정보통신 인프라의 웜 전파 분석 및 모델링

대구가톨릭대학교 컴퓨터정보통신공학부 전용희

차 례

1. 서 론

인터넷 웜의 효시인 모리스(Morris)가 1988년에 알려진 이후, 인터넷 웜은 네트워크 보안 연구의 주요한 문제가 되었다. 2001년 7월 코드 레드 웜의 발생으로, 인터넷 웜은 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.

우리나라에서는 2003년 11월 차세대 정보통신망의 근간이 될 광대역통합망(BcN: Broadband convergence Networks) 구축 계획안이 발표되었다. BcN은 유무선, 방송 및 통신이 융합되는 정보통신 환경에서 광대역 멀티미디어 서비스를 고품질로 이용할 수 있는 차세대 통합 네트워크이며 디지털 홈 네트워킹을 통한 유비쿼터스의 실현을 목표로 하고 있다. 그러나 역설적으로, 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.

미래의 웜에 대한 방어를 위하여, 웜의 수명동안 전파(propagation) 형태, 패칭(patching)의 영향, 인식(awareness) 및 다른 인적 대책, 네트워크 트래픽, 네트워크 토폴로지의 영향 등 웜의 여러 가지 특성을 이해할 필요가 있다[14].

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[13]:

- 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

본 고에서는 인터넷 웜에 대한 전파 형태를 분석하고 모델링에 대하여 고찰하고 기술하고자 한다.

2. 웜 전파 분석

2.1 코드 레드(Code Red) 웜

윈도우 IIS(Internet Information Server) 인덱스 서비스 DLL의 버퍼 오버플로우 버그를 이용한 웜이 2001년 7월 본격적으로 확산되기 시작하였다. 이 웜은 TCP 포트 80번을 이용하여 취약 호스트를 스캔한다. 잠재적인 타겟과 TCP 연결 설정에 따른 지연을 보상하기 위하여, 복수 쓰레드를 채택하였다. 이것은 100개의 쓰레드(thread)로 구성되어 있으며, 99개의 쓰레드는 임의로 IP 주소를 선택하여 타겟 머신에 포트 80 번상으로 연결 설정을 시도한다. 만약 연결이 성공적이면, 웜은 침해를 목적으로 희생(victim) 웹 서버에게 자신의 복사를 전송하고 다른 웹 서버를 계속하여 발견한다. 이와 같이 코드 레드 웜은 병렬성을 통하여 감염률을 빠르게 한다. 연결이 설정되지 않거나 목표가 웹 서버가 아닌 경우, 웜 쓰레드는 probe 하기 위한 다른 IP 주소를 임의로 생성하는 랜덤 스캐닝(random scanning)을 사용한다.

7월의 코드 레드 사고에 대한 세 개의 독립적인 관측 데이터가 있다. Goldsmith와 Eichman은 두 개의 클래스 B 네트워크에 대한 다음과 같은 두 가지 형태의 데이터를 독립적으로 수집하였다[2, 4]:

- 매 시간동안 코드 레드 웜 포트 80 스캔의 수
- 매 시간동안 이런 스캔을 생성한 유일한 소스들의 수

각 감염 컴퓨터가 99개의 동시 스캔을 생성하기 때문에, 웜 스캔의 수가 유일한 소스의 수보다 크지만, 시간에 따라서 동일한 전개를 보여주는 것으로 밝혀졌다. 이것들이 인터넷상의 코드 레드 전파를 대표할 수 있다.

Moore 등은 7월 19일 하루 종일동안 코드 레드 웜에 대하여 수집된 다른 자료 집합을 제공한다[10]. 그들은 네트워크에 웜을 확산하기 위하여 모든 감염 호스트의 첫 번째 시도 시간을 기록하였다. 그리하여 감염 호스트의 수는 시간에 대한 비감소 함수(non-decreasing function)임을 보여준다. 관측된 감염 호스트의 수가 시간 t의 함수로 그림 1에서 보여준다.

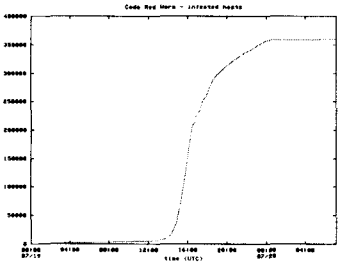

그림 1. 관측된 코드 레드 전파 - 감염 호스트의 수
(출처: www.caida.org)

코드 레드웜이 7월 20일 00:00 UTC 이후에 확산을 정지하도록 프로그램되었기 때문에, 감염 호스트의 수가 00:00 UTC 이후에는 증가를 멈춤을 볼 수 있다.

2.2 Slammer 웜

코드 레드와 마찬가지로, SQL 슬래머 웜도 마이크로소프트사의 SQL 서버를 운영하는 컴퓨터 내의 버퍼 오버플로우 취약성을 이용하여, 2003년 1월 25일 05:30 UTC 조금 전에, 슬래머는 호스트를 감염시키기 시작하였다. 때로는 Sapphire라고도 불리는, 슬래머의 가장 두드러진 특징은 전파 속도(propagation speed)이다. 4K 바이트 코드 레드보다 훨씬 작아 하나의 UDP 패킷의 376 바이트 페이로드 안에 맞다. 포트 1434로 향하는 한 개의 UDP 패킷이 서비스에서의 버퍼 오버플로우를 일으키기 충분하며 웜의 복사를 설치한다. 인터넷을 통한 확산이 시작된 후 대략 3분 안에, 웜은 완전한 스캐닝 속도를 얻었으며, 10 분 내에 취약한 호스트의 90 % 이상인, 대략 75,000 서버를 감염시켰다. 처음 1분 안에 감염은 매 8.5초마다 두 배가 되었으며, 단지 3분 후에 초당 5천 5백만 스캔의 피크 스캐닝 율을 기록한다. 대조적으로, 코드 레드 감염은 37분에 두 배가 되었으며, 그 대신 더 많은 기계를 감염시켰다.

코드 레드와 슬래머는 취약한 호스트를 찾기 위하여 동일한 기초적인 스캐닝 기술을 사용하지만, 스캐닝 제한 사항(constraints)에서는 다르다. 코드 레드는 TCP를 사용하기 때문에 지연-제한(latency-limited)적이며, UDP를 사용한 슬래머는 대역폭-제한(bandwidth-limited)적이라고 할 수 있다.

그림 2는 슬래머가 나온 후 30분 안에 확산된 지역을 보여준다.

그림 2. 30분 후 슬래머의 지역적인 확산
(출처: www.caida.org)

3. 웜의 분류

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[13]: 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 5 가지로 분류한다[17]:

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, 다음 절에서 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[16]. 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 전략과 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.

4. 전파 모델

4.1 개요

질병의 역학(epidemiology) 연구에서, 바이러스 확산에 대하여 결정적이고 확률적인 모델들이 많이 있으나, 인터넷 웜 전파 모델링을 위한 모델은 별로 없는 편이다. 1990년대 초에 IBM에서 역학 모델에 기초한 바이러스성 감염에 대한 일련의 연구를 수행하였다[6-8].

전통적인 역학적 모델은 감염된 호스트가 어떤 다른 취약한 호스트에게라도 똑같이 감염시킬 수 있다는 의미에서 모두 동질성(homogeneous)이다[13]. 지역적인 바이러스의 상호작용을 고려하여, 그러한 동질성 역학적 모델들이 랜덤 그래프, 2-차원 격자(lattice) 및 트리 같은 계층적 그래프와 같은 비동질성 네트워크에 대한 역학적 모델로 확장되었다[4]. 지역적인 상호 작용에 대한 가정(assumption)은 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 사고 이후 인터넷 웜 전파를 모델하고 분석하기 위한 계기가 되었다. Staniford 등은 사고 발생 후 바로 코드 레드 웜 확산을 모델하기 위하여 고전적인 단순 역학적 방정식을 사용하였다[12]. 그리고 코드 레드 웜 전파의 시각적 시뮬레이션[5], 코드 레드 웜 행위의 관측 데이터와 상세 분석[9], 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[15]:

- 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.

감염 모델링 영역에서, 바이러스 감염 율은 일정하다고 가정하였다. 이전의 인터넷 바이러스 및 웜 모델들의 대부분은 감염된 호스트가 타겟을 발견하는데 필요한 시간이, 이미 감염되었거나 혹은 여전히 취약한가에 관계없이, 여전히 상수로 취급한다. 상수 감염율은 역학 모델링에 적합하지만 인터넷 바이러스 및 웜에게는 유효하지 않다[15].

2001년 7월의 코드 레드 사고 분석을 통하여, Zou 등은 코드 레드 전파에 영향을 미치는 두 가지의 요인(factor)이 있는 것을 발견하였다. 그 요인은 다음과 같다.

- ISP와 사용자에 의하여 취해지는 동적인 대응책
- 코드 레드의 과도한 전파로 어떤 라우터들에게 혼잡(congestion)과 고장야기로 인한 웜 감염률의 저하

위와 같이 인적 대응책의 동적인 측면과 가변 감염률을 고려하여, Two-factor 모델을 유도하였다. 이 모델에 대해서는 4.4절에서 기술한다.

4.2 단순 역학 모델

고전적인 단순 역학 모델(epidemic model)에서, 각 호스트는 두 상태 중 하나에 있다: 취약적(susceptible) 혹은 감염성(infectious). 일단 호

스트가 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 (1)과 같다.

$$\frac{dJ(t)}{dt} = \beta J(t)[N - J(t)] \quad (1)$$

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.

이 모델은 균일 스캔 웜의 메커니즘을 캡쳐할 수 있다[7]. 특별히 인간적 대응책과 혼잡의 영향이 무시될 수 있는 웜 전파의 초기 부분에 유효하다. 그림 1에서 보면, 웜의 전파를 세 가지의 단계로 대략 구분할 수 있다[18]: *늦은 시작 단계(slow start phase), 빠른 확산 단계(fast spread phase), 늦은 종료 단계(slow finish phase).*

*늦은 시작 단계*에서, $J(t) \ll N$이므로, 감염 호스트의 수는 지수적으로 증가한다. $a(t) = J(t)/N$을 시간 t에서 감염성인 모집단의 부분이라고 하자. 식 (1)의 양변을 N^2으로 나누면, 다음 식 (2)가 유도된다.

$$\frac{da(t)}{dt} = ka(t)[1 - a(t)] \quad (2)$$

여기서, $k = \beta N$. 식 (2)는 $1-a(t)$가 대략 1과 같은 초기에는 감염 호스트의 수가 거의 지수적으로 증가됨을 보여준다.

많은 호스트들이 감염되고 다른 호스트를 감염시키는데 참여하고 난 후, 웜은 취약 호스트들이 빠르게 거의 선형적인 속도로 감염되는 *빠른 확산 단계*에 들어간다. 전파 율은 모든 취약적 호스트의 약 80%가 감염될 때 감소되기 시작하며, 대부분의 취약 호스트들이 감염된 때에, 웜은 *늦은 종료 단계*에 들어간다.

4.3 Kermack-Mckendrick(KM) 역학 모델

역학 분야에서, KM 모델은 감염성 호스트의 제거 과정을 고려하고 있다[4]. 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 (3)이 성립된다.

$$J(t) = I(t) + R(t) \quad (3)$$

단순 역학 모델 (1)에 기반, KM 모델은 식 (4)와 같이 나타낼 수 있다.

$$\begin{cases} dJ(t)/dt = \beta J(t)\,[N - J(t)] \\ dR(t)/dt = \gamma I(t) \\ J(t) = I(t) + R(t) = N - S(t) \end{cases} \quad (4)$$

여기서 β는 감염률이고, γ는 유통에서 제거된

감염 호스트의 제거율, $S(t)$는 시간 t에서 취약 호스트의 수, N은 모집단의 크기이다.

$\rho = \gamma/\beta$를 상대적 제거율이라 정의하자. 결과적으로 이 모델로부터 식 (5)와 같은 결과를 얻을 수 있다.

$$\text{만일 } S(t) > \rho \text{이면 , } \frac{dI(t)}{dt} > 0 \qquad (5)$$

생성될 새로운 취약 호스트가 없기 때문에, 취약 호스트의 수 $S(t)$는 시간 t의 단순 감소함수(monotonically decreasing function)이다. 만약 $S(t_0) < \rho$이면, 모든 미래의 시간 $t > t_0$에 대하여 $S(t) < \rho$이고 $dI(t)/dt < 0$가 된다. 즉, 만약 취약 호스트의 초기 수가 어떤 임계 값, $S(0) < \rho$보다 작다면, 전염과 창궐이 없게 된다[4].

KM 모델은 어떤 감염 호스트가 약간의 시간 후에 회복되든지 혹은 죽게 된다는 것을 고려함으로써 고전적인 단순 역학적 모델을 개선시켰다. 그러나 이 모델도 인터넷 웜 전파를 모델링하는데 다음과 같은 문제로 적합하지 않다[15]:

- 웜에 대한 청소(cleaning), 패칭, 필터링 대응책이 취약 호스트 및 감염 호스트의 유통을 없앤다. 그러나 KM 모델은 단지 감염 호스트의 제거만 고려한다.
- 감염률이 일정하다고 가정하는데, 코드 레드와 같은 과도하게 확산하는 웜에 대하여는 맞지 않다.

4.4 Two Factor 모델

4.1절에서 기술한 바와 같이, 코드 레드 사고 이후, Zou 등은 코드 레드 웜 전파에 영향을 미친, 전통적인 역학 모델에서는 고려되지 않았든 두 요인을 발견하였다: 인간 대응책과 감소되는 감염률.

감소된 웜 스캔 율을 고려하기 위하여, 식 (1)의 감염률 β는 시간의 함수인 $\beta(t)$로 모델되어야 한다. 웜의 관점에서, 인간의 대응책은 어떤 호스트들을 웜 확산 유통으로부터 제거한다. 제거되는 호스트들은 감염 호스트 및 여전히 취약한 호스트들을 다 포함한다. $R(t)$를 감염 집단으로부터 제거된 호스트의 수라 정의하고, $Q(t)$를 취약 집단으로부터 제거된 호스트의 수라 정의한다. KM 모델(5)을 유도하는데 사용한 같은 원칙에 의하면, 시간 t에서 시간 $t+\triangle t$까지의 취약 호스트 $S(t)$의 수 변화는 다음과 같이 된다:

$$S(t+\triangle t) - S(t) = -\beta(t)S(t)I(t)\triangle t - \frac{dQ(t)}{dt}\triangle t \qquad (7)$$

그러므로

$$\frac{dS(t)}{dt} = -\beta(t)S(t)I(t) - \frac{dQ(t)}{dt} \qquad (8)$$

어떤 시간 t에 대해서도 $S(t)+I(t)+R(t)+Q(t)=N$이 성립되므로, $S(t)$의 값을 식 (8)에 대체하면 감염 호스트 $I(t)$ 수의 행위를 기술하는 (9)와 같은 미분 방정식이 만들어진다.

$$\frac{dI(t)}{dt} = \beta(t)[N - R(t) - I(t) - Q(t)]I(t) - \frac{dR(t)}{dt} \qquad (9)$$

식 (9)에 의하여 기술되는 웜 모델을 Two-factor 모델이라 한다.

엄격히 말하자면, 웜 전파는 이산 사건 프로세스이지만, [15]에서는 웜 전파를 연속 프로세스로 취급하고 (9)와 같은 연속 미분 방정식을 사용한다. 그림 3은 어떤 파라미터들에 대한 Two-factor 모델의 수치 해를 보여준다.

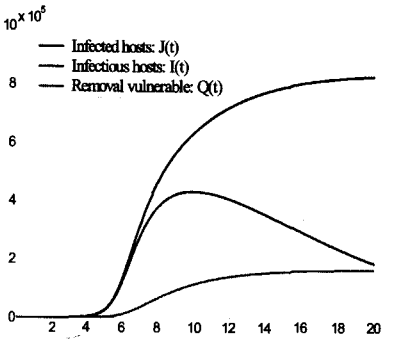

그림 3. Two-factor 웜 모델의 수치해

그림 3에서, t=10에서 감염성(infectious) 호스트의 수 $I(t)$가 최대가 되고, 그 이후로 제거된 취약 호스트의 수 $Q(t)$가 증가됨에 따라 감소한다. 그림 3의 감염성 호스트의 수 $I(t)$의 행위가 코드 레드 스캔 시도가 발생일 마지막 몇 시간동안 저하된 이유를 설명한다.

4.5 AAWP 모델

AAWP(Analytical Active Worm Propagation) 모델은 랜덤 스캐닝을 채택하는 웜의 전파를 특성화한다[1]. 이 모델은 이산 사건 및 연속 상태 결정적 근사화 모델을 사용한다. 본 논문에서는 역학 모델과 비교하여 기술하며, 보다 자세한 내용은 [1]을 참조할 수 있다.

AAWP 모델과 역학 모델과의 차이는 아래와 같다[1]:

- 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[7]. 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.

4.6 고속망에서의 전파

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 (2)식에서 $\beta(1-a(t))$를 증가시킬 수 있다. 단순 역학 모델은 다음과 같이 재배열 될 수 있다:

$$T_p = \frac{\ln P(N-J(0)) - \ln(1-P)J(0)}{\beta N} \qquad (10)$$

여기서 T_p는 모집단의 P 부분, 즉 PN 호스트를 감염시키기 위하여 걸리는 시간을 나타낸다. 이 결과는 만약 웜이 조사 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[1, 14]. 따라서 고속망에서 웜은 더 높은 감염률을 얻을 수 있고 목표 집단을 더욱 빠르게 포화시킬 수 있다.

5. 웜의 탐지

새로운 웜을 자동으로 탐지하고 봉쇄하기 위한 필요성이 오래전부터 인식되어져 왔다. 멀웨어(malware)에 대한 전통적인 방어는 항바이러

스 소프트웨어, 운영 체제 패치 및 방화벽과 같은 네트워크 보안 장비의 특별한 결합으로 구성된다. 본 장에서는 웜을 포함한 악성 코드의 탐지 메커니즘에 대하여 간단히 기술한다.

5.1 악성 코드의 형태

악성 코드의 주요 형태는 다음과 같다[3]:

- 바이러스: 파일을 감염시키는 자기-복제 프로그램으로 확산하기 위하여 보통 인간의 중재가 필요하다.
- 웜: 독립적으로 네트워크를 통하여 확산되는 자기-복제 프로그램
- 악성 모바일 코드: 원거리 호스트로부터 다운로드된 프로그램으로 보통 웹서버와 상호작용하기 위하여 설계된 언어로 작성된다.
- 백도어: 보안 메커니즘을 회피하는 프로그램
- 트로이 목마: 유용한 것처럼 보이지만, 대신에 어떤 악성 기능을 수행하는 프로그램
- 사용자 레벨 루트킷: 시스템 관리자 및 사용자에 의하여 수행되는 프로그램을 대치하거나 변경하는 프로그램
- 커널 레벨 루트킷: 발생했다는 것을 나타내지 않고 운영 체제를 수정하는 프로그램
- 결합 멀웨어: 범주 경계에 걸쳐있는 악성 코드

5.2 악성 코드 탐지 방법

악성 코드를 탐지하기 위하여 침입탐지시스템(IDS: Intrusion Detection System)과 침입방지시스템(IPS: Intrusion Prevention System)은 일반적인 공격을 탐지하는 것과 거의 같은 방법을 사용한다. 아래에 악성 코드를 탐지하는 방법을 기술한다:

- 네트워크 상으로 전송된 악성코드는 항바이러스 소프트웨어에 의하여 인식되는 것과 같이 시그너처(signature)에 의하여 특성화된다. IDS와 IPS는 네트워크 데이터와 시그너처를 대조하여, 트래픽이 암호화되지 않는 한 실행문 안의 악성 코드 스트링을 구별한다.
- 포트 활성화를 기반으로 툴이 적용될 수 있다.
- 감염시키기 위한 스캔 공격 탐지
- 시스템 파일과 디렉터리 변화를 탐지하기 위한 툴
- 시스템 내의 징후를 이용하는 방법

웜의 탐지 및 방지 방법을 좀 더 포괄적으로 기술하면 다음과 같다[11]:

- 이상적: 이상적인 방법으로 웜 발생을 조기에 탐지하는 방법이다. 자동으로 시그너처를 생성하여 패킷을 즉시 필터링한다. 웜이 확산되는 것보다 경보와 시그너처를 더 빨리 분배한다. 그러나 이것은 현실적으로 매우 어려운 방법이다.
- IPv6 대 웜: IPv6는 2^{128}개의 IP 주소를 가지고 있다. 가장 작은 서브넷도 2^{64}개의 주소를 가지게 된다. 예를 들어, 백만 개의 취약 호스트, 초당 십만 개의 스캔, 천 개의 초기 감염 호스트를 가정하여도 랜덤 스캐닝으로 취약 호스트의 50%를 감염시키기 위하여 40년이 필요하다. 따라서 스캔-기반 웜은 효과적이지 못할 것이다.
- Earlybird: "플로우"가 패킷 내용이나 내용의 해시(hash)에 의하여 식별된다. 인기 있는 플로우에 대한 분명한 근원지 및 목적지의 카운터가 유지된다. 카운터가 경계(threshold)를 초과하면, 플로우는 웜으로 간주되고 내용이 시그너처를 위하여 사용된다.

- Honeycomb: 하니팟으로의 모든 트래픽은 의심스러운 것으로 가정한다. 모든 입력 패킷에 대하여, 헤더 분석을 수행한 후 시그너처를 발견하기 위하여 가장 긴 공통 서브스트링(LCS: longest common substring) 알고리즘을 사용한다. 시그너처 풀에 시그너처를 추가한다.
- BGP 정보: 스캔을 조사하기 위하여 블랙 주소 공간을 사용한다. 이것은 랜덤 스캐닝 웜의 탐지에만 유용하다. 각 AS(Autonomous System)로부터 얼마나 많은 트래픽이 오는지 모델을 구축하기 위하여 AS 프로파일링을 사용하고 급격한 변화를 조사한다.
- Kalman 필터: 다음을 측정하기 위하여 네트워크 주위에 (입력 및 출력 필터) 센서를 분산 시킨다: 스캔 율, 스캔 분배, 스캔의 총수, 감염 호스트의 총수. 정보가 중앙 멀웨어 경고 센터에 전송된다. 멀웨어 센터는 트래픽 증가 경향을 계산하기 위하여 칼만 필터를 사용한다. 만약 증가가 지수적 모델에 부합되면, 웜으로 간주된다. 센서는 블랙 IP 주소로 전송된 패킷에 의하여 정보를 측정한다. 센서는 정확한 정보를 얻기 위하여 2^{20}개의 IP 주소를 감시해야 한다. 히트-리스트나 위상적 웜에 의하여 회피될 수 있다.
- Hidden 마코브 모델: 웜 탐지에는 그다지 유용하지 않다.
- E-mail 웜 탐지: 사용자가 특별한 집합의 사용자들에게 통상적으로 메일을 보낸다는 사실을 이용하는데, 방송 메일의 경우 사용될 수 없다.

5.3 문제점

IDS와 IPS에서 가장 중요한 문제는 탐지의 정확성이다. IDS에서 공격 분석의 두 가지 기본적인 방법은 오용 탐지와 비정상 탐지이다. 통상적인 IDS에서는 일련의 공격 시그너처를 정의하여 놓고 그것과 비교되는 행위를 조사한다. 이 시그너처 기반 방법은 알려진 시그너처에 따르지 않는 새로운 웜을 탐지할 수 없다는 것이다. 이것을 소위 "zero-day" 공격이라 하며, 패치 혹은 시그너처가 갱신되기 전에 발생한 취약성을 이용하여 공격하는 것을 의미한다. 대조적으로 비정상 탐지는 정상 행위에 대한 통계 패턴을 정의하고 그 패턴에서의 어떤 편차를 의심스러운 행위로 규정한다. 이 방법은 알려진 시그너처가 없는 새로운 웜을 탐지할 수 있지만, 정상 행위를 얼마나 정확하게 정의하느냐가 관건이다. 예를 들어, 웜의 대표적인 표시로 네트워크 트래픽 양이 급격히 증가하는데, 포트 스캔과 같은 정상적인 네트워크 트래픽 패턴에서도 이런 현상이 발생할 수 있다는 것이 문제이다.

6. 웜 트래픽 모델링

6.1 개요

웜을 연구하고, 조기 경보 시스템을 구축하고, IDS 및 다른 보안 제품의 시험을 위하여 한 가지 주요한 요구사항이 있다면, 웜 트래픽에 대한 모델링이라 할 수 있다. 웜 트래픽의 모델은 다음과 같이 4 가지로 분류할 수 있다.

- **수학적 모델**

가장 강력한 방법은 아마도 닫혀진 형태(closed form)로 웜의 행위 예측을 할 수 있는 현실적인 수학 모델을 생성하는 것이다. 이 접근의 문제점은 이러한 모델이 일반적으로 이용가능하지 않을 뿐만 아니라, 생성하는 것이 통상적으로 매우 어렵고 불가능하기 조차하다. 웜의 생명주기 동안 잠복 단계와 같은 비연속성을 쉽게 반영할 수 없고, 또한 랜덤 요소를 도입하고 적응적 행

위를 모델하기가 힘 든다는 점이다.

• **테스트 베드**

고립되고 제한적인 환경에서 어떤 자기 복제 코드를 실제로 설치하여 행위를 관측하는 것이다. 테스트 베드의 가장 분명한 제한은 인터넷의 크기에 가까운 크기로 만들 수 없다는 것이며, 인터넷의 특성으로 인하여 작은 모델만으로는 사용될 수 없고 결과가 더 큰 모델로 확장된다는 것이다. 또 다른 문제로 테스트 베드는 구하기 힘든 실제 자기-전파 코드를 사용할 필요가 있다는 것이다.

• **실제 환경**

웜 코드 제작자들은 이 방법을 채택할 수 있지만, 발생되는 손해 때문에 과학 연구용을 위하여 사용될 수 없다.

• **시뮬레이션**

어떤 의미에선 어떤 함수들이 반복(iteration)에 많이 의존하는 수학모델이라 할 수 있다. 수학 모델링의 해석적 방법이 여러 시나리오들이 모의실험 되고 분석되는 실험적 방법으로 대체된다.

6.2 시뮬레이션 모델링

실제적이고 재생할 수 있는 방법으로 실험실 환경에서 대규모 웜 공격의 효과를 만들 수 있는 것은 웜의 탐지와 방어 시스템의 개발을 위하여 중요한 문제이다. 이를 위하여 시뮬레이션은 어떤 다른 방법보다도 유용하며, 시뮬레이션의 장점은 아래와 같다.

- 시뮬레이션에서 필요한데로 어떤 파라미터를 모델하기가 쉽다.
- 어떤 크기의 네트워크도 대표적으로 모의실험이 가능하다.
- 어떤 특별한 현상을 필요한 경우 어느 때라도 발생시켜 준비할 수 있다.

시뮬레이션은 동시에 단점도 가지고 있으며, 사용자는 옳지 않은 결과에 이르지 않도록 요인에 유의하여야 한다. 어떤 시뮬레이션이라도 자신이 모델하는 실제 시스템의 요소들을 무시할 수 있는데, 아래와 같은 것이 있을 수 있다.

- 시간이 연속이 아닌 이산 단계로 지나간다.
- 모의실험 네트워크는 라우터, 방화벽 혹은 브릿지와 같은 성능 병목현상을 포함하지 않는다.
- 모든 호스트는 같은 성능을 나타낸다.

아래에 웜 시뮬레이션을 위하여 이용할 수 있는 시뮬레이터에 대하여 간단히 소개한다.

- 네트워크 시뮬레이터(ns)-2: 네트워크 성능 분석에 일반적으로 많이 사용되는 이산 사건 시뮬레이터이다. 유선 및 무선 네트워크 상으로 TCP, 라우팅 및 멀티캐스트 프로토콜을 위한 시뮬레이션을 상당부분 지원한다.
- 네트워크 웜 시뮬레이터(NWS): Perl로 쓴 네트워크 웜 시뮬레이션을 작성하기 위한 완전한 프레임워크이다[19]. NWS는 웜이 감염하는 모든 엔티티에서 "웜 코드"를 실제로 실행한다. 실제로 코드를 실행함으로써 웜이 실제와 꼭 같은 임의의 행동을 수행하도록 허용할 수 있다.
- SSF(Scalable Simulation Framework) 웜 시뮬레이터: SSF는 자바와 C++로 작성된 대규모, 복잡한 시스템의 이산-사건 시뮬레이션을 위한 공개-도메인 표준이다. 네트워크 웜의 확산을 모델하기 위하여 SSF.App.Worm 패키지를 작성하였다[20]. 이것은 패킷 레벨 시뮬레이터와 통합된 모델이다. 이것은 'two-tier' 모델을 이루고 있으

며, 대규모 행위는 거시적 레벨로 모델하고 선택된 부분들은 미시적 레벨에서 모델할 수 있도록 하였다.

네트워크에서의 웜의 확산은 거시적 확산 모델을 사용한다. 현재 지원되는 특징은 다음과 같다:

- 결정적 대 확률적 감염 모델. 모델은 코드 레드 v2 혹은 사파이어/슬래머 웜과 같이 균일 랜덤 스캐닝에 의한 웜 확산을 가정한다.
  - Kermack-McKendrick에 의한 결과로부터 유도된 잘 알려진 미분 방정식을 기초로 한 결정적 모델
  - 유사한 가정을 기초로 한 확률적 모델
- 동질성 혹은 계층화된 집단 모델
  - 동질성: 모든 취약 호스트는 동질성 집단의 인터넷에 있다.
  - 계층화: 취약 호스트의 집단이 AS에 의하여 계층화된다. 각 AS는 하부 집단을 나타낸다.
- 계층화된 감염 모델을 위한 초기화 선택
  - leaf AS 상의 균일 취약 분포 및 AS들 사이의 균일 감염률
  - 코드 레드 웜의 경험 데이터를 기초로 한 취약 분포
- 단순 웜 스캔 트래픽 모델: 단순한 평균 스캔률 플로우 모델을 사용한다.
- DML(Domain Modeling Language) 구성가능
  - 총 취약 집단, 초기 감염 집단, 감염 파라미터와 같은 파라미터
  - 감염 모델, 감염을 위한 초기화 코드, 감염을 위한 제거 기능과 같은 구현 선택
- 네트워크의 예제 모델
- 회귀(regression) 시험

7. 맺음말

본 논문에서는 인터넷 웜 전파에 대한 분석을 하고 웜 모델링에 대하여 기술하였다. 이를 위하여 먼저 웜의 스캐닝 전략에 따른 웜을 분류하고, 웜의 전파 특성을 분석하였다. 웜 모델로 단순 역학 모델, Kermack-Mckendrick 모델, Two-factor 모델, AAWP 모델, 고속망에서의 전파 특성에 대하여 기술하였다. 이 모델들이 코드 레드 웜 전파를 모델하기 위하여 어떻게 사용되는지 살펴보았다. 또한 웜을 비롯한 악성 코드의 탐지 메커니즘과 웜 트래픽 모델링 방법 중 시뮬레이션 모델링에 대하여 살펴보았다.

향후 연구로는 인터넷 웜을 효과적으로 모니터링을 하고 조기에 탐지하여 대응하는 방안에 대한 연구가 있다[18].

참고문헌

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

[2] Ram Dantu, Joao Cangussu, Arun Yelimeli, "Dynamic Control of Worm Propagation", International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 1, April 05 - 07, 2004, Las Vegas, Nevada.

[3] Carl Endorf, Eugune Schultz, Jim Mellander, Intrusion Detection and Prevention, McGraw-Hill, 2004.

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

1980.

[5] T. Heberlin, Visual simulation of Code Red worm propagation patterns. http://www.incidents.org/archives/intrusions/msg00659.html

[6] 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.

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

[8] J. O. Kephart and S. R. White, "Measuring and Modeling Computer Virus Prevalence", Proc. of the IEEE Symposium on Security and Privacy, 1993.

[9] D. Moore, The Spread of the Code-Red Worm, http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml

[10] David Moore et al., "Inside the Slammer Worm", IEEE Security & Privacy, pp33-39, 2003.

[11] Mark Shaneck, Worms: Taxonomy and Detection, ppt document, Feb. 6, 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] Arno Wagner, Thomas Dubendorfer, "Experiences with Worm Propagation Simulations", WORM'03, pp.34-41, Oct. 2003, Washington DC. USA.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Simulating Network Worms, NWS by Bruce Ediger, http://www.users. qwest.net/~eballen1/nws/

[20] SSF.App.Worm, A Network Worm Modeling Package for SSFNet. http://www.cs.dartmouth.edu/~mili/research/ssf/worm/index.html

전 용 희

1971년 3월 ~ 1978년 2월 : 고려대학교 전기공학과

1985년 8월 ~ 1987년 8월 : 미국 플로리다공대 대학원 컴퓨터공학과

1987년 8월 ~ 1992년 12월 : 미국 노스캐롤라이나주립대 대학원 Elec. and Comp. Eng. 석사, 박사

1978년 1월 ~ 1978년 11월 : 삼성중공업(주)

1978년 11월 ~ 1985년 7월 : 한국전력기술(주)

1989년 1월 ~ 1989년 6월 : 미국 노스캐롤라이나주립대 Dept of Elec. and Comp. Eng. TA

1989년 7월 ~ 1992년 9월 : 미국 노스캐롤라이나주립대 부설 CCSP(Center For Comm. & Signal Processing) RA

1992년 10월 ~ 1994년 2월 : 한국전자통신연구원 광대역통신망연구부 선임연구원

1994년 3월 ~ 현재 : 대구가톨릭대학교 컴퓨터·정보통신공학부 교수

2000년 1월 ~ 현재 : 한국통신학회 학회지 편집위원

2001년 3월 ~ 2003년 2월 : 대구가톨릭대학교 공과대학장 역임

2004년 2월 ~ 현재 : 한국전자통신연구원 정보보호연구단 초빙연구원

<관심분야> 네트워크 보안, 통신망 자원관리 및 성능분석, QoS 보장 기술