜이 활동할 때에만 증가하는 것은 아니라. 대량의 파일을 전송할 때에도 이런 현상이 발생할 수 있으며 하나의 서버에 많은 접속자가 발생 했을 때에도 동일한 현상이 발생 할 수 있다. 따라서 단순한 트래픽의 증가로 인터넷 웜의 공격이 시작되었다고 볼 수 없게 된다.

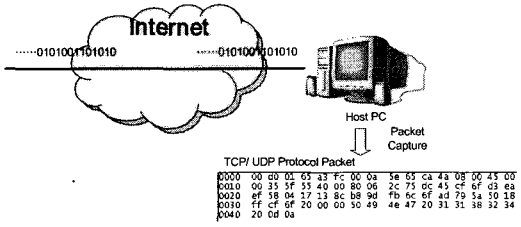


그림 1. 패킷 수집 과정

그러므로 인터넷 웜을 판단하기 위해서는 단순한 트래픽의 증가여부가 아니라 패킷의 형태와 source IP, destination IP를 확인하고 접근하고자 하는 포트를 분석해야 한다. 이러한 분석을 위해 그림 1과 같이 DHT 기반의 네트워크에서 발생하는 패킷을 수집하고 분석하는 것은 시그니쳐 생성에 있어 인터넷 웜의 판단에 중요한 기준으로 적용 된다.

따라서 DHT 네트워크 기반에서 알려지지 않은 웜을 자동으로 빠르게 탐지하기 위하여 그림 2와 같이 입력되는 모든 패킷을 헤더와 payload로 분류하고 헤더의 source IP, destination IP, source Port, destination Port, protocol 정보를 추출 패킷 분석에 활용한다.

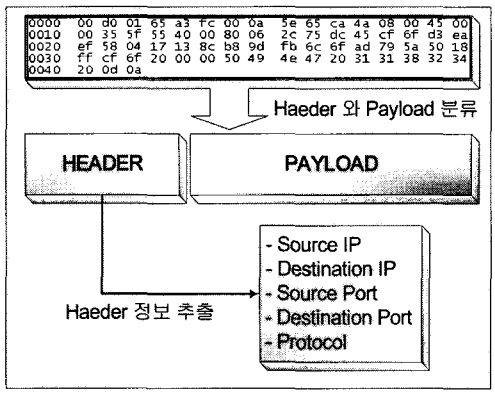


그림 2. 패킷의 정보추출

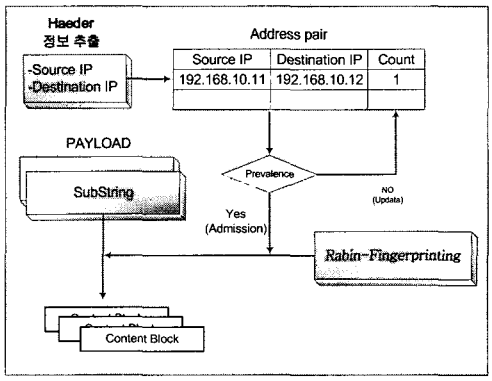


그림 3. Address Pair를 이용한 패킷 필터링

분리된 payload의 부분 문자열은 라빈 핑거프린팅 [6] 알고리즘을 사용하여 컨텐츠 블럭을 생성한다. 라빈 핑거프린팅은 다항식을 이용한 복잡한 계산으로 이루어지며 많은 메모리를 사용하기 때문에 CPU에 많은 작업을 요구한다. 이러한 작업이 자주 일어나면 CPU를 사용하기 위한 데이터의 대기 시간이 길어지고 실시간으로 처리되어야 하는 패킷 분석이 원할 하게 이루어 질수 없으며, 이를 방지하기 위하여 payload 부

분문자열을 핑거프린팅하기 전에 패킷의 헤더정보를 이용해 패킷을 필터링 한다. 필터링은 그림 3과 같은 과정으로 이루어지며 목적지 주소와 출발지 주소가 쌍을 이루고 있는 패킷의 응답 횟수를 카운트하여 조건이 만족하면 핑거프린팅하는 방법으로 CPU의 연산 시간을 단축하고 인터넷 웜에 대한 시그니쳐 생성을 효율적으로 가능하게 한다. DHT 네트워크 기반의 시그니쳐 생성 알고리즘은 content prevalence table, local address dispersion table, global content prevalence & address-dispersion table을 사용하며 그림 4와 같다. 라빈 핑거프린팅된 컨텐츠 블럭은 local address dispersion table에 엔트리가 있는지 확인하여 있으면 업데이트하고 없으면 content prevalence table에 보관되어 동일한 컨텐츠 블록이 발견될 때 마다 카운터를 증가 시켜 카운터가 한계치를 넘어서면 local address dispersion table에 컨텐츠가 이동되어 source IP와 destination IP 등 엔트리를 생성하게 된다. 이 테이블에서는 source IP와 destination IP를 카운트하여 한계치를 넘으면 DHT 네트워크에 의해서 빠르게 root 모니터에게 컨텐츠 블럭의 카운터와 source IP, destination IP 카운터가 컨텐츠 블럭과 함께 보내져 global content prevalence & address-dispersion table에 저장되게 되며 저장된 global prevalence 카운터와 global address dispersion의 카운터 둘 다 한계치를 넘으면 웜으로 간주하여 모든 게이트웨이에 리포트되며 빠르게 대응책이 전파되게 된다.

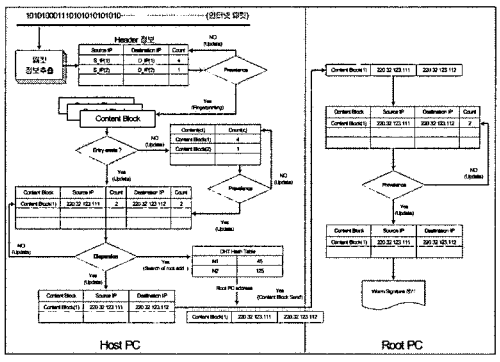


그림 4. DHT 네트워크 기반 시그니쳐 생성 알고리즘

4. DHT 네트워크 기반의 시그니쳐 생성 알고리즘의 구현

이전 장에서는 DHT 네트워크 기반의 인터넷 웜을 탐지 하고 시그니쳐 생성을 위한 알고리즘에 대하여 설명하였다. 본 장에서는 이전 장에서 제시한 알고리즘을 구현한다.

구현된 DHT 네트워크 기반의 시그니쳐 생성기는 프로토타입으로 prevalence와 dispersion의 한계치를 관리자가 설정 할 수 있도록 하였으며, 이 값은 오탐율과 매우 밀접한 관계가 있다.

4.1 시스템의 구성

인터넷 웜 시그니쳐를 생성하기 위한 시스템의 구성은 host PC에서 인터넷 웜을 탐지하기 위한 프로그램과 DHT 네트워크 기반의 global Table을 관리하기 위한 프로그램으로 구성된다.

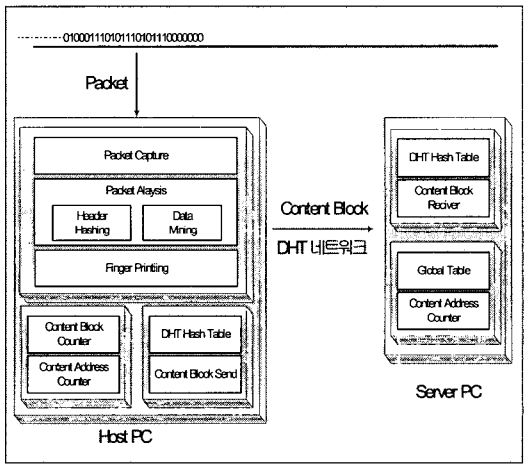


그림 5. 시그니쳐 생성 시스템 구성

Host PC에서 사용되는 프로그램의 구성은 네트워크의 패킷을 캡처하고, 컨텐츠 블럭을 생성하기 위한 트래픽 분석기, 핑거프린팅을 통해 생성된 컨텐츠 블럭의 출현 빈도와 주소 분산을 분석하기 위한 content prevalence table, local address dispersion table로 구성되어 있다. 이들은 정해진 한계치를 넘어서면 DHT 네트워크상의 root 모니터에게 해당 정보를 전송하게 된다. 그리고 server PC상에서 root 모니터의 상태를 확인할 수 있는 즉 global content prevalence & address dispersion table을 관리하기 위한 프로그램으로 구성된다.

4.2 시스템 구현 환경

시그니쳐 생성 시스템을 구현하기 위해서는 네트워크상의 패킷을 캡처할 수 있어야 하며, 이를 위해서

Winpcap에서 제공하는 C++ 언어로 구현된 라이브러리를 이용하였다. Winpcap은 패킷 캡처를 위해 리눅스 OS에서 사용하던 TCPdump를 윈도우 운영체제(Window OS)용으로 개발 한 것으로 기존에 리눅스 운영체제를 중심으로 이루어지고 있던 네트워크 패킷에 대한 연구를 윈도우 상에서도 가능하도록 하였다. 본 논문에서 구현하고 있는 시그니쳐 생성 시스템은 기존의 연구와 달리 윈도우 환경에서 시그니쳐 생성이 가능하도록 하였다. 패킷 분석을 위해 사용되는 packet capture, packet analysis, fingerprintring, content block counter, content address counter, DHT hash table은 Winpcap에서 제공하는 라이브러리를 사용하기 위해 동일한 C++ 프로그래밍 언어를 이용하여 구현 하였다. 사용자의 편의성과 데이터의 확인을 위한 사용자 환경은 비쥬얼 베이직을 이용하여 보다 편리한 시스템을 구현 하고자 하였다. 표1은 시그니쳐 생성 시스템 개발 환경을 나타내고, 그림 6은 host PC의 웜 탐지 시스템 GUI이고, 그림 7은 global content prevalence & address-dispersion table 관리를 위한 root 모니터 GUI이다.

표 1. 시그니쳐 생성 시스템 개발 환경

OS	CPU	RAM	Programing Lange
Window XP	Pentium 4 3.1G	1G	C++ ,비쥬얼베이직

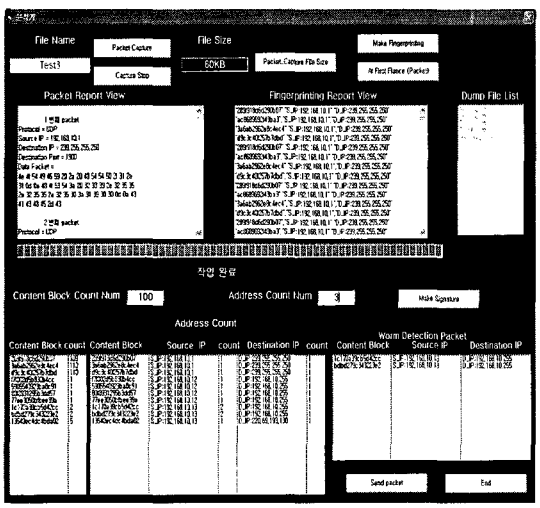


그림 6. Host PC의 웜 탐지 시스템 GUI

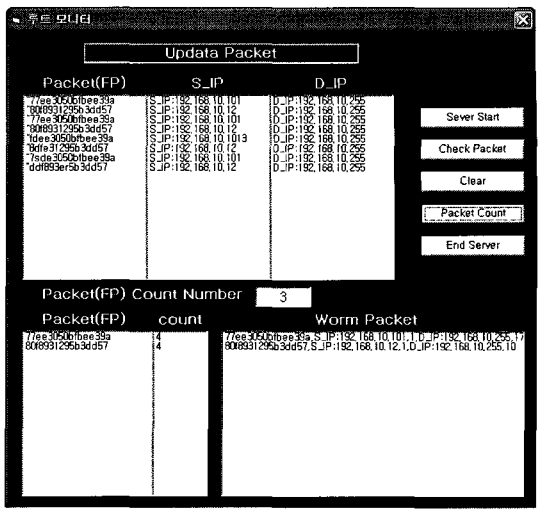


그림 7. Global content prevalence & address -dispersion table 관리를 위한 Root 모니터 GUI

5. 성능평가

본 논문에서 제안하고 있는 인터넷 웜 시그니쳐 생성 시스템의 성능을 평가하기 위해 소규모의 DHT기반 네트워크를 구성했다. 인터넷에서 활동 중인 실제 웜을 이용하여 실제 상황에서의 시그니쳐 생성에 근접한 시험을 수행하기 위하여 그림 8과 같은 시험 환경을 구축하였다. 시험 환경은 표 1과 같으며, 시험을 위한 네트워크에 웜의 활동에 대한 패킷 탐지 시간은 1시간으로 제한하였다. 첫 번째 시험은 외부 망과 분리된 네트워크에 트래픽을 발생시키기 위해 트래픽 발생기를 자체 제작하여 시험 네트워크상에 트래픽을 발생 시켜 시그니쳐를 생성하게 하였다. 두 번째 시험은 실제 인터넷에서 활동 중인 블레스터 웜을 시험 네트워크상에서 실행 시켜 인터넷 웜의 활동으로 인한 패킷을 발생 시키고 이를 분석 하고자 하였다. 블레스터 웜을 시험에 사용한 것은 인터넷 웜의 대표적인 특징인 IP 주소를 스캐닝하고 자신을 복제하여 다른 컴퓨터에서 실행 하도록 하는 공격 형태를 가지고 있기 때문에 그 전파가 빠르고, 테스트에 필요한 충분한 트래픽 발생하기 때문이다.

또한 핑거프린팅으로 인한 CPU의 과도한 계산을 방지하기 위해 제안된 헤더정보를 이용한 패킷 필터링의 효율성을 확인하기 위해 위에 실시한 두 가지 시험에 각각 패킷 필터링 을 사용하지 않는 경우와 패킷 필터링을 사용한 경우를 분리하여 시험하였다. 본 시험에서 사용되는 단계별 한계치는 표2와 같으며, 시험

네트워크 시스템 구축환경은 표 3과 같다.

표 2. 시험에 사용된 단계별 한계치

Packet Count	Header Count	Content Block Count (Max /Min)	Content Block Address Count (Max /Min)	Content Block Count (Global Table)
제한 없음	2	2	1	2

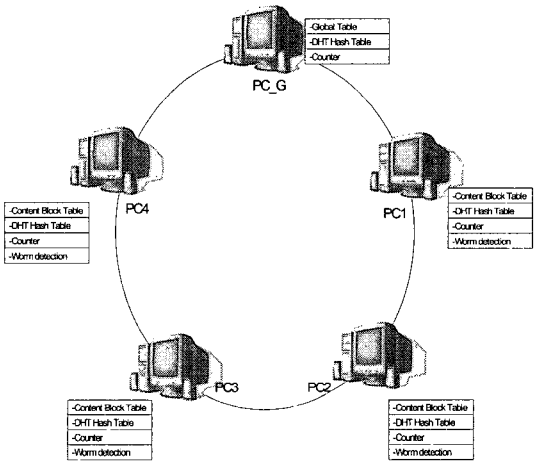


그림 8. DHT 기반의 시험 네트워크

표 3. 시험 네트워크 시스템 구축 환경

PC Name	OS	CPU 속도	메모리
PC1	Window XP	Pentium 4	1G
PC2	Window XP	Pentium 4	512M
PC3	Window XP	Pentium 4	1G
PC4	Window XP	Pentium 4	1G
PC_G (Global Table)	Window2003 Server	Pentium 4	2G

첫 번째 시험한 결과 일반적인 네트워크 트래픽에 대해서 시그니쳐 생성 시스템의 host PC4에서 하나의 웜 시그니쳐를 생성하고 이를 global Table을 관리하는 PC_G에 전송 하였다. 하지만 global Table에서는 최종 인터넷 웜으로 시그니쳐를 생성하기 위한 조건이 만족하지 않기 때문에 인터넷 웜에 대한 최종 시그니쳐는 생성 되지 않게 된다. 동일한 패킷을 이용하여 라빈 핑거프린팅에 대한 CPU의 연산 시간을 줄이기 위한 방법으로 핑거프린팅 단계 전에 캡처된 패킷에 대한 헤더 정보를 이용한 필터링을 실시하여 일차 필터링을 실시하고 결과에 대한 패킷을 대상으로 라빈 핑거프린팅을 실시했을 때의 결과 연산에 사용된 패킷의 수와 라빈 핑거프린팅 시간 소요가 줄어든 것을 확인 하였다.

표 4. 시험 1에서 실험 결과

PC Number	Packet Count	Content Block Count (Max /Min)	Content Block Address Count (Max /Min)	Worm detection	Finger Printing Time (sec)
PC1	2314	274 / 1	1 / 0	0	5.94
PC2	2317	276 / 1	1 / 0	0	5.95
PC3	2702	274 / 1	1 / 0	0	6.94
PC4	2314	274 / 1	4 / 0	1	5.94
PC_G (Global Table)	-	1	-	0	

표 5. 시험 1에서 데이터 헤더 정보를 이용한 필터링 실험 결과

PC Number	Packet Count	Content Block Count (Max /Min)	Content Block Address Count (Max /Min)	Worm detection	Finger Printing Time (sec)
PC1	1631	274 / 1	1 / 0	0	4.19
PC2	1502	276 / 1	1 / 0	0	3.86
PC3	2122	274 / 1	1 / 0	0	5.45
PC4	1312	274 / 1	4 / 0	1	3.37
PC_G (Global Table)	-	1	-	0	

첫 번째 시험 결과에서 확인 할 수 있듯이 헤더 정보를 이용한 필터링을 실시하지 않은 경우와 실시했을 때 인터넷 웜에 대한 시그니쳐 생성 결과는 동일하며, 핑거프린팅에 소요된 시간이 줄어들었음을 알 수 있다. 각 표에서 사용한 content block count 와 content block address count의 수는 각각에 대한 최대 수와 최소 수 만을 표시 했으며, 핑거 프린팅 시간 측정은 다른 동작으로 안한 시간 지연을 피하고자 테스트에 사용된 동일한 패킷을 이용해 패킷에 대한 핑거프린팅 시간을 별도로 측정하였다.

표 6. 시험 2에서 실험 결과

PC Number	Packet Count	Content Block Count (Max /Min)	Content Block Address Count (Max /Min)	Worm detection	Finger Printing Time (sec)
PC1	3712	210 / 1	6 / 0	1	9.54
PC2	2111	152 / 1	1/ 0	0	5.42
PC3	3357	232 / 1	5 / 0	1	8.62
PC4	1950	190 / 1	1 / 0	0	5.01
PC_G (Global Table)	-	2	-	1	

표 7. 시험 2에서 헤더 정보를 이용한 필터링 실험 결과

PC Number	Packet Count	Content Block Count (Max /Min)	Content Block Address Count (Max /Min)	Worm detection	Finger Printing Time (sec)
PC1	3112	210 / 1	6 / 0	1	8.00
PC2	1403	152 / 1	1/ 0	0	3.60
PC3	2987	232 / 1	5 / 0	1	7.67
PC4	1720	190 / 1	1 / 0	0	4.42
PC_G (Global Table)	-	2	-	1	

두 번째 시험에서는 실제 네트워크에서 활동하는 블레스터 웜에 대한 시그니쳐 생성을 실험하였다. 이를 위해 PC1, PC2, PC3, PC4에서 생성된 컨텐츠 블럭을 확인하고 global table에 전송된 컨텐츠 블럭을 확인 하였다. 블레스터 웜의 최초 발생지인 PC3과 블레스터 웜이 전파된 PC1의 경우 다른 PC보다 많은 패킷이 캡처 되었으며, 동일한 컨텐츠 블럭이 다른 PC보다 많은 것을 확인 하였다. PC1과 PC3에서 1개씩의 웜을 검출 하였고 이를 global table이 있는 PC_G에 전송 하였다. PC_G는 동일한 컨텐츠 블럭이 들어오는 것을 확인하고 카운트가 시그니쳐 생성 조건에 일치 하게 됨으로써 최종적으로 웜 시그니쳐를 생성하게 되었다. 또한 두 번째 시험에서도 핑거프린팅 전 단계에서 헤더정보를 이용한 헤더 정보를 이용한 필터링을 실시하여 시그니쳐 생성에 차이점과 시간을 비교 하였으며 그 결과 헤더 정보를 이용한 패킷의 필터링이 라빈 핑거프린팅의 시간을 단축 하는 것을 확인 하였다. 테스트 결과 블레스터 웜에 대한 시그니쳐 생성이 되었으며 이는 헤더 정보를 이용한 필터링을 통하여 필터링 한 경우에도 동일하게 생성되었다. 또한 헤더 정보를 이용한 필터링 이후 핑거프린팅 시간이 줄어드는 것을 확인 하였다.

6. 결 론

본 논문에서는 최근 인터넷 웜 공격에 대응 할 수 있는 실시간 시그니쳐 자동 생성 방법에 대하여 연구하고 이를 DHT 기반 네트워크에 적용하여 Zero-day 공격에 신속하게 대응할 수 있는 웜 시그니쳐 생성기의 프로토타입을 구현하였다.

구현된 시그니쳐 생성 시스템을 시험해본 결과 일반적인 패킷의 입력과 인터넷 웜 패킷을 정확히 구분하여 시그니쳐를 생성하는 것을 확인 하였으며, 헤더 정보를 이용한 필터링은 핑거프린팅에 대한 시간은 단축시키고 결과에는 영향을 미치지 않는 것을 확인했다.

본 논문에서 구현한 웜 시그니쳐 자동 생성기는 프로토타입으로 향후 보다 정확한 웜 시그니쳐 생성을 위하여 polymorphic웜을 검출할 수 있는 방법의 추가 보완이 필요하며, 또한 다중 웜에 대해서도 대처할 수 있는 웜 검출 방법에 대한 연구가 더 계속되어야 할 것으로 사료된다. 그리고 실제 네트워크상에서 사용되기 위하여 시험 네트워크가 아닌 실제 사용되는 네트워크에서 보다 많은 인터넷 웜에 대한 테스트가 필요할 것으로 사료된다.

참고문헌

[1] C. Kreibich and j. Crowcroft. "Honeycomb - creating intrusion detection Signatures Using Honeypots." In Proceedings of the Second Workshop on Hot Topics in Networks, Boston, November 2003

[2] H-A. Kim and B.Karp. Autograph "Toward automated, distributed worm signature detection." In proceedings of the USENIX Security Symposium, 2004

[3] S. Singh, C. Estan, G. Varghese, and S. Savage., "Automated worm fingerprinting." In OSDI 2004

[4] J. Newsom, B. Karp, and D. Song. Polygraph, "Automatically generating signatures for polymorphic worms.", In SP' 05 : Proceedings of the 2005 IEEE Symposium on Security and Privacy, Washington, DC, USA 2005. IEEE Computer Society

[5] V. Yegneswaran, J.T.Giffin, and a.S.J.Paul Barford. "An architecture for generating semanticaware signatures." In Proceedings of the 14th USENIX security Symposium, August 2005

[6] Calvin Chan, Hahua Lu "CMPUT690 Term Project Fingerprinting using Polynomial"

[7] E.M.McCreight, "A space-economical suffix-tree construction algorithm." Journal of the ACM, vol. 23, pp. 262-272,1976

[8] 손영성 "피어의 컴퓨팅 능력을 고려한 분산 해쉬 테이블 기반 피어 투 피어 컴퓨팅 시스템" MagicSquare, 2006

[9] Chord, http://www.pdos.lcs.mit.edu/chord/, Project Homepage.

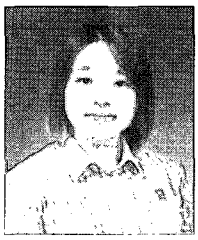
[10] 손영성 "DHT 기반 P2P 네트워크에서 네트워크 토플로지를 고려한 메시지 라우팅 매커니즘", 한국해양정보통신학회 2005

김 지 헌(Ji-Hun Kim) [준회원]

- 2005년 2월 : 순천향대학교 정보기술공학과 (공학사)
- 2005년 3월 ~ 현재 : 순천향대학교 정보통신공학과 석사 재학

<관심분야>
컴퓨터 프로그래밍, 네트워크 보안

이 유 리(You-Ri Lee) [정회원]



- 2004년 2월 : 순천향대학교 정보기술공학과 (공학석사)
- 2004년 3월 ~ 현재 : 순천향대학교 정보통신공학과 박사 재학

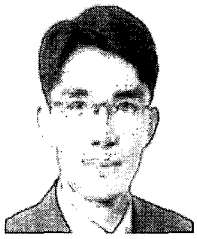
<관심분야>
컴퓨터 프로그래밍, 네트워크 보안

박 동 규(Dong-Gue Park) [정회원]

- 1992년 : 한양대학교 대학원 전자공학과 공학박사
- 1992년 ~ 현재 : 순천향대학교 정보통신공학과 교수

<관심분야>
접근제어, 보안

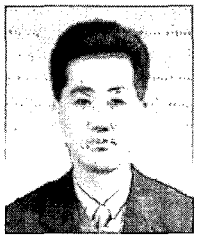
오 진 태(Jin-Te Oh) [정회원]



- 1990년 : 경북대학교 전자공학과 학사 졸업
- 1992년 : 경북대학교 전자공학과 석사 졸업
- 1992년 ~ 1998년 : 한국전자통신연구원 선임연구원
- 2003년 ~ 현재 : 현재 한국전자통신연구원 네트워크보안 연구팀 팀장/선임연구원

<관심분야>
네트워크 보안, 비정상행위 탐지기술

장 종 수(Jong-Soo Jang) [정회원]



- 1984년 : 경북대학교 전자공학과 학사 졸업
- 1986년 : 경북대학교 전자공학과 석사 졸업
- 2000년 : 충북대학교 박사졸업
- 1989년 ~ 현재 : 한국전자통신연구원 네트워크보안 그룹 그룹장/책임연구원

<관심분야>
네트워크 보안, 정책기반 보안관리기술

민 병 준(Byeoung-Jun Min) [정회원]

- 1979년 ~ 1983년 : 연세대학교 전자공학과 학사
- 1983년 ~ 1985년 : 연세대학교 대학원 전자공학과 석사
- 1986년 ~ 1991년 : 미국 캘리포니아대학교(UC Irvine) 컴퓨터공학과 박사
- 1984년 ~ 1986년 : 삼성전자 연구원
- 1992년 ~ 1994년 : KT 선임연구원
- 1995년 ~ 현재 : 인천대학교 컴퓨터공학과 교수

<관심분야>
컴퓨터 및 네트워크 보안, 유비쿼터스 컴퓨팅