情 報 保 護 學 會 誌
第 16 卷 第 1 號, 2006. 2

웜 공격의 특성 분석 및 확산 방지 기술

전 용 희*

요 약

가장 통상적인 사이버 공격 형태는 컴퓨터 네트워크에 대한 공격으로 악성 코드가 이용된다. 이 중에서도 웜에 의한 공격이 현재의 고도로 네트워크화 된 환경에서 심각한 위협이 되고 있다. 인터넷 웜의 효시인 모리스가 발표된 이후로 웜에 대한 많은 연구가 이루어지고 있으나, 웜의 공격을 사전에 방지하기 위한 확실한 대책이 없는 실정이다. 최근에 웜의 대응 기술 중의 하나로 웜의 빠른 확산을 방지하기 위하여 트래픽 율(rate)을 제한하는 방법에 많은 관심이 집중되고 있다. 따라서 본 논문에서는 웜 공격에 따른 특성을 분석하고, 웜의 확산을 막기 위하여 사용되는 Rate-limiting을 포함하여, 봉쇄(containment) 기술에 대하여 제시하고자 한다.

Ⅰ. 서 론

소프트웨어나 시스템 구성상의 약점을 이용하여 발생하는 사이버 공격(cyber attack)은 보통 악성 코드(malicious code)를 통하여 발생하며, 데이터의 무결성(integrity)이나 신빙성(authenticity)을 파괴한다. 악성 코드의 형태로는 바이러스, 웜, 악성 모바일 코드, 백도어(backdoor), 트로이 목마, 사용자-수준 및 커널-수준의 루트 킷(root kit), 그리고 혼합 멀웨어(malware) 등이 있으며 다음과 같이 정의된다.[13,17]

- 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 루트 킷: 시스템 관리자 및 사용자에 의하여 수행되는 프로그램을 대치하거나 변경하는 프로그램
- 커널 레벨 루트 킷: 발생했다는 것을 나타내지 않고 운영 체제를 수정하는 프로그램
- 혼합 멀웨어: 범주 경계에 걸쳐있는 악성 코드

본 논문에서는 이 중에서 웜에 대하여 초점을 맞추고자 한다.

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.

코드 레드(Code Red) 웜은 윈도우 IIS(Internet Information Server) 인덱스 서비스 DLL의 버퍼 오버플로 버그를 이용한다. 이 웜은 TCP 포트 80번을 이용하여 취약 호스트를 스캔한다. 잠재적인 타겟

* 대구가톨릭대학교 공과대학 컴퓨터정보통신공학부(yhjeon@cu.ac.kr)

과 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 병열성을 통하여 감염률을 빠르게 한다. 연결이 설정되지 않거나 목표가 웹 서버가 아닌 경우, 웜 쓰레드는 탐사(probe)하기 위한 다른 IP 주소를 임의로 생성하는 랜덤 스캐닝(random scanning)을 사용한다.

코드 레드와 마찬가지로, SQL 슬래머(Slammer) 웜도 마이크로소프트사의 SQL 서버를 운영하는 컴퓨터 내의 버퍼 오버플로 취약성을 이용하여, 2003년 1월 25일 호스트를 감염시키기 시작하였다. 때로는 Sapphire라고도 불리는, 슬래머의 가장 두드러진 특징은 전파 속도(propagation speed)이다. 4K 바이트 코드 레드보다 훨씬 작아서, 하나의 UDP 패킷의 376 바이트 페이로드 안에 맞다. 포트 1434로 향하는 한 개의 UDP 패킷이 서비스에서의 버퍼 오버플로를 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.[3]

위와 같은 웜을 효과적으로 탐지하고 사이버 공격에 대한 방어를 위하여 웜 탐지 및 방어 시스템은 다음과 같은 능력을 가져야 한다.[6]

- 실제 인터넷 환경에서 보안 기능 수행의 견고성 및 탄력성
- 트러스트(trust) 통합 및 경보-상호연관 방법론에 의한 다수 사이트 사이의 협동 실행 가능성
- 기대치 않은 웜 혹은 플러딩 공격에 대한 대응 (신속한 비정상 탐지 및 분산 서비스 거부 (DDoS: distributed denial-of-service) 공격 방어에 의하여 가능해야 함)
- DDoS 공격-경유(transit) 라우터 추적을 위한 정확한 트래픽 모니터링과 신속한 웜 시그너처 탐지 및 분류에 의하여 실행 가능한 효율성 및 확장성

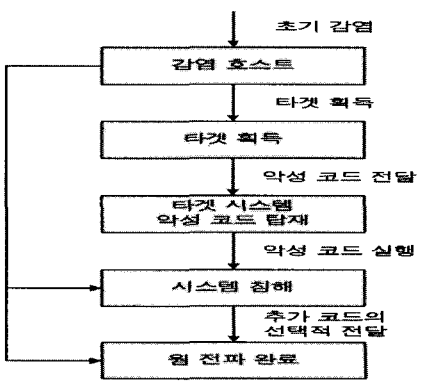
II. 웜의 공격 특성

2.1 웜의 정의 및 전파단계

바이러스와는 다르게, 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. 웜은 자기 스스로 전파하며, 여기서 자기 스스로 전파한다는 의미는 사용자의 간섭 없이 컴퓨터 시스템 사이에 확산될 수 있는 것으로 정의될 수 있다. 따라서 웜에 대한 정의를 다음과 같이 내릴 수 있다. "웜은 컴퓨터 네트워크를 통하여 확산되는 자기 스스로 전파되는 바이러스이다."

주요한 웜의 전파 방법으로 다음과 같은 네 가지 방법이 있다.

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



〔그림 1〕 웜의 대표적인 전파 단계

스캐닝을 통하여 전파되는 웜은, 스캐닝 단계에서 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 1은 웜의 대표적인 전파 단계를 보여준다.

단계별로 발생하는 대표적인 활동은 아래와 같다.[3]

1) 초기 감염 단계: 웜에 의하여 이미 감염된 시스템이 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 셀, 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.

웜이 보여주는 주요한 행위 특성은 아래와 같다.[16] 이러한 행위를 기준으로 웜을 탐지하고 특성화하는데 사용한다.

- 콘텐트 불변(content invariance): 알려진 모든 기존 웜과 거의 모든 웜 프로그램은 모든 복제에서 동일하다. 대표적으로 전체 웜 프로그램은 감염시키는 모든 호스트에 걸쳐서 동일하다. 그러나 일부 웜들은 제한된 다형태성(polymorphism)을 사용한다. 이 경우에도 웜 몸체(body)의 대부분은 변하지만, 중요 부분은 여전히 불변으로 남아있게 된다.
- 콘텐트 유포(content prevalence): 웜은 확산하기 위하여 주로 설계되기 때문에, 웜 콘텐트의 변하지 않는 부분이 웜이 확산되거나 확산을 시도할 때 네트워크상에 빈번히 나타나게 된다. 이런 웜의 변하지 않는 부분이 시그너처를 생성하는데 유용하다.
- 주소 확산(address dispersion): 웜은 다양한 소스와 목적지 주소로 확산된다.

2.2 분류 기준

웜은 그들의 통상적인 특성에 따라서 분류할 수 있다. 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.[3]

- 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.

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

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

2.3 공격 속성

웜에 의하여 취해지는 특정한 관측 가능한 행동 특성을 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 레지스터리(registry)에 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.

[5]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.

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

Ⅲ. 웜의 전파 특성

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.[18] 그리고 코드 레드 웜 전파의 시각적 시뮬레이션, 코드 레드 웜 행위의 관측 데이터와 상세 분석, 웜 설계 원칙 등에 대한 연구가 수행되었다.

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.[8,24]

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.

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

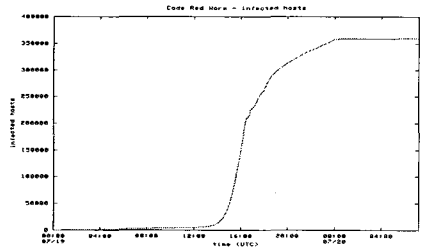
이것은 웜의 임의 고정 확산률(spread rate)을 나타낸다. 이 미분방정식을 풀면 (3)과 같다.

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

여기서 a는 침해된 취약 호스트의 비율, t는 시간, K는 초기 침해 율, N은 전체 취약 집단의 크기, 그리고 T는 성장이 시작된 고정 시간을 의미한다. 율 K는 이미 감염된 호스트를 고려하여 조정되어야 하며, $e^{K(t-T)}$이 된다.

이 방정식을 병참(logistic) 성장 모델이라 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.[13]

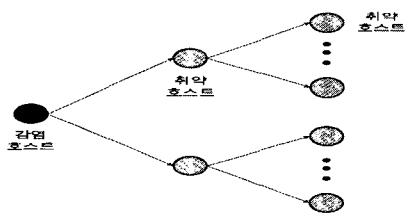


〔그림 2〕 웜의 대표적인 전파 곡선

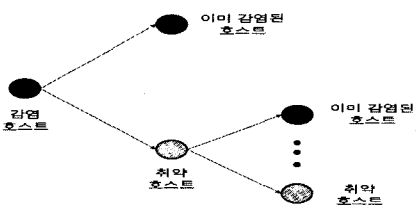
그림 2는 시간에 따른 감염 호스트 수의 변화를 보여준다. 그림 2에서 보면 웜의 전파를 세 가지 단계로 대략 구분할 수 있다[25]: 늦은 시작 단계, 빠른 확산 단계, 늦은 종료 단계. 늦은 시작 단계에서는 감염 호스트의 수는 지수적으로 증가한다.

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



(a) 확산 단계에서의 웜의 전파



(b) 종료 단계에서의 웜의 전파

〔그림 3〕 웜의 단계별 전파 특성

Ⅳ. 웜의 탐지 기술

4.1 시그너처 기반 탐지

웜이 유포되고 난 후 이들에 대한 정보를 수집하여 이 정보를 기반으로 웜에 대한 시그너처(signature)를 생성한 후 이것을 가지고 탐지하는 방식이며, 대부분의 침입탐지 및 방지 시스템에서 적용하고 있다. 이 방식에서는 이미 알려진 웜에 대한 탐지를 위하여 기존의 패턴 매칭 방식을 사용한다. 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.[4] 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(Denial of Service) 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 새

로운 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, DEWP(Detecting Early Worm Propagation) 방식, 통계적 침입탐지 방식, 국내의 ETRI에서 개발한 CPD(Change Point Detection) 방식 등이 있다.[3,4,10,12]

4.3 하니팟 탐지

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 하니팟을 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[13]:

- 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.

V. 봉쇄 기술

웜의 확산을 방지하기 위하여 봉쇄(containment) 혹은 격리(quarantine) 기술이 개발되고 있다. 봉쇄는 활성 웜의 확산을 지연시키거나 방지하기 위하여 사용되는 메커니즘을 뜻한다. 현재 사용 중인 봉쇄 기법의 종류로는 세 가지가 있다:

- 호스트 격리(host quarantine)
- 스트링 매칭
- 연결 throttling

호스트 격리는 감염 호스트가 다른 호스트와 통신하는 것을 단순히 막는 행동으로, 라우터나 방화벽 상의 IP-레벨 접근 제어 목록을 통하여 보통 구현된다. 스트링-매칭 봉쇄는 시그너처-기반 네트워크 침입방지 시스템에서 대표적으로 사용하며, 네트워크 트래픽을 알려진 웜의 특정 스트링이나 시그너처와 매치하여 관련 패킷들을 탈락시킬 수 있다. 고속 매칭을 위하여 FPGA-기반 하드웨어 시스템이 개발되고 있다.[4]

마지막으로, 연결 throttling 방법은 웜의 확산을 막는 것이 아니라, 지연시키기 위하여 외부 연결의 rate를 제한하는 기술이다. 이 기술에 대하여는 VI장에서 기술한다. 웜의 봉쇄를 위하여 개발되고 있는 주요 시스템은 다음과 같다.[6]

5.1 Earlybird 시스템

Earlybird 시스템[16]은 콘텐트 유포를 탐지하기 위하여 콘텐트-감별(content-sifting) 접근을 사용한다. 이 과정에서 발생하는 메모리 소비량을 감소시키기 위하여 다단계 필터와 값 표본추출(value sampling) 기술을 사용한다. 주소 확산을 평가하기 위하여 기존 알고리즘 보다 훨씬 적은 메모리를 사용하여 정확하게 평가하는 눈금 비트맵(scaled bitmap)을 사용한다. 콘텐트 유포는 잠재적인 웜 시그너처 식별을 위한 주요 측정기준이고, 주소 확산은 이 집합에서 오탐율(false positive)을 줄이기 위하여 중요한 것이다.

센서가 구성 가능한 주소 공간 지역상의 트래픽을 감별하여 집합기(aggregator)에게 시그너처를 보고한다. 집합기는 센서들로부터의 실시간-갱신을 조정하며, 관련 시그너처들을 합병하고, 네트워크 혹은 호스트-레벨 차단(blocking) 서비스를 활성화한다.

이 시스템의 특징은 단일 웜 센서 안에서 견고하고 확장 가능한 와이어-속도(wire-speed) 구현을 지원하는 것이다. 반면에, 이 시스템은 중앙 집중기를 통한 분배만 허용한다. 내용 유포에서 다른 센서들 사이의 정보 공유를 지원하지 않는 특징이 있다.

5.2 Autograph 기법

오토그라프[11]는 TCP 전송 프로토콜을 사용하여

전파되는 웜으로부터 시그너처를 자동으로 생성하는 시스템이다. 이 시스템은 부분적인 플로 페이로드의 유포를 분석하여 높은 민감성(sensitivity, 즉 true positive)과 낮은 특정성(specificity, 즉 false positive)을 나타내는 시그너처를 생성한다. 콘텐트 유포 분석 수행을 위한 트래픽 양을 감소하기 위하여 포트-스캔-기반 플로 분류기를 사용한다. 오토그라프는 분산 배치를 위하여 보다 나은 지원을 하며, 분산 모니터 사이에 포트-스캔 보고를 공유하기 위하여 애플리케이션-레벨 멀티캐스트를 사용한다.

Earlybird에서와 마찬가지로, 모니터는 웜 페이로드 정보를 공유하지 않는다. 그래서 각 모니터는 독립적으로 페이로드를 축적한다.

5.3 TRW(Threshold Random Walk) 알고리즘

[20]에서는 TRW 온라인 악성-호스트 탐지 알고리즘을 기반으로 하는 스캔 탐지 및 억압(suppression) 알고리즘을 개발하였다. 알고리즘의 단순화로 하드웨어 및 소프트웨어 구현에 적합하도록 하였다. 주소 별 및 개별 연결들의 활동을 추적하기 위하여 캐시를 사용한다.

봉쇄 장치 사이에서 협동(cooperation)을 통하여 봉쇄를 증진시킨다. 통신을 통하여 경계치(threshold)를 감염 수준까지 동적으로 조정할 수 있다. 웜 봉쇄 시스템들은 감염 경계치(epidemic threshold)를 가진다. 만약 취약 머신의 수가 특정한 봉쇄 배치에 비하여 충분히 적으면, 봉쇄가 웜을 거의 완벽하게 정지시킬 것이다. 그러나 더 많은 취약 머신이 존재한다면, 그때 웜은 지수적으로 증가하게 된다. 이 감염 경계치는 아래에 의존한다.[20]

- 봉쇄 대응 장치의 민감성
- 네트워크상의 취약 머신의 밀집도
- 웜이 정확한 네트워크로 확산되는 정도

5.4 기 타

기타 웜의 봉쇄 기술로 동향 탐지(trend detection)라고 하는 웜 모니터링 및 조기 경보 시스템[25], 자동 시그너처 생성과 분류를 가진 고속 웜 탐지를 위한 DHT(distributed hash table)-기반 오버레이 시스템을 이용하는 NetShield 시스템[6], 네트워크 트래픽 분석 대신에 취약 호스트가 감염되는 것을 방지하기 위하여 종단-시스템 접근을 채용한 웜 백신 프로젝트[15]와 Microsoft의 Shield 시스템[19]이 있다. Symantec 등에서도 조기-경보 및 모니터링 시스템과 웜 봉쇄 기술에 대하여 연구를 진행하고 있다.

Ⅵ. Rate-Limiting 기술

이 기술은 자동 대응 기법의 한 종류로, 합법적인 애플리케이션의 지속적인 운용은 허용하면서 웜 트래픽의 외부 확산을 rate limit하는 방법이다. 최근의 분석 연구는 rate limiting이 네트워크에서 적절한 지점에 배치되면 감염 확산을 상당부분 감소할 수 있다는 것을 보여준다.[22, 23]

Rate limiting 기법으로 다음과 같은 것이 있다.

- IP throttling[21]
- 실패-연결-기반 스킴[7]
- 크레디트-기반[14]
- DNS-기반[9]

6.1 IP throttling

이 기법에서 정상적인 애플리케이션은 대표적으로 웹 서버나 파일 서버와 같은 외부 호스트로 제한된 회수만큼의 안정된 접촉 율(contact rate)을 보인다는 가정에서 운용된다. 유일한 IP들로 호스트 레벨 접촉율을 제한하는 것은 임의의 주소에 대한 빠른 연결을 제한할 수 있다. [21]에서는 각 호스트에 대한 주소들의 활동 집합(active set)을 유지하며, 이것은 호스트의 정상 접촉 행위를 모델 한다.

Throttling 기법은 활동 집합 안에 있는 주소들을 위한 외부(outgoing) 연결은 허용하지만, 다른 패킷들은 지연 버퍼에 넣어 지연 시킨다. 만약 지연 버퍼가 차면, 추가적인 패킷들은 단순히 탈락된다. 지연 버퍼에 들어 있는 패킷들은 일정한 율로 처리된다. 같은 율로 활동 집합 안의 가장 적게 사용된 최근 주소는 새로운 연결을 위한 공간을 위하여 삭제된다. 결과적으로, 빈번히 사용되는 주소에 대한 연결은 높은 확률로 허용되고, 반면에 스캐닝 웜에 의하여 개시되는 임의 주소에 대한 연결은 지연되고, 이루어지지 않도록 하는 것이다.

이 기법에서 활동 집합과 지연 버퍼의 크기가 중요하다. 활동 집합의 크기가 크면 높은 접촉율을 허용하고 지연 버퍼의 길이는 이 기법이 얼마나 자유스러운지 혹은 제한적인지를 결정한다. [21]에서는 호스트-기반 기법에 대하여 다섯 개의 주소 활동 집합과 길이 100의 지연 버퍼를 권고하였다. Throttling의 위치

에 따라서 다음과 같은 기법이 있다.

- 종단 호스트 throttling: 종단 호스트 상에서 IP throttling을 수행한다.
- 에지 라우터 throttling: 네트워크 에지 라우터에서 집적(aggregate) 트래픽에 대하여 실행한다.

6.2 실패-연결(FC: Failed Connection) 기반 스킴

이 기법은 스캐닝 웜에 의하여 감염된 호스트가 많은 수의 실패 TCP 요구를 생성한다는 가정을 기반으로 한다. 이 기법은 이런 현상을 감염의 지시로 사용하며 그러한 행위를 가진 호스트를 rate limit 한다. FC 기법은 에지 라우터 기반으로 두 과정으로 구성된다.

처음 단계에서, 잠재적인 "감염" 호스트를 식별한다. 이 단계 동안 호스트에 대한 실패 통계를 저장하기 위하여 해시(hash) 테이블이 사용된다. 라우터에서 보관되는 호스트-별 상태의 양을 제한하기 위하여 심하게 경쟁하는 해시 테이블이 사용된다. 해시 테이블에서 한 항목에 대한 실패율이 어떤 경계를 초과하면, 알고리즘은 두 번째 단계로 들어간다. 이 단계에서 특정 해시 항목에서 호스트에 대한 rate limit를 시도한다. [7]에서는 "기본"과 "임시" rate limiting 알고리즘을 제한하였다.

- 기본(basic) FC 알고리즘: 이 알고리즘은 단기간 실패율(λ)에 초점을 맞춘다. 해시 항목이 초당 λ(예를 들어, 한개) 실패를 초과하면, 에지 라우터에 있는 엔진이 해시 항목에 있는 각 호스트의 실패율을 최대 λ로 제한하기 위한 시도를 한다. 각 호스트에서 rate limit을 위하여 리키 버킷(leaky bucket) 알고리즘이 사용된다. 각 실패된 연결에 대하여 버킷으로부터 하나의 토큰을 제거하며, 매 λ초마다 하나의 새로운 토큰이 버킷에 추가된다. 특정 호스트에 대한 버킷이 비면, 해당 호스트로부터의 추가적인 외부 연결은 탈락(drop)된다.
- 임시(temporal) FC 알고리즘: 이 기법에서는 단기간 실패율뿐만 아니라 장기간 율 Ω(일 별 율)을 제한한다. Ω의 값은 $\lambda\times$(하루의 초 단위 시간)보다 훨씬 작게 유지된다. 항목의 실패율이 초당 λ 혹은 일당 Ω를 초과하면 rate limiting을 받게 된다. 이 알고리즘의 목적은 λ 율 이하로 확산되는 다소 덜 공격적인 스캐닝 행위를 가진 웜을 잡기위한 것이다. 장기간 실패율을 비교적 낮게 유지하면서 실패 연결의 일시적인 군집성(burstiness)을 수용하기 위하여 λ 값을 더 높은 값으로 조정할 수 있다.

6.3 크레디트-기반(CB: Credit-based) rate limiting

CB 기법은 FC 기법과 두 가지 측면에서 상당히 다르다.[14] 먼저 CB 기법은 첫 번째 연결 즉, 이전에 방문한 적이 없는 주소들에 대한 외부 연결에 대하여만 한정하여 rate limiting을 수행한다. 이것은 스캐닝 웜이 많은 양의 실패 연결을 생성하는데, 대부분 실패 일차-접촉 연결 수로 이루어진다는 관측을 바탕으로 한다. 따라서 비정상 일차-접촉 통계치가 rate limiting을 유발하는 스캐닝 행위의 표시이다. 일차 접촉 개념은 CB에 기본적이다. 두 번째로, CB는 실패와 성공 연결 통계치 모두를 고려한다. 간단히 말해서, CB는 호스트 당 어떤 수의 연결 크레디트를 할당한다. 각 실패 일차-연결은 한 개의 크레디트를 소모하며, 성공적인 일차 연결은 한 개를 추가한다. 호스트는 자신의 남은 크레디트가 양수일 때만 일차-접촉 연결이 허용된다.

CB는 각 호스트에서의 일차-접촉 실패율을 제한하지만, 남은 크레디트가 양수이면 성공적인 연결의 수를 제한하지 않는다. 게다가, 크레디트 잔량에 관계없이 모든 비일차-접촉 연결(즉 합법 트래픽)은 허용된다. 결국 성공하지 못하는 TCP 요구를 많이 발생하는 스캐닝 웜은 자신의 크레디드 잔량을 급속히 소모하게 되고 포획된다. 합법적인 애플리케이션은 보통 이전에 접속한 주소들에 접촉하기 때문에, rate limiting 기법에 의하여 대부분 영향을 받지 않는다.

CB 기법은 호스트 별 통계치 위에서 운용되는 에지-라우터 구현이다. 에지 라우터에서 개별 내부 호스트에 대한 크레디트 은행을 유지한다.

6.4 DNS-기반 rate limiting

DNS-기반 스킴은 웜 프로그램이 합법적인 애플리케이션과는 분명히 다른 DNS 통계치를 유발한다는 원칙에 기반을 두고 있다.[19] 예를 들어, DNS 룩업이 존재하지 않는 것이 스캐닝 활동의 징표이기 때문이다. 이 관측은 Ganger[9]에 의하여 처음 행하여 졌다. [22]에 의하여 제안된 기법은 Ganger의 NIC-기반 DNS 탐지 기법의 변형이다.

DNS 기반 기법은 모든 출력 TCP SYN에 대하여, 목적지 IP에 대하여 이전의 DNS 번역이 존재하면 패킷을 통과시키고, 그렇지 않으면 SYN 패킷이

rate limit된다. 알고리즘은 다음과 같이 동작한다.

TCP SYN이 사전(prior) DNS 번역이 없는 주소로 전송될 때, 목적지 IP가 현재 시간 간격 동안 버킷(bucket)에 추가되며 패킷은 지연된다. 버킷은 번역이 안 된 IP 연결을 포획하기 위한 것이다. 한 버킷에 q 개의 다른 IP가 채워지면, 새로운 연결 요구는 후속(subsequent) 버킷에 위치한다. 즉 각 버킷은 다음 버킷에 직렬로 연결된다. 마지막 n번째 버킷은 오버 플로 버킷이 없으며, 버퍼가 일단 차면 DNS 번역 없는 새로운 TCP SYN 패킷은 단순히 탈락된다. i-번째 버킷 안에 있는 패킷들은 i+1 시간 간격의 시작까지 지연된다. 모든 n 버킷이 만료되면, 다음 $n \times t$ 시간 기간 동안 본래 데로 된다. 이 알고리즘은 시간 간격 t 당(DNS 번역 없이), 최대 q 개의 다른 IP 만큼 허용하며, 탈락되지 않는 패킷들은 최대 $n \times t$까지 지연된다.

이 DNS throttling 기법은 개별 호스트나 개별 라우터에서 구현될 수 있다. 호스트-레벨 구현은 각 호스트 상에서 DNS-관련 통계치를 유지함으로써 이루어진다. 에지-라우터 기반 구현은 DNS 정보에 대한 접근을 지원하기 위하여 에지 라우터 상에서의 세도우(shadow) DNS 캐시를 필요로 한다.

Ⅶ. 결 론

분산 서비스거부 공격을 초래하는 대규모 웜 공격은 인터넷 인프라구조 보안에 심대한 위협이 되고 있다. 이에 따라 세계적으로 네트워크 공격에 대한 손해를 최소화하고 플러딩 공격을 방지하기 위하여 웜 봉쇄와 같은 보안 대책을 강구하고 있다.[1,2] 본 논문에서는 웜 공격 특성에 대하여 분석을 하고, 웜의 공격을 지연시키거나 방지할 수 있는, rate-limiting 기술을 포함한 웜의 봉쇄 기술에 대하여 살펴보았다.

국내에서 현재 구축되고 있는 광대역 통합망(BcN)과 같은 초고속 통합망 환경에서는 네트워크 자원이 여러 가지의 침입 행위에 쉽게 노출될 수 있는 위협이 존재하며, 또한 네트워크 대역폭의 증가로 인하여 웜의 확산을 가속화 시킬 수 있는 위험성도 있다.

그러므로 국내에서도 BcN 환경에서 웜의 발생 시 웜의 확산을 방지하기 위한 웜 봉쇄 기술에 대하여 체계적인 연구가 필요하다고 사료된다.

참 고 문 헌

[1] 전용희, 장종수, 손승원, "미국의 정보인프라 보호 연구개발 동향 분석", 전자통신동향분석, 제 19권 제4호, pp.96-108, 한국전자통신연구원, 2004년 8월.

[2] 전용희, 손승원, "정보통신 인프라 보호를 위한 미국의 연구개발 동향", 한국통신학회지 제 21권 제9호, pp.1033-1047, 2004년 9월.

[3] 전용희, "인터넷 웜의 탐지 및 대응기술", 한국통신학회지 제22권 제8호, pp. 1088-1103, 2005년 8월.

[4] 신승원, 오진태, 김기영, 장종수, "인터넷 웜 공격 탐지 방법 동향", 전자통신동향분석, 제 20권 제1호, pp.9-16, 2005년 2월.

[5] David J. Albanese et al., The Case for Using Layered Defenses to Stop Worms, Report #C43-002R-2004, Ver. 1.0, June 18, 2004, National Security Agency, USA.

[6] Min Cai et al., "Collaborative Internet Worm Containment", IEEE Security and Privacy, pp.24-33, May/June 2005.

[7] Shigang Chen and Yong Tang, "Slowing Down Internet Worms", Proc. of 24th IEEE International Conference on Distributed Computing Systems(ICDCS'04), Tokyo, Japan, March 2004.

[8] Z. Chen, L. Gao, and K. Kwiat, "Modeling the Spread of Active Worms", INFOCOM, pp.1890-1900, April 2003, California.

[9] Gregory R. Ganger, Gregg Economou, and Stanley M. Bielski, Self-Securing Network Interfaces: What, Why and How. Technical Report.

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

[11] Hyang-Ah Kim and Brad Karp, "Autograph: Toward Automated Distributed Worm Signature Detection", Proc. of Usenix Security Sym., pp.271-286, 2004.

[12] D. Moore et al., "Internet Quarantine: Requirements for Containing Self-Propagating Codes", Proc. IEEE INFOCOM, pp.1901-1910, IEEE CS Press, 2003.
[13] Jose Nazario, *Defense and Detection Strategies against Internet Worms*, Artech House, 2004.
[14] Stuart E. Schechter, Jaeyoun Jung, and Arthur W. Berger, "Fast Detection of Scanning Worm Infections", Proc. of 7th International Symposium on Recent Advances in Intrusion Detection (RAID), Sep. 2004, France.
[15] Stelios Sidiroglou and Angelos D. Keromytis, "Countering Network Worms Through Automatic Patch Generation", IEEE Security and Privacy, 2005.
[16] Sumeet Singh et al, "Automated Worm Fingerprinting", Proc. of Usenix Symp. Operating System Design and Implementation, pp.45-60, 2004.
[17] Ed Skoudis, Lenny Zeltser, *Malware: Fighting Malicious Code*, Prentice-Hall, 2004.
[18] S. Staniford, V. Paxon and N. Weaver, "How to Own the Internet in Your Spare Time", 11th Usenix Security Symposium, San Francisco, August 2002.
[19] Helen J. Wang et al., "Shield: Vulnerability-Driven Network Filters for Preventing Known Vulnerability Exploits", Proc. of ACM SIGCOMM '04, Aug. 30-Sep. 3, 2004, USA.
[20] N. Weaver, S. Staniford, and V. Paxson, "Very Fast Containment of Scanning Worms", Proc. of 13th USENIX Security Symposium, pp.29-44, August 2004, California.
[21] Matthew M. Williamson, "Throttling Viruses: Restricting propagation to defeat malicious code", Proc. of ACSAC Security Conference, pp.61-68, 2002.
[22] Cynthia Wong, Chenxi Wang, Dawn Song, Stanley M. Bielski, and Gregory R. Ganger, "Dynamic Quarantine of Internet Worms", The International Conf. on Dependable Systems and Networks (DSN-2004), pp.62-71, 2004.
[23] Cynthis Wong, Stan Bielski, Ahren Studer, Chenxi Wang, On the Effectiveness of Rate Limiting Mechanisms, CMU-PDL-05-103, Carnegie Mellon University, March 2005.
[24] Cliff C. Zou, Weibo Gong, Don Towsley, "Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense", ACM WORM '03, Washington DC, USA, Oct.2003.
[25] Cliff C. Zou et al., "Monitoring and Early Warning for Internet Worms", Proc. of 10th ACM Conf. Computer and Comm. Security(CCS 03), pp. 190-199, Oct. 2003.

〈著 者 紹 介〉

전 용 희(Yong-Hee Jeon)

1971.3~1978.2 고려대학교 전기공학과
1985.8~1987.8 미국 플로리다 공대 대학원 컴퓨터공학과
1987.8~1992.12 미국 노스캐롤라이나주립대 대학원 Elec. and Comp. Eng. 석사, 박사
1978. 1~1978.11 삼성중공업(주)
1978.11~1985.7 한국전력기술(주)
1979.6~1980.6 벨기에 벨가톰사 연수
1989.1~1989.6 미국 노스캐롤라이나주립대 Dept of Elec. and Comp. Eng. TA
1989.7~1992.9 미국 노스캐롤라이나주립대 부설 CCSP (Center For Comm. & Signal Processing) RA
1992.10~1994.2 한국전자통신연구원 광대역통신망 연구부 선임연구원
1994.3~현재 대구가톨릭대학교 컴퓨터·정보통신공학부 교수

2001.3~2003.2 대구가톨릭대학교 공과대학장 역임
2004.2~2005.2 한국전자통신연구원 정보보호연구단 초빙연구원
관심분야: 네트워크 보안, BcN QoS & Security, 웜 모델링 및 대응 기술, 통신망 성능분석