주 제

인터넷 웜의 탐지 및 대응기술

대구가톨릭대학교 **전용희**

차 례

I. 서 론

인터넷 웜의 효시인 모리스(Morris)가 1988년 발표된 이후로 현재까지 많은 웜 공격이 발생되고 있으며, 인터넷 웜은 네트워크 보안 연구의 주요한 문제가 되었다. 웜의 확산으로 국가 경제에 막대한 피해를 끼치고 있으나, 아직 웜에 대한 공격을 사전에 방지하기 위한 확실한 대책이 없는 실정이다. 본 논문에서는 차세대 네트워크 인프라의 하나로 구축되고 있는 BcN(Broadband convergence Networks) 환경에서 웜의 전파 특성과 탐지 및 대응 기술에 대하여 기술하고자 한다. 이를 통하여 BcN 환경에서의 웜의 전파 특성과 모델링에 대하여 보다 나은 이해를 할 수 있고, 웜의 조기 탐지 및 효과적인 대응 기술을 개발하는데 도움을 줄 수 있을 것이다. BcN 전달망의 주요한 특징은 서비스 품질(QoS : Quality of Service) 보장, 고도의 통신망 관리 기능과 보안(Security) 기능, IPv6 주소체계의 수용을 통한 다양한 서비스를 쉽게 창출할 수 있는 개방형 망구조(Open API)를 도입하는 것이다. BcN 환경에서는 통신망의 기능이 다양화되고 고도화됨에 따라 통신망에 보안침해 사고가 발생하면 그 피해가 전체 네트워크로 광범위하게 확산될 수 있고, 또한 다양한 경로를 통하여 통신망에 불법적인 접근이 가능하므로 통신망의 신뢰성과 보안성을 위하여 보안 기술의 고도화가 요구된다. BcN과 같은 초고속 통신망에서는 웜의 확산을 가속화 시킬 수 있다. 네트워크 대역폭이 증가함에 따라, 웜의 전파에 대응할 수 있는 시간도 단축된다. 따라서 웜의 전파 특성에 대한 이해와 웜을 조기에 탐지하고 격리할 수 있는 메커니즘에 대한 연구가 시급하다[2].

2001년 7월 코드 레드 웜의 발생으로, 인터넷 웜에

대하여 더 많은 관심을 갖게 되었다. 웜은 항상 연결된 광대역 접속을 포함하여 인터넷 연결성이 유비쿼터스 하여짐에 따라 더욱 유행하게 되었고, 네트워크 애플리케이션의 폭발적인 증가와 함께 네트워크 보안에 대한 인터넷 웜의 위협이 점차적으로 심각해지고 있다. 바이러스와는 달리, 웜은 수많은 복제를 가지고 취약한 호스트를 탐색하고 감염시키기 위하여 설계된 독립적인 자동 프로그램이다. 가장 단순한 웜은 감염시킬 호스트를 임의로 스캔 한다. 새로운 공격 목표가 발견되면 웜은 공격 코드를 전파하며, 피해 시스템 내에서 공격 코드를 실행한다.

정확한 웜 모델을 통하여 웜의 행위에 대한 통찰력을 얻을 수 있고, 웜 확산 체인에서의 약점을 식별하고 새로운 웜 위협에 대한 손해 평가를 위하여 정확한 예측을 할 수 있게 한다. 웜의 행위를 예측하는 이유는 다음과 같다[19-23]:

- 과거에 관측된 웜 행위에 대한 보다 나은 이해
- 웜의 위협 가능성에 대한 평가
- 인터넷상의 미래 웜의 영향에 대한 평가
- 웜 확산에 대한 탐지 메커니즘 설계의 기초
- 웜 특성화에 관련 있는 파라미터의 결정

II. 인터넷 웜

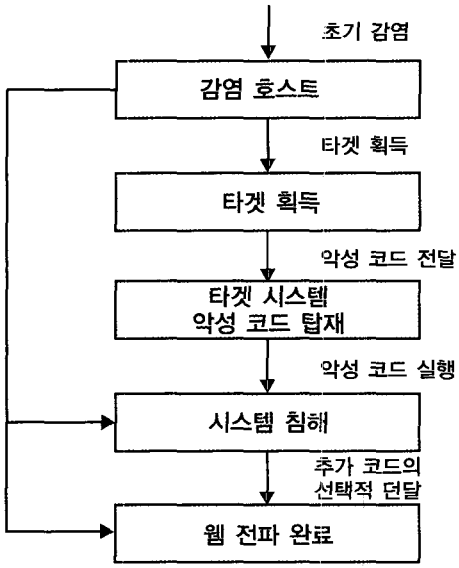
2.1 정의 및 전파 단계

웜은 지난 수년 동안 인터넷에 대하여 증가하는 위협이 되고 있다. 바이러스와는 달리, 웜은 네트워크를 인식하여(network-aware) 자기 스스로 전파되며(self-propagating), 인터넷 상의 거의 모든 곳에 다다를 수 있으며 매우 큰 영향을 끼친다. 웜은 감염되는 시스템만의 문제가 아니라, 네트워크 부하를 증가시킴으로 인하여 감염되지 않은 시스템들에게도 문제를 일으킨다. 웜에 대한 정확한 이해와 분석을 쉽게 하기 위하여 웜의 행위를 정확하게 말해주는 정의가 요구된다. 웜과 유사한 것으로 바이러스가 있다. Webopedia[16]는 바이러스를 다음과 같이 정의한다. "바이러스는 우리가 알지 못하는 사이에 컴퓨터에 적재되어 임의로 수행되는 프로그램이나 코드의 조각이다." 이것은 시스템 상에서 사용자가 원하지 않는 악성코드를 수행하는 웜의 기본 성질을 나타낸다. 그러나 전파의 의미에서 웜의 행위를 나타내지 못한다. Webopedia에서 웜은 다음과 같이 정의되고 있다[17]. "컴퓨터 네트워크 상으로 자신을 복제하는 프로그램이나 알고리즘으로, 컴퓨터 자원을 소모하거나 아마도 시스템을 정지시키는 것과 같은 악성 행동을 통상적으로 수행한다." 본 논문에서는 이런 웜의 정의에 추가적인 수식어를 첨가하는 것이 좋을 것 같다[6]. 웜은 자기 스스로 전파하며, 여기서 자기 스스로 전파한다는 의미는 사용자의 간섭 없이 컴퓨터 시스템 사이에 확산될 수 있는 것으로 정의될 수 있다. 위의 정의들을 종합하면, 웜에 대한 정의는 다음과 같다. "웜은 컴퓨터 네트워크를 통하여 확산되는 자기 스스로 전파되는 바이러스이다."

웜은 대표적으로 스캐닝 단계와 전파 단계를 통하여 감염된다. 스캐닝 단계에서 웜은 자신이 감염시킬 수 있는 네트워크에 연결된 호스트를 찾는다. 사용하는 프로토콜에 따라서 스캐닝 방법이 다르다. TCP 프로토콜을 사용하는 경우 TCP SYN 패킷을 보내 응답을 받아서 스캐닝 정보를 확인한다. UDP 프로토콜의 경우 UDP request 메시지를 보내 응답 메시지를 받아서 정보를 받는다. 그러나 진화한 웜의 경우에는 스캐닝과 감염을 동시에 수행하는 것도 있다. (그림 1)은 웜의 대표적인 전파 단계를 보여준다.

단계별로 발생하는 대표적인 활동은 아래와 같다[6]:

1) 초기 감염 단계: 웜에 의하여 이미 감염된 시스



(그림 1) 웜의 대표적인 전파 단계

템이 존재하고 그 웜이 시스템 상에서 활동적이라는 가정에서 시작한다.

2) 타겟 획득: 이 단계에서 IP 주소, 전자 메일, 파일 시스템 전달 등을 통하여 목적지 시스템에 도달을 시도한다. 웜은 또한 타겟 시스템에 수동적으로 전달될 수 있다. 예를 들어, 웜에 감염된 웹 콘텐츠가 웹 서버에 의하여 타겟으로 전달 될 수 있다.

3) 악성 코드 전달: 일단 시스템이 목적지로 정해지면, 감염을 준비하기 위하여 목적지 시스템으로 웜을 전달 할 필요가 있다. 코드 전달은 네트워크 파일 시스템, 전자 메일, 웹 클라이언트, 원격 명령 셸, 혹은 버퍼 오버플로 등과 관련된 패킷 페이로드의 일부로써 전달됨이 관측되었다.

4) 악성 코드 실행: 웜 전파를 위하여 악성 코드가 다음과 같은 방법으로 실행된다.

- 명령 라인으로부터 직접 호출
- 버퍼 오버 플로 혹은 다른 프로그램적 공격
- 전자 메일 클라이언트
- 웹 클라이언트
- 사용자 간섭
- 타겟 시스템에 의한 자동 실행

5) 추가 코드의 선택적 전달: 타겟 시스템이 침해된 후, 추가적인 코드가 FTP/TFTP, 네트워크 파일 시스템을 통하여 전달될 수 있다.

2.2 종류

[15]에서는 웜의 분류 기준으로 아래와 같은 5가지를 이용하고 있다.

- 웜이 감염시키고자 하는 호스트를 찾는 방법
- 웜이 전파되는 네트워크 매개체
- 웜이 활동하는 방식
- 웜의 내부에 있는 데이터
- 웜 프로그램의 의도

공격을 개시하기 전에, 웜은 타겟 호스트의 시스템 취약성을 조사할 필요가 있다. 그리고 웜의 전파를 가속화하기 위하여 스캐닝 전략을 이용한다. 웜 저자들은 기본적으로 다음의 프로세스를 구현한다[19]: 취약 호스트 식별, 타겟 호스트 침해, 웜 전달 및 활성화 웜을 특성화할 수 있는 몇 가지의 파라미터는 다음과 같다:

- 전달 프로토콜: TCP vs. UDP
- 전달되는 데이터 양
- 스캐닝 전략
- 지연 대 대역폭 제한

Zou 등은 스캐닝 전략에 따라 웜을 다음과 같이 5가지로 분류한다[22]:

1) 이상적(idealized) 웜

인터넷상의 모든 취약한 호스트의 완전한 IP 주소를 가지고 있으며, 다음과 같이 두 가지로 세분 된다:

- 완벽한(perfect) 웜: 가장 전파가 빠른 웜이 될 것이다. 인터넷상의 모든 취약한 호스트 주소를 알고 있으며, 모든 감염 호스트들은 서로 완전하게 협력한다. 감염 지연을 고려한 모델과 고려하지 않은 모델이 있다. 지연을 고려하지 않은 경우, 이 웜은 수 초 안에 모든 취약 호스트를 감염시킬 수 있다.
- Flash 웜: Staniford 등이 도입한 분류로, 인터넷상의 모든 취약 호스트의 IP 주소를 알고 있으며 n개의 스캐닝 공간을 가진다[12]. 이 웜의 전파는 동질 시스템에서의 감염 확산 가정을 만족하며, 3.2절에서 기술하는 단순 역학 모델에 의하여 모델 될 수 있다.

2) 균일(uniform) 스캔 웜

이 웜은 다시 다음과 같이 네 가지로 세분 된다:

- 코드 레드 웜: 전체 IPv4 공간을 스캔하는 웜을 말한다.
- 히트 리스트 웜: 히트 리스트 스캔은 초기 확산 속도를 증진하기 위하여 잠재적인 취약 호스트의 목록을 수집하는 방법이다. 이 웜은 짧은 시간 내에 많은 취약 호스트를 감염시키나 다음에 기술하는 라우팅 웜보다는 늦은 확산 속도를 가지는 것으로 관측되었다.
- 라우팅 웜: 이 웜은 네트워크 안의 경로 정보를 기반으로 선택적으로 IP 주소 공간을 스캔한다. Zou 등이 도입한 분류로, 웜의 스캐닝 공간을 감소시키기 위하여 BGP 라우팅 접두사(prefix)를 이용한다[21]. 보통의 균일 스캔 웜을 라우팅 웜으로 변환하는 것은 스캐닝 전략은 그대로 두고, 웜의 스캐닝 공간을 단지 변경하는 것이다. 그러므로 이 웜도 단순 역학 모델에 의하여 모델 될 수 있다. 예를 들어, 코드 레드에서 라우팅 웜의 감염 확률이 랜덤 스캔을 사용한 웜보다 3.5배 빠르다.
- 분할 및 정복(divide-and-conquer) 스캔 웜: 균일 스캔 웜에서 다른 감염 호스트들이 IP 공간의 다른 부분에 있는 취약 호스트를 스캔하고 감염시키기 위하여 "분할 및 정복" 방법을 사용하는 것이다. 즉, 두개의 감염 호스트가 동일한 타겟에 대하여 그들의 감염력(infection power)을 허비하지 않는 것이다.

3) 지역 우선(local preference) 스캔 웜

이 웜은 감염 호스트가 멀리 있는 주소보다는 더 높은 확률을 가지고 자신의 주소와 가까운 IP 주소를 스캔하는 스캐닝 전략을 사용한다. 이렇게 함으로써 취약 호스트가 더욱 밀집하게 분산된 IP 공간에서 스캐닝 속도를 증가시키고, 방화벽을 고려할 수 있다.

만일 네트워크 혼잡의 영향을 고려한다면, 집중적인 웜 트래픽이 로컬 네트워크에 대하여 혼잡을 야기하고 웜의 확산 속도를 저하시킬 수 있다.

4) 순차적(sequential) 스캔 웜

지금까지의 웜들은 IP 주소를 임의로 선택하는 것을 가정하였지만, 이 웜은 IP 주소를 오름차순이나 내림차순으로 순차적으로 스캔 한다. 많은 취약한 호스트를 가진 네트워크를 일단 스캔하면, 전파는 더욱 효과적이다. 이 방법의 단점은 한 호스트를 반복적으로 스캔할 수 있어 네트워크 트래픽을 차단할 수 있다. 블래스터(Blaster) 웜이 대표적인 예이다.

5) 선택적 공격 웜

IP 주소의 지역적 정보를 기초로 선택적인 공격을 수행하는 웜이다. 선택적 랜덤 스캔이 지역 우선 전

략과 연계된다면, 웜은 더욱 효과적으로 전파될 수 있다. 코드 레드와 슬래머 모두 빠른 확산을 위하여 선택적 랜덤 스캐닝을 사용한다.

이상의 5가지 이외에, DNS 서버로부터 타겟 주소 테이블을 획득하는 DNS 스캔 등이 있다. 그 외에 웜의 외부적인 특성에 따른 분류로, E-Mail 웜, 윈도우 파일 공유 웜, 일반적인 웜 등의 분류도 있다[10].

2.3 특성 분석

웜은 그들의 통상적인 특성에 따라서 분류할 수 있다. 본 절에서는 웜을 탐지하고 방지하기 위하여 사용되는 방어 메커니즘에 따른 분류를 기술한다[1]. 웜이 보여주는 가장 기본적인 특성은, 호스트의 통제를 획득하고, 그 통제를 유지하고, 다른 호스트에 전파되며, 그리고 페이로드를 실행하는 것이다. 이런 요구사항을 기술하기 위하여 웜은 그들이 수행하는 기본적인 생명 기능(life function)에 의하여 분류한다. 다음과 같은 네 가지의 기능에 의하여 웜을 분류 한다:

- 감염(infection)
- 생존(survival)
- 전파(propagation)
- 페이로드(payload)

감염은 웜이 어떤 시스템의 초기 제어를 획득하는 방법을 의미한다. 웜은 호스트를 감염시키기 위하여 보통 두 가지 방법을 사용한다. 시스템 상에 운영되는 소프트웨어 결점을 이용하거나, 혹은 사용자에 의하여 취해진 어떤 행동의 결과이다. [1]에서는 아래와 같이 네 가지 범주의 감염 벡터를 식별하였다:

- 네트워크 인식 코드의 이용 가능한 부분
- 네트워크 인식 컴포넌트의 취약성 있는 구성
- 사용자의 행동
- 기존 시스템 백도어

1988년의 전통적인 모리스 웜처럼 제로데이(zero-day) 익스플로잇을 사용한 웜이 있었지만, 지금까지 거의 모든 웜들은 공개적으로 알려진 취약성을 이용하거나 사용자들을 속여 웜을 실행하는 방법을 이용하였다.

생존 생명 기능은 웜이 호스트의 방어를 일단 침투한 후, 호스트 상에 통제를 유지하는 방법을 기술한다. 이 범주는 아래 행위를 포함한다.

- 시간이 지난 후에 실행 재개
- 탐지 회피
- 탐지 소프트웨어 불능화
- 역공학 방지

다음과 같은 네 가지의 주요한 웜의 전파 방법이 있다.

- 감염 이메일 전송
- P2P 네트워크에 복사본 삽입
- 파일 공유에 복사본 위치
- 원거리 취약 호스트 스캐닝 및 이용

웜의 페이로드는 웜의 표준 생명 주기 기능을 넘어 어떤 일을 수행하기 위하여 수반하는 코드 혹은 패키지이다. 아래와 같은 네 가지의 주요한 페이로드가 발견되었다.

- 백도어 통제 확립
- 분산 서비스 거부 에이전트 확립
- 정보 획득
- 파괴 야기

2.4 공격 속성

웜에 의하여 취해지는 특정한 관측 가능한 행동 특

성을 웜의 공격 속성(attack attribute)이라 부르며, 세 가지 기본 범주로 나눈다.

1) 성공적인 웜 공격을 허용하도록 존재하는 어떤 조건을 의미한다. 예로써 취약 네트워크 서비스 혹은 잘못 구성된 시스템이 있다.
2) 웜이 시스템을 감염시킬 때 남기는 관측 가능한 찌꺼기. 예로써 파일 변경, 윈도우 레지스터리에 대한 변경, 혹은 변경된 프로세스 등이다.
3) 웜 감염으로 인한 부작용에 의하여 생기는 어떤 행위이다. 예로써 웜이 새로운 타겟을 찾는 시도를 할 때 네트워크 트래픽의 증가가 관측된다.

[1]에서는 거의 200개의 식별된 자세한 공격 속성들 중에서 아래와 같은 14개의 일반적인 공격 속성을 분류하였다. 이런 속성들이 과거 웜에 의하여 열거된 속성의 범위를 포함하고 미래 웜에서도 같이 적용될 것으로 믿고 있다. 번호는 5.2절에 있는 〈표 1〉에서 해당 공격 속성을 나타내기 위하여 사용된다.

① 취약 네트워크 코드 이용 : 가장 통상적인 취약성은 버퍼 오버플로 조건이다.
② 사용자 속임 : 이메일 등을 통하여 전달된 웜을 사용자를 속여 실행하도록 한다.
③ 취약 구성 이용 : 결점 코드, 약한 패스워드 설정, 잘못 구성된 신뢰 관계 등을 포함한다.
④ 사전에 설치된 백도어 이용 : 시스템 상의 기존 백도어를 이용한다.
⑤ 파일 시스템 변경 : 거의 모든 웜이 파일 시스템 내에 어떤 증거를 남긴다.
⑥ 시스템 설정 변경 : 대표적으로 이 변경은 웜을 자동으로 수행하기 위한 것이다.
⑦ 프로세스 수정 : 운영 프로세스를 수정하거나 다른 프로세스들을 기동 혹은 정지시킨다.
⑧ 네트워크 접근 : 네트워크상으로 전파되거나 네트워크를 통하여 명령을 수신한다.
⑨ 향상된 특권 요구 : 자원 접근을 위한 충분한 특권을 구한다.
⑩ 비정상 질의 수행 : 어떤 웜은 그들이 감염시킨 시스템으로부터의 정보를 이용한다.
⑪ 중요 API 호출 : 웜은 일반적으로 어떤 중요한 행동을 수행하기 위하여 API들을 호출한다.
⑫ 네트워크 범람 야기 : 공격적으로 전파되는 웜은 이용 가능한 네트워크 대역폭에 영향을 준다.
⑬ 지역 시스템 지연 : 웜은 시스템 응답 시간에 영향을 주거나 많은 양의 로깅 행위를 일으킨다.
⑭ 웜 시그너처 포함 : 새로운 웜을 식별하기 위하여 이전 웜에 대표적인 코딩 패턴을 조사할 수 있다.

III. 전파 특성

3.1 개요

1990년대 초에 IBM에서 질병 역학 모델에 기초한 바이러스성 감염에 대한 일련의 연구를 수행하였다[8,9]. 전통적인 역학 모델은 감염된 호스트가 어떤 다른 취약한 호스트에게라도 똑같이 감염시킬 수 있다는 의미에서 모두 동질성(homogeneous)이다. 지역적인 바이러스의 상호작용을 고려하여, 그러한 동질성 역학 모델들이 랜덤 그래프, 2-차원 격자(lattice) 및 트리 같은 계층적 그래프와 같은 비동질성 네트워크에 대한 역학 모델로 확장되었다[5]. 지역적인 상호 작용에 대한 가정은 현재 대부분의 웜이 인터넷을 통하여 전파되고 목표를 직접 공격할 수 있기 때문에 웜 모델링을 위하여 더 이상 유효하지 않다[14].

2001년 7월 Code Red 웜 사고가 인터넷 웜 전파를 모델하고 분석하기 위한 계기가 되었다. Staniford 등은 사고 발생 후 바로 코드 레드 웜 확산을 모델하기 위하여 고전적인 단순 역학 방정식을 사용하였다[12]. 그리고 코드 레드 웜 전파의 시각적 시뮬레이션, 코드 레드 웜 행위의 관측 데이터와 상세 분석, 웜 설계 원칙[12] 등에 대한 연구가 수행되었다.

웜 모델링에 대한 앞의 연구는 웜 행위에 대한 인적 대응책(human countermeasures)의 동적인 영향을 무시하고 있다. 실제로, 인적 대책이 동적인 활동이며 웜 전파를 늦추고 웜 발생을 방지하는데 주요한 역할을 수행한다. 바이러스나 웜에 대한 인적 대응책은 다음과 같다[19]:

- 감염된 컴퓨터를 청소하기 위하여 항-바이러스 소프트웨어나 특별 프로그램의 사용.
- 취약한 컴퓨터를 바이러스나 웜에 면역적이도록 패칭이나 업그레이딩.
- 바이러스나 웜 트래픽을 여과하거나 차단하기 위하여 방화벽이나 라우터에 필터 설정.
- 유효한 방법이 없을 때 네트워크나 컴퓨터의 연결 해지.

3.2 전파 모델

웜 네트워크는 공격을 위하여 새로운 호스트를 적극적으로 찾아서 네트워크상의 집합 노드에 추가한다. 웜이 호스트를 발견하고 공격할 때, 웜 네트워크는 지수적으로 성장한다. 이 성장 패턴은 박테리아나 바이러스 같은 일반 역학모델에서의 패턴과 동일하다.

웜 감염은 처음에는 빠르게 지수적인 형태로 성장하고 안정기 값에 도달하면 느려진다. 이것은 다음과 같은 일차 방정식에 의하여 기술될 수 있는 대표적인 동역학 모델이다.

$$Nda = (Na)K(1-a)dt \tag{1}$$

이것을 미분방정식 형태로 쓰면 (2)와 같이 된다.

$$\frac{da}{dt} = Ka(1-a) \tag{2}$$

이것은 웜의 임의 고정 확산률(spread rate)을 나타낸다. 이 미분방정식을 풀면 (3)과 같다.

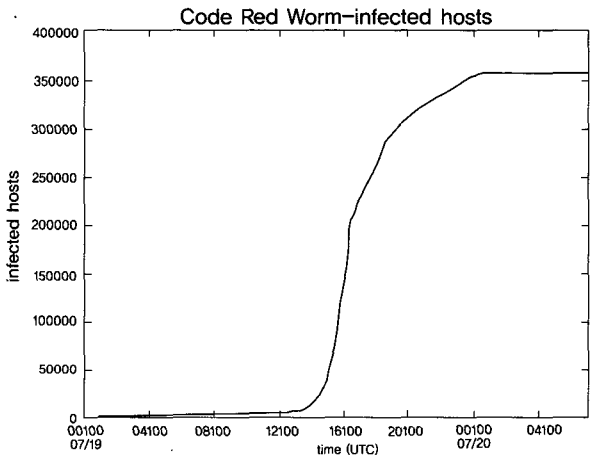
$$a = \frac{e^{K(t-T)}}{1+e^{K(t-T)}} \tag{3}$$

여기서 a는 침해된 취약 호스트의 비율, t는 시간, K는 초기 침해 율, 그리고 T는 성장이 시작된 고정 시간을 의미한다. 율 K는 이미 감염된 호스트를 고려하여 조정되어야 하며, 이 된다.

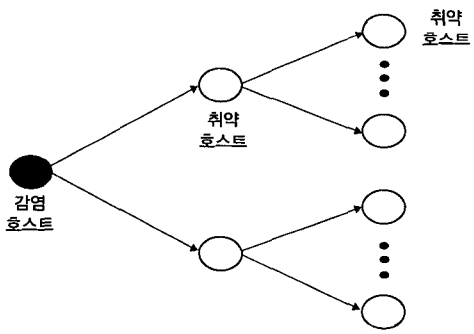
이 방정식을 logistic growth model이라 하며, 네트워크 웜에 대하여 볼 수 있는 성장 데이터의 핵심이다. 더 복잡한 모델이 유도될 수 있지만, 대부분의 네트워크 웜은 이 경향을 따른다. 이 모델을 사용하여 웜의 성장률에 대한 측정치를 얻을 수 있다. Nimda와 Code Red같은 웜은 매우 높은 율 상수 K를 가진다. 이것은 시간당 많은 호스트를 침해할 수 있다는 것을 의미한다[11].

(그림 2)는 시간에 따른 감염 호스트 수의 변화를 보여준다. (그림 2)에서 보면 웜의 전파를 세 가지 단계로 대략 구분할 수 있다[23]: 늦은 시작 단계, 빠른 확산 단계, 늦은 종료 단계. 늦은 시작 단계에서는 감염 호스트의 수는 지수적으로 증가한다.

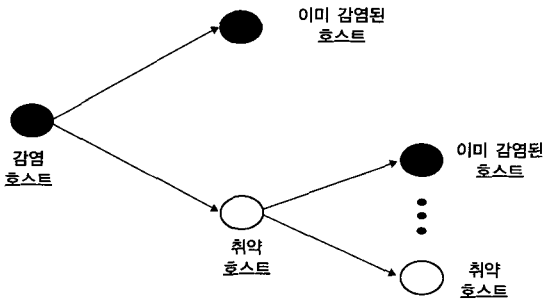
(그림 3 (a))에서는 웜의 일반적인 단계별 전파 특성을 보여준다. 빠른 확산 단계에서 감염 호스트는 많은 취약 호스트를 스캔하게 되고 결과적으로 감염 호스트의 지수적인 증가를 가져온다.



(그림 2) 웜의 대표적인 전파 곡선



(a) 확산 단계에서의 웜의 전파



(b) 종료 단계에서의 웜의 전파

(그림 3) 웜의 단계별 전파 특성

(그림 3 (b))에서는 종료 단계에서 웜의 전파 특성을 보여준다. 많은 취약 호스트들이 감염되고 나면, 전파 율은 감소되기 시작하고, (그림 2)의 늦은 종료 단계에 들어간다.

고전적인 단순 역학 모델에서, 각 호스트는 두 상태 중 하나에 있다: 취약적(susceptible) 혹은 감염성(infectious). 일단 호스트가 바이러스에 의하여 감염되면, 감염 상태에 영원히 머무른다고 이 모델은 가정한다. 그리하여 호스트의 상태 전이는 취약적에서 감염성으로의 일 방향만 가능하다.

유한 모집단에 대한 고전적 단순 역학적 모델은 식 (4)와 같다.

$$\frac{dJ(t)}{dt} = \beta J(t)\left[N - J(t)\right] \qquad (4)$$

여기서, $J(t)$는 시간 t 에서 감염된 호스트의 수이고; N은 모집단 크기, β는 감염률이다. 처음 t=0에서, $J(0)$호스트가 감염성이고 다른 N-$J(0)$ 호스트가 모두 취약적이다. $a(t) = J(t)/N$을 시간 t에서 감염성인 모집단의 부분이라고 하자. 식 (4)의 양변을 N^2으로 나누면, 다음 식 (5)가 유도된다.

$$\frac{da(t)}{dt} = ka(t)\left[N - a(t)\right] \qquad (5)$$

식 (5)는 1-$a(t)$가 대략 1과 같은 초기에는 감염 호스트의 수가 거의 지수적으로 증가됨을 보여준다.

이 모델은 균일 스캔 웜의 메커니즘을 반영할 수 있다[9]. 특별히 인간적 대응책과 혼잡의 영향이 무시될 수 있는 웜 전파의 초기 부분에 유효하다.

KM(Kermack-Mckendrick) 모델은 감염성 호스트의 제거 과정을 고려하고 있다[5]. 전염병의 감염동안 어떤 감염성 호스트는 복구하든지 죽는다. 호스트가 일단 질병으로부터 회복되면, 그 질병에 대하여 영원히 면역적이라고 가정한다. 해당 호스트가 그 질병으로부터 복구되든지 죽은 후에 "제거" 상태에 놓이게 된다. 그리하여 각 호스트는 어떤 때라도 세 상태 중 하나에 있게 된다: 취약 상태, 감염 상태, 제거 상태. 시스템의 어떤 호스트도 "취약 → 감염 → 제거"의 상태 전이 혹은 "취약" 상태에 영원히 머무르게 된다.

$I(t)$를 시간 t에서 감염 호스트의 수라 정의하자. $R(t)$를 시간 t에서 이전 감염 호스트로부터 제거된 호스트의 수라고 하자. 시간 t에서 감염 집단으로부터 제거된 호스트는 일단 감염되었으나 시간 t전에 소독되거나 유통에서 제거된 호스트를 의미한다. $J(t)$를 감염 상태에 있거나 제거되었거나 관계없이, 시간 t까지의 감염 호스트의 수라고 하자. 그러면, 식 (6)이 성립된다.

$$J(t) = I(t) + R(t) \tag{6}$$

단순 역학 모델 (4)에 기반, KM 모델은 식 (7)과 같이 나타낼 수 있다.

$$\begin{cases} dJ(t)/dt = \beta J(t)\left[N - J(t)\right] \\ dR(t)/dt = \gamma I(t) \\ J(t) = I(t) + R(t) = N - S(t) \end{cases} \tag{7}$$

여기서 β는 감염률이고, γ는 유통에서 제거된 감염 호스트의 제거율, $S(t)$는 시간 t에서 취약 호스트의 수, N은 모집단의 크기이다.

코드 레드 사고 이후, Zou 등은 코드 레드 웜 전파에 영향을 미친, 전통적인 역학 모델에서는 고려되지 않았든 두 요인을 발견하였다: 인간 대응책과 감소되는 감염률. $R(t)$를 감염 집단으로부터 제거된 호스트의 수라 정의하고, $Q(t)$를 취약 집단으로부터 제거된 호스트의 수라 정의한다. KM 모델(4)를 유도하는데 사용한 같은 원칙에 의하면, 시간 t에서 시간 $t+\Delta t$까지의 취약 호스트 $S(t)$의 수 변화는 다음과 같이 된다:

$$S(t+\Delta t) - S(t) = -\beta(t)S(t)I(t)\Delta t - \frac{dQ(t)}{dt}\Delta t \tag{8}$$

그러므로

$$\frac{dS(t)}{dt} = -\beta(t)S(t)I(t) - \frac{dQ(t)}{dt} \tag{9}$$

어떤 시간 t에 대해서도 $S(t)+I(t)+R(t)+Q(t)=N$이 성립되므로, $S(t)$의 값을 식 (9)에 대체하면 감염 호스트 $I(t)$ 수의 행위를 기술하는 (10)과 같은 미분 방정식이 만들어진다.

$$\frac{dI(t)}{dt} = \beta(t)\left[N - R(t) - I(t) - Q(t)\right] \\ I(t) - \frac{dR(t)}{dt} \tag{10}$$

식 (10)에 의하여 기술되는 웜 모델을 Two-factor 모델이라 한다.

AAWP(Analytical Active Worm Propagation)

모델은 랜덤 스캐닝을 채택하는 웜의 전파를 특성화 한다[4]. 이 모델은 이산 사건 및 연속 상태 결정적 근사화 모델을 사용한다. 본 논문에서는 역학 모델과 비교하여 기술하며, 보다 자세한 내용은 [4]를 참조할 수 있다.

AAWP 모델과 역학 모델과의 차이는 아래와 같다 [19]:

- 역학 모델은 연속 시간 미분 방정식을 사용하는 반면, AAWP 모델은 이산 시간 모델을 기반으로 한다.
- 역학 모델은 패칭율이나 웜이 머신을 감염시키기 위하여 걸리는 시간을 고려하지 않지만, AAWP 모델은 고려한다.
- AAWP 모델에서는 웜이 동시에 동일한 목적지를 감염시킬 수 있는 경우를 고려하지만, 역학 모델은 이런 경우를 무시한다.

위에서 기술한 모델들 이외에 SIS(Susceptible-Infectious-Susceptible) 모델이 있다[9]. 이 모델에서는 모든 호스트가 반복적으로 감염될 수 있는 같은 확률을 가진다고 가정한다. 그러나 이 모델은 감염된 호스트가 웜으로부터 면역적이기 위하여 패치되고 갱신되는 것을 고려하지 않는다. 그러므로 SIS 모델은 웜의 감염 모델로 적합하지 않다.

3.3 BcN에서의 전파 특성

코드 레드나 슬래머와 유사한 웜은 고속망에서 훨씬 높은 감염률을 가질 수 있고 타겟 집단을 더욱 빠르게 포화시킬 수 있다. 고속망에서는 감염된 호스트가 잠재적인 타겟과 통신하는 것을 용이하게 만들어 (5)식에서 $\beta(1-\alpha(t))$를 증가시킬 수 있다. 단순 역학 모델은 다음과 같이 재배열 될 수 있다:

$$T_p = \frac{\ln P(N - J(0) - \ln(1 - P)J(0)}{\beta N} \quad (11)$$

여기서 T_p는 모집단의 P 부분, 즉 PN 호스트를 감염시키기 위하여 걸리는 시간을 나타낸다. 이 결과는 만약 웜이 탐사 율(probe rate)을 두 배로 하기위한 대역폭을 발견한다면, 즉 감염 파라미터 β를 실제적으로 두 배로 한다면, 반 정도의 시간으로 목표 집단을 포화시킬 수 있다는 것을 의미한다[2]. 따라서 고속망에서 웜은 더 높은 감염률을 얻을 수 있고 목표 집단을 더욱 빠르게 포화시킬 수 있다.

IV. 웜의 탐지 기술

4.1 시그너처 기반 탐지

대부분의 침입탐지 및 방지 시스템에서 적용하는 방식으로 이미 알려진 웜에 대한 탐지를 위하여 기존의 패턴 매칭 방식을 사용하는 것이다. 웜이 유포되고 난 후 이들에 대한 정보를 수집하여 이 정보를 기반으로 웜에 대한 시그너처를 생성한 후 이것을 가지고 탐지하는 방식이다. 예를 들어, 코드레드 웜의 경우 ASCII 값으로 이루어진 일정한 패턴이 발생한다. 이것이 시그너처가 되어 나중에 다른 패킷들 안에서 동일한 비트 패턴이 발견되면 코드 레드 웜으로 간주하게 된다[25]. 공개 보안 도구인 Snort를 비롯하여, 대부분의 상용 시스템에서 채택하고 있는 방식이다.

이 방식은 오탐율이 낮게 웜을 탐지할 수 있는 장점은 있으나, 이미 알려진 웜에 대해서만 탐지가 가능하고 새로운 웜에 대하여는 탐지할 수 없기 때문에, DoS 공격 등을 통하여 네트워크를 동시에 마비시키는 웜들에 대하여 효과적으로 대응 할 수 없다는 문제가 있다.

4.2 트래픽 기반 탐지

웜 트래픽은 지속적인 증가와 반복적인 성질로 인하여 특성화 될 수 있다. 네트워크 기반 침입탐지 시스템에서 사용되는 탐지 엔진을 위하여 앞에서 기술한 시그너처를 구축한다. 이와 더불어 트래픽 특성을 조사하고 그들의 동향을 감시함으로써 더욱 융통성 있게 웜을 탐지 할 수 있다.

트래픽 분석은 네트워크 통신과 그 속에 내재되어 있는 패턴을 분석하는 것을 말한다. 연구될 트래픽의 특성으로는 프로토콜, 연결에 사용된 포트, 연결의 성공과 실패, 통신 상대, 시간상 및 호스트 당 트래픽 양 등을 포함한다. 웜에 대한 감시를 위하여 트래픽 분석 관점에서 세 가지 관심 있는 주요한 특징으로 트래픽 양, 발생하는 스캔 형태의 수, 어떤 호스트가 웜 네트워크의 일부일 때 트래픽 패턴의 변화가 있다. 이와 같이 네트워크 웜의 성장과 재생을 모델하는 것이 가능하다. 성장 패턴은 어떤 한 시점에서의 감염율과 취약 호스트의 수에 의하여 지배된다. 비슷하게, 웜 스캔과 공격에서의 트래픽 패턴도 어느 시간에서 활성 웜의 수와 노드 당 트래픽 양에 의하여 결정된다. 이 방법은 이미 알려진 웜보다는 알려지지 않은 새로운 웜을 발견하기 위하여, 네트워크상의 트래픽 특성을 분석하고 이들 중 웜으로 생각되는 트래픽 패턴을 찾아서 웜의 공격을 식별하는 것이다. 아직 오탐률이 상대적으로 높은 편이고 구현이 쉽지 않다는 단점이 있다.

트래픽 특성 분석을 통한 웜의 탐지 방법에 대하여 현재 많은 연구가 세계적으로 진행되고 있다. 대표적으로 국외의 TRW(Threshold Random Walk) 방법[7], DEWP(Detecting Early Worm Propagation) 방식[3], 통계적 침입탐지 방식[20], 국내의 ETRI에서 개발한 CPD(Change Point Detection)[25] 방식 등이 있다.

4.3 하니팟

하니팟(honeypot)은 공격에 의하여 요구되는 대응을 이끌어내는 방법으로 악성 탐사(probe)에 대응하는 기능 시스템으로 정의할 수 있다.

이것은 전체 시스템, 단일 서비스, 혹은 가상 호스트를 사용하여 구축될 수 있다. 네트워크 하니팟은 탐사되거나 공격되기를 기다려 관련 자료를 분석할 수 있는 시스템이다.

Spitzner의 정의에 의하면, 하니넷(honeynet)은 하니팟의 네트워크이다. 만일 하니폿을 포함하여, 네트워크상의 호스트를 웜이 공격하면, 공격에 대하여 추후 분석할 수 있고 공격 에이전트에 의하여 사용된 방법에 대하여 알 수 있다. 하니팟은 다음과 같은 세 가지 형태가 있다[11]:

- 완전 전용 시스템: 대표적으로 어떤 운영체제가 견고하지 않게 설치된 시스템이다. 기본 설치를 반영하기 위한 시도로 최소한의 설정으로 설치되어 네트워크상에 위치한다. 호스트 입출력 네트워크 트래픽을 잡기 위하여 외부 모니터가 사용된다.
- 서비스-레벨 하니팟: 보호된 프로세스와 메모리 공간 영역에 한 개 이상의 서비스가 설치된 호스트를 말한다. 공격자는 서비스를 탐사 및 공격할 수 있으나, 침해는 호스트 상에 수행되고 있는 가상 머신에 제한된다.
- 가상 호스트 및 네트워크: 공격자에 대하여 호스트와 관련된 서비스로 착각하게 하는 것을 말한다. 이것은 대표적으로 다른 호스트를 위장하여 네트워크상의 한 개의 호스트에 수용된다.

보다 자세한 내용에 대하여는 [24]를 참조할 수 있다.

V. 웜 대응 기술

5.1 종류

[1]에서는 바이러스 스캐닝이나 소프트웨어 패칭과 같은 수동적인 웜의 방어를 제외하고 현재 웜을 방지하기 위하여 사용될 수 있는 능동적인(proactive) 웜 방어 기술을 제시하고 있다. 또한 가상 사설 근거리망과 같은 OSI 계층 2와 3 방어 메커니즘도 고려하지 않고 있다. 아래에 11가지의 웜 방어 기술에 대하여 간단히 기술한다.

- 패킷 필터링 방화벽(PFFW): 패킷의 네트워크 혹은 트랜스포트 헤더를 정해진 룰 셋에 대하여 조사한다.
- 스테이트풀(stateful) 방화벽(SFW): 스테이트풀 방화벽은 네트워크 연결을 추적하고 그들의 상태를 감시한다.
- 애플리케이션 프락시 방화벽(APFW): 이 방화벽은 애플리케이션 프로토콜의 콘텐츠 조사와 검정을 할 수 있다.
- 침입탐지시스템(IDS: Intrusion Detection System): 연결된 네트워크 세그먼트 상의 모든 트래픽을 감시하여 시그너처 데이터베이스에 대하여 조사한다.
- 호스트 방화벽(HFW): 호스트 상의 애플리케이션과 네트워크 사이에 위치하는 방화벽으로, 사용자의 정상 네트워크 패턴을 감시하여 룰셋을 개발하고 비정상적인 연결을 만드는 웜을 차단한다.
- 가상 머신(VM): 대표적으로 운영체제와 물리적인 하드웨어 사이에 위치하며, 악성 소프트웨어가 불법 행동을 위하여 운영체제를 사용하는 것을 막기 위하여 사용될 수 있다.
- 구성(configuration)(Conf): 여기서 구성은 애플리케이션이나 운영체제 환경을 공격에 더욱 저항적으로 만들기 위하여 조정될 수 있는 설정을 의미한다. 강한 구성의 대표적인 측면은 애플리케이션 설정을 단단히 하고, 네트워크 포트 사용을 제한하며, 단지 필요한 서비스를 운영하며, 파일 시스템과 레지스리에 대하여 가장 제한적이고 입상적인 허용(granular permission)을 설정하는 것이다.
- 항-바이러스 휴리스틱(AVH): 많은 항-바이러스 제품은 악성 코드를 식별하기 위하여 시그너처를 사용하고 있으며, 또한 악성 코드를 탐색하는데 휴리스틱을 채택하고 있다. 항-바이러스 제품은 알려진 웜과 바이러스와 유사한 방법으로 기능하는 새로운 악성 코드를 식별할 수 있다.
- 호스트-기반 침입방지시스템 (HIPS: Host-based Intrusion Prevention System): HIPS는 특정 애플리케이션에 대한 합법적인 행위를 구성하는 것을 기술하는 미리 정해진 룰로부터 보통 운용된다.
- 무결성 검사(IC): 파일의 알려진 좋은 인스턴스에 대한 암호 해쉬를 보관하여 어느 때라도 무결성 비교가 행해질 수 있다.
- stackguarding(SG): 이 기술은 프로그램이 버퍼 오버플로 공격에 저항적이도록 하기위한 것이다.

5.2 웜의 방어

본 절에서는 앞에서 기술한 각 공격 속성에 대하여 사용할 수 있는 방어 기술을 나열한다. 〈표 1〉에서 공격 속성과 방어의 교차부분은 그 공격 속성을 다루기 위하여 방어가 제공하는 보호의 종류를 나타낸다. 이 종류로는 다음과 같이 4가지가 있다.

- D(탐지): 방어 시스템이 공격을 탐지만 할 수 있

고 그것을 막기 위하여 아무것도 할 수 없다.

- P(부분적인 보호 제공): 어떤 공격을 방지하는 데 좋지만, 공격의 구체적인 구현 세부사항에 따라서 방어를 피할 수 있음을 의미한다.
- R(수동적인 보호): 방어 시스템이 공격을 탐지하고 없앨 수 있지만, 단지 공격이 알려진 후에 가능하다. 예로써 일반적인 시그너처-기반 침입탐지시스템이 있다.
- B(차단): 방어 시스템이 공격을 효과적으로 차단한다.

빈 항목은 방어 시스템이 공격 속성에 대하여 효과적인 보호를 제공할 수 없음을 나타낸다.

〈표 1〉의 방어 메트릭스에 의하면, 스택보호(SG) 기술을 채택하는 것이 논리적인 첫 단계임을 제시한다. 이렇게 함으로써 웜에 의하여 이용되는 대부분의 취약성을 다룰 수 있다. 방화벽이 네 가지의 감염-관련 속성에 대하여 가장 포괄적인 보호를 제공함을 또한 나타낸다. 그러나 방화벽은 시스템에 "합법적으로" 입장하는 파일과 그런 후 사용자를 속여 스스로 감염시키는 것에 대하여는 아주 제한된 보호를 제공한다. 또한 노출된 네트워크 인터페이스 안의 알려지지 않은 로직 결함에 대하여는 보호하지 못한다. 그리하여 페리미터(perimeter) 방어만으로는 웜-기반 공격을 완전히 방어하기에는 충분하지 않다.

나머지 10개의 공격 속성들은 호스트-기반 침입방지시스템과 적절한 시스템 구성에 의하여 가장 잘 방어될 수 있다. 이 두 가지의 방어 기술은 10개의 비-감염 속성 중에서 9개에 대하여 차단하거나 방어할 수 있고, 또한 전체 14개의 항목 중에서 12개를 포함하고 있다. HIPS와 구성은 웜의 후반 생명 단계가 수행되는 호스트 애플리케이션과 운영체제 레벨에서 동작한다. 방어 매트릭스에 있는 방어 기술을 선택함으로써, 모든 형태의 공격 속성들에 대처할 수 있는 다중-계층 보호 스킴을 만드는 것이 가능하다고 지적하고 있다[1].

VI. IPv6의 영향

6.1 개요

IPv4에서의 제한된 주소 공간과 다른 기본적인 결점을 극복하기 위하여, 새로운 인터넷 프로토콜 버전인 IPv6가 개발되었다. 128 비트의 주소길이를 가진 IPv6는 미래에 기대할 수 있는 모든 연결된 기계에

〈표 1〉 세부 방어 메트릭스[1]

속성	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭
PFFW	R		B	B				P				B		
SFW	R		B	B				P				B		
APFW	B	B	B	B				P				B		P
IDS	R													P
HFW	R	B	B	B				B				B		
VM											B			
Conf	B	P	B		B	B	B		B	B				
AVH				B										B
HIPS					B	B	B	B	B	B	B	B	P	
IC					D	D								
SG	B								B					

거의 무한대의 주소 풀을 쉽게 제공할 수 있다. 이런 영향으로 일반적으로 IPv6에서는 매우 흩어진 주소 공간으로 인하여 랜덤-스캐닝 웜에 대하여 더 큰 보호를 제공할 것이라고 알려져 있다. 이와 같은 낙관적인 측면에서 보면, 모든 빠르게 전파하는 웜은 어떤 형태의 스캐닝을 사용하기 때문에, IPv6를 기반으로 하는 BcN에서 이런 웜이 사라질 것이라는 생각이다. 그러나 수행된 연구에 의하면, IPv6에서도 IPv4와 마찬가지로 어떤 스캐닝 전략을 사용하는 지능적인 웜은 빠른 속도로 전파될 수 있음을 보여준다[13,18].

6.2 침입탐지에 대한 영향

현재 많은 침입탐지시스템들이 개발되어 있지만, 아직 IPv6 프로토콜을 지원하기 위하여 현재의 IDS를 확장하기 위한 노력은 아주 적은 편이다. 본 절에서는 웜의 탐지를 위하여 사용되는 IDS가 IPv6에 대하여 어떤 영향을 받을 것인가에 대하여 기술한다[13].

IPv6와 IPv4사이의 주요 차이와 그 영향은 아래와 같이 요약될 수 있다.

- 단순화된 헤더: IPv6에서 헤더와 확장 헤더의 여러 필드의 분해(decomposition)가 효율적으로 일어날 수 있다. 이것은 패킷-당 처리 속도에서의 이득을 의미하며 기가급의 인터페이스에서 중요한 척도이다. 성능은 헤더 내에 단편화 정보가 없음(확장 헤더에 의하여 취급)으로 인하여 더욱 증진된다. 그러나 IPv4를 위하여 개발된 단편화 공격은 부정확한 데이터그램 재결합이 여전히 발생할 수 있기 때문에 쓸모없게 되지는 않는다.
- 대규모 주소 공간: 대규모 주소의 활용으로 모든 장치가 인터넷에 연결되는 유비쿼터스 네트워킹 시대가 열릴 것이다. 특히 홈네트워크의 구현으로 여러 가전기기들이 인터넷에 연결된다. 이런 어플라이언스들은 통신을 위하여 TCP/IP를 가진 임베디드 운영체제를 실행할 것이다. 만일 이런 장치들의 소프트웨어에 보안 결점이 존재한다면, 원격 사용자로 하여금 통제를 허용하게 되고 분산 서비스 거부(DDOS) 호스트로 사용할 수 있다. 이런 홈네트워크 환경이 널리 사용됨에 따라 헤아릴 수 없을 만큼의 단순 시스템에 의한 공격의 가능성에 직면할 수도 있다.
- 내재된 인증 및 암호 패킷-레벨 지원: 이런 IPv6의 내장된 기능으로 인하여 "암호화된 공격 문제"를 아주 급격하게 악화시킬 가능성이 있다는 분석이 있다[8]. 인증된 암호화된 링크에 의하여 운반되는 트래픽이 합법적이라는 보장이 없고 게다가 NIDS(Network-based IDS)에 관한한 불법적이라는 것이다. 이것이 NIDS에 IPv6를 도입하기 위하여 가장 중요한 어려움으로 지적하고 있다. 왜냐하면, 정교한 NIDS는 ESP의 내용을 조사하지 않더라도 각 패킷 내의 AH의 유효성을 적어도 검정하기를 원한다. 그런데 SSL과 SSH 키 생성을 위한 그리고 메가비트/초 율에서 on the fly로 패킷을 복호화하기 위하여 지원되는 고속 암호화 사이에 분명한 차이가 있기 때문이다[13].
- 단순화된 경로배정
- 헤더 내의 체크섬 없음
- 헤더 내에 단편화 정보 없음

VII. 맺음말

웜이 인터넷 보안과 안전성에 심대한 위협을 주고 있기 때문에, 웜의 확산을 방지할 수 있는 방안에 대

한 연구가 필요하다. 더구나 인터넷 망이 광대역 통합망 환경으로 진화함에 따라서 인터넷 자원이 네트워크 위협에 노출될 수 있는 위협이 증대되고 있다. BcN은 유무선, 방송 및 통신이 융합되는 정보통신 환경에서 광대역 멀티미디어 서비스를 고품질로 이용할 수 있는 차세대 통합 네트워크이며 디지털 홈네트워킹을 통한 유비쿼터스의 실현을 목표로 하고 있다.

그러나 역설적으로, BcN과 같은 초고속 통합망 환경에서는 네트워크 자원이 여러 가지의 침입 행위에 쉽게 노출될 수 있으며, 네트워크 대역폭의 증가로 전송 속도가 빨라져 웜의 확산을 가속화 시킬 수 있다. 그러므로 웜의 전파에 대응할 수 있는 시간도 단축된다. 따라서 본 논문에서는 인터넷 웜의 탐지 및 대응기술에 대하여 기술하였다. BcN 환경에서 웜의 전파 특성 변화와 침입탐지에 대한 IPv6의 영향에 대하여도 살펴보았다. 향후 이에 대한 보다 체계적인 연구가 필요하다고 하겠다.

[참 고 문 헌]

[1] David J. Albanese, Michael J. Wiacek, Christopher M. Salter, and Jeffrey A. Six, The Case for Using Layered Defenses to Stop Worms, Report #C43-002R-2004, Version 1.0, June 18, 2004, National Security Agency.

[2] Thomas M. Chen, Jean-marc Robert, "Worm Epidemics in High-Speed Networks", IEEE Computer, pp48-53, June 2004.

[3] Xuan Chen and John Heidemann, Detecting Early Worm Propagation through Packet Matching, Technical Report ISI-TR-2004-585, 2004.

[4] Z. Chen, L. Gao and K. Kwiat, "Modeling the Spread of Active Worms", In Proc. of IEEE Infocom, 2003.

[5] J. C. Frauenthal, Mathematical Modeling in Epidemiology, Springer-Verlag, New York, 1980.

[6] Glenn Gebhart, Worm Propagation and Countermeasures, SANS Institute, 2004.

[7] J. Y. Jung, S. Schechter, and Arthur W. Berger, "Fast Detection of Scanning Worm Infections", RAID 2004, Sophia Antipolis, France, Sep. 2004.

[8] J. O. Kephart and S. R. White, "Directed-graph Epidemiological Models of Computer Viruses", Proc. of the IEEE Symposium on Security and Privacy, pp343-359, 1991.

[9] J. O. Kephart, D. M. Chess and S. R. White, "Computers and Epidemiology", IEEE Spectrum, 1993.

[10] D. M. Kienzie and M. C. Elder, "Recent Worms: A Survey and Trends",

[11] Jose Nazario, Defense and Detection Strategies against Internet Worms, Artech House, 2004.

[12] S. Staniford, V. Paxon and N. Weaver, "How to Own the Internet in Your Spare Time", 11th Usenix Security Symposium, San Francisco, August, 2002.

[13] Arrigo Triulzi, "Intrusion Detection Systems and IPv6", SPI2003.

[14] N. Weaver. Warhol Worms: The Potential for Very Fast Internet Plagues, http://www.cs.berkeley.edu/~nweaver/warhol.ht

ml

[15] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "A Taxonomy of Computer Worms", ACM WORM' 03, Washington DC, USA, Oct. 2003.

[16] Webopedia, "Virus", Webopedia URL: http://www.webopedia.com/TERM/v/virus.html, Feb. 2005.

[17] Webopedia, "Worm", Webopedia URL: http://www.webopedia.com/TERM/W/worm.html, Feb. 2005.

[18] Jing Yang, "Fast Worm Propagation in IPv6 Networks".

[19] Cliff Changchun Zou, Weibo Gong, Don Towsley, "Code Red Worm Propagation Modeling and Analysis", 9th ACM Conference on Computer and Communication Security (CCS' 02), Washington, DC, USA, Nov. 2002.

[20] Cliff Changchun Zou, Weibo Gong, Don Towsley, "Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense", ACM WORMS' 03, Washington DC, USA, Oct. 2003.

[21] C. C. Zou, D. Towsley, W. Gong, and S. Cai, Routing Worm: a Fast, Selective Attack Worm based on IP Address Information, Univ. of Massachusetts Tech. Report TR-CSE- 03-06, Nov. 2003.

[22] Cliff Changchun Zou, Don Towsley, Weibo Gong, On the Performance of Internet Worm Scanning Strategies, Tech. Report: TR-03-CSE-07, Department of Computer Science, Univ. of Massachusetts, Amherst, USA.

[23] Cliff Changchun Zou, Lixin Gao, Weibo Gong, Don Towsley, "Monitoring and Early Warning for Internet Worms", 10th ACM Conference on Computer and Communication Security (CCS' 03), Washington, DC, USA, Oct. 2003.

[24] http://www.tracking-hackers.com

[25] 신승원, 오진태, 김기영, 장종수, "인터넷 웜 공격 탐지 방법 동향", 전자통신동향분석, 제 20권 제 1호, pp.9-16, 2005년 2월.

전용희

1978년 고려대학교 전기공학과 졸업(공학사)
1985년 ~ 1987년 미국 플로리다공대 대학원 컴퓨터공학과
1989년 미국 노스캐롤라이나주립대 대학원 Elec. and Comp. Eng. 졸업(공학석사)
1992년 미국 노스캐롤라이나주립대 대학원 Elec. and Comp. Eng. 졸업(공학박사)
1978년 ~ 1978년 삼성중공업(주) 근무
1978년 ~ 1985년 한국전력기술(주) 근무
1979년 ~ 1980년 벨기에 벨가톰(Belgatom)사 연수
1989년 ~ 1989년 미국 노스캐롤라이나주립대 Dept of Elec. and Comp. Eng. TA
1989년 ~ 1992년 미국 노스캐롤라이나주립대 부설 CCSP(Center For Comm. & Signal Processing) RA
1992년 ~ 1994년 한국전자통신연구원 광대역통신망연구부 선임 연구원
1994년 ~ 현재 대구가톨릭대학교 컴퓨터 · 정보통신공학부 교수
2001년 ~ 2003년 동 공과대학장 역임
2004년 ~ 2005년 한국전자통신연구원 정보보호연구단 초빙연구원
관심분야 : BcN 보안 및 QoS 보장 기술, 네트워크 보안, 통신망 성능분석