인터넷 웜 공격 탐지 방법 동향

A Survey of Worm Detection Techniques

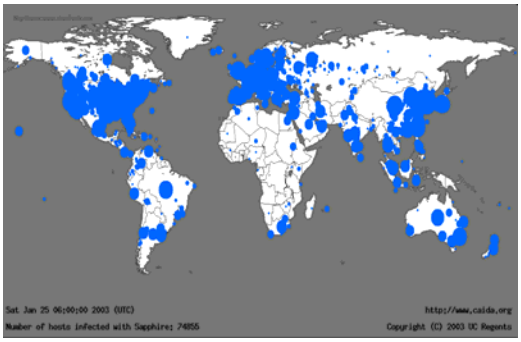
신승원(S.W. Shin)	보안게이트웨이연구팀 연구원
오진태(J.T. Oh)	보안게이트웨이연구팀 선임연구원
김기영(K.Y. Kim)	보안게이트웨이연구팀 책임연구원, 팀장
장종수(J.S. Jang)	네트워크보안그룹 책임연구원, 그룹장

최초의 인터넷 웜(worm)으로 불리는 Morris 웜이 1988년 11월에 발표된 이래로 현재까지 많은 웜 공격이 발생되고 또 발표되어 왔다. 초기의 웜은 작은 규모의 네트워크에서 퍼지는 정도였으며, 실질적인 피해를 주는 경우는 거의 없었다. 그러나 2001년 CodeRed 웜은 인터넷에 연결된 많은 컴퓨터들을 순식간에 감염시켜 많은 피해를 발생시켰으며 그 이후 2003년 1월에 발생한 Slammer 웜은 10분이라는 짧은 시간안에 75,000여 대 이상의 호스트를 감염시키고 네트워크 자체를 마비시켰다. 특히 Slammer 웜은 국내에서 더욱 유명하다. 명절 구정과 맞물려 호황을 누리던 인터넷 쇼핑 몰, 은행 거래 등을 일시에 마비시켜 버리면서 경제적으로도 막대한 피해를 우리에게 주었다. 이런 웜을 막기 위해서 많은 보안 업체들이 나서고 있으나, 아직은 사전에 웜의 피해를 막을만한 확실한 대답을 얻지 못하고 있다. 본 문서에서는 현재 웜의 발생 초기 단계를 탐지하고 이를 피해가 커지기 이전에 막기 위한 연구들을 설명할 것이다.

I. 서 론

네트워크는 엄청난 진화를 해왔다. 수 kbps에 이르던 전송 속도는 Mbps를 넘어서 이제는 Gbps를 지나가고 있으며, 처음 두 대학간을 연결하는 것으로 시작된 인터넷은 전세계를 연결하고 있다. 가히 전세계의 거의 모든 사람이 인터넷이라는 매체로 서로 연결되어 있으며, 동시에 이런 기회를 가질 수 있는 환경이 된 것이다. 그리고 이런 진화와 발전은 인터넷을 이용하는 사람들에게 많은 기회를 안겨 주고 있다. 서로 마주보며 사고 팔던 물건들을 인터넷 상에서 화면으로 보면서 거래를 할 수 있게 됐으며, 돈을 찾고 예금하기 위해 은행으로 향하던 사람들은 간단하게 웹 브라우저를 통해서 돈을 넣고 옮기고 확인할 수 있게 되었다. 사람들은 인터넷이라는 가상의 세계를 통해서 또 다른 세상을 만들어 가고 있는 것이다. 그러나, 이런 기회의 세계 역시 수 많은 위험이 도사리고 있는 현 세상처럼 그 자체로도 여러 가지 위험성을 지니고 있다. 안전하리라 믿었던 인터넷을 통해서 사람들은 자신의 정보를 타인이 훔쳐 보는 것을 겪어야 했고, 이익을 위해 구축한 인터넷 서버들이 분산 서비스 거부 공격(DDOS)과 같은 것에 맥없이 무너져 멈춰 버리는 것을 경험하게 되었다. 가상의 세계에서 가상의 경제가 아닌 실제 돈이 오가고 개인의 실제 정보가 오가는 상황에서 이러한 피해는 현실의 피해와 별반 차이가 없을 것이다. 따라서 이러한 피해를 막기 위하여 네트워크 보안 업체들이 등장하고, 바이러스 백신, 방화벽, 침입 탐지/방지 시스템, 통합 보안관리 시스템 등이 등장하게 되었다. 이러한 시스템들은 여러 해커들의 침입 시도나 무분별한 분산 서비스 거부 공격들로부터 우리의 시스템들을 잘 보호해 왔고, 앞으로도 그럴 것이다.

그러나 이런 시스템들도 최근 (실질적으로는 1988년부터) 네트워크 상에 급속하게 퍼지고 있는 인터넷 웜에 대해서는 1차적인 차단을 하지 못하고 있는 실정이다. 웜은 다른 네트워크 공격과는 달리 스스로 자신을 복제하여 네트워크 상에 연결된 다른 컴퓨터들로 스스로 자신을 전송하는 프로그램이다. 맨



(그림 1) Slammer 웜에 감염된 호스트 분포

처음 Morris 웜이 등장하였을 때는 그 속도나 파급력이 크지 않았으나, 2001년의 CodeRed나 2003년의 Slammer 웜 같은 경우 대다수의 보안 장비들을 무력화시키고, 급속도로 네트워크로 퍼져 나가서 엄청난 사회적 피해를 주었다. Slammer 웜의 경우, 전세계에 미친 여파는 엄청났다. (그림 1)에서 볼 수 있는 바와 같이 10분 동안, 전세계에 있는 75,000개 이상의 호스트를 감염시켰으며, 스캐닝을 위한 엄청난 양의 UDP 패킷은 호스트 감염뿐만 아니라 네트워크 자체를 마비시켜 버렸다. 특히 국내의 경우 2003년 1월 25일 구정과 맞물려 1.25대란이라 불리게 될 정도로 많은 경제적 사회적 피해를 안겨 주었다.

그렇다면 이렇게 많은 피해를 주고 있는 웜은 어떤 것이고 도대체 어떤 방법으로 탐지를 해야 하는 것일까? 현재까지 완벽한 답은 아직 없지만, 웜에 대한 탐지 및 대응을 위해서 많은 보안 회사들이 팔을 걷어 붙여 나섰고, 많은 연구 기관이나 대학에서 이들에 대한 조사 및 연구가 이루어지고 있다. 본 문서에서는 이런 현재의 연구와 실제 사례를 들어 보이려고 한다. 먼저, II장에서는 인터넷 웜에 대한 간략한 정의와 소개, 그리고 웜의 종류 등에 대해서 언급할 것이다. 그리고 III장에서, 웜의 탐지 방법에 대한 것을 언급할 것이며, 마지막으로 IV장에서 결론을 맺으려 한다.

II. 인터넷 웜이란?

본 장에서는 인터넷 웜의 정의와 그리고 이들의 종류 등에 대해서 간단하게 언급할 것이다.

1. 웜의 정의

웜은 스스로 자신을 복제하여 네트워크 상에 연결된 다른 컴퓨터들로 스스로 자신을 전송하는 프로그램이다. 바이러스처럼 스스로 자신을 복제하긴 하지만, 바이러스는 보통 사람의 행동에 의해서 (파일 복사 등) 퍼지게 되지만 웜의 경우, 자신을 스스로 복사해서 네트워크 상에 연결된 곳에 스스로를 전파시킨다는 점에서 차이를 보인다. 보통 웜의 경우 스캐닝 단계와 전파 단계를 통해서 감염된다. 스캐닝 단계는 웜이 자신이 감염시킬 수 있는 네트워크에 연결된 호스트를 찾는 단계로, TCP 프로토콜을 사용하는 웜의 경우 TCP SYN 패킷을 보내서 응답을 받아서 스캐닝 정보를 받곤 한다. UDP의 경우엔 UDP request 메시지를 보내고 이에 대한 응답 메시지를 통해서 그 정보를 확인할 수 있다. 그리고 좀 더 진화한 웜의 경우 스캐닝과 감염을 동시에 시키는 경우도 있다. 특히 UDP 프로토콜을 이용하는 경우가 많은데, 스캐닝 패킷에 자신의 전체 패킷을 담아서 스캐닝을 하면서 응답을 하는 호스트에 바로 자신의 복제 이미지를 전송하게 되는 것이다.

2. 웜의 종류

현재 웜을 분류하는 시도는 크게 웜의 외부적인 특성 (전파되는 방식)을 바탕으로 분류한 것과[1], 웜 외부적인 특성 및 내부의 동작을 바탕으로 분류한 것이 있다[2]. 웜 내부의 동작을 바탕으로 분류하는 것은 그 분석과 내용이 복잡하기에 우선 분류하기 쉬운 외부적인 특성을 바탕으로 분류한 예를 들어 보자. Symantec 사의 Darrell M. Kienzie와 Network Associates 사의 Matthew C. Elder는 웜을 크게 E-Mail 웜, Windows File Sharing 웜, 그 외 일반적인 웜의 세 가지로 나누었다. 먼저 이메일 웜은 말 그대로 이메일을 통해서 전파되는 웜에 대한 것이다. 이메일에 첨부된 파일 형태로 전파되는 것으로, 사용자가 이메일을 보고 실행파일을 실행해야만 전파되는 것부터, 이메일을 여는 순간 감염되는 것까지 다양한 종류가 있다. 이런 웜들로는

Christmas Tree 웜이나 Sircam 등의 웜이 있다. 그리고, Windows File Sharing 웜의 경우 운영체제가 windows 계열인 경우에 발생하는 것으로써, windows 운영체제에서 file sharing(파일 공유) 관련 네트워크 서비스를 실행시켰을 때 이들을 통해서 감염되는 웜에 대한 것이다. 이런 웜들로는 Net-Log 웜이나, Shorm 웜 등이 있다. 그리고 마지막으로 일반적인 웜들은 위의 두 가지 경우가 아닌 다른 방식으로 전파되는 웜들을 말하는 것으로, 우리에게 많이 익숙한 CodeRed나 Slammer 등과 같은 웜들이 있다. 이에 대한 전체적인 정리는 <표 1>에 요약되어 있다[2].

그리고 좀 더 복잡하게 분류한 방식은 UC. Berkeley 대학의 Nicholas Weaver와, ICSI의 Vern Paxson, Silicon Defense사의 Stuart Staniford, 그리고 MIT Lincoln Lab의 Rebert Cunningham이 제안한 것이다. 이들은 웜을 분류할 때 5가지 분류를 이용하였는데, 그 내용을 보면, 먼저 웜이 자신이 감염시키고자 하는 호스트를 찾는 방법에 의한 분류, 그리고 웜이 전파되는 네트워크 매개체에 의한 분류, 그리고 웜이 활동하는 방식에 의한 분류, 그리고 웜의 내부에 있는 데이터에 대한 분류, 그리고 마지막으로 웜 프로그램의 의도와 같은 사회학적 분류 체계를 작성하였다. 그러나, 이들은 실제적인 웜의 분류보다는 웜을 분류할 수 있는 체계적인 분류 방법을 제시한 것으로 차후 발생하는 웜이나 기존의 웜을 분류하고자 할 때 하나의 기준으로 활용할 수 있을 것이다. 이에 대한 좀 더 자세한 설명은 [2]에 서술되어 있다.

Ⅲ. 인터넷 웜 탐지 기법

앞장에서 서술한 웜의 정의와 웜의 종류에 대한 것을 바탕으로 이번 장에서는 웜을 탐지하는 기법들에 대해서 알아볼 것이다.

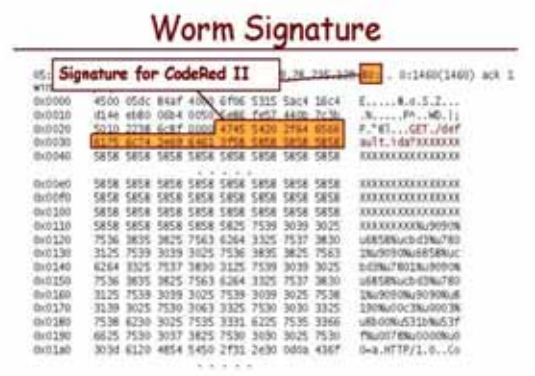
1. 알려진 웜에 대한 탐지 방법

말 그대로 이미 널리 알려진 웜에 대한 탐지를 위한 것이다. 이것은 기존의 패턴 매칭 방식으로 네트워크 공격을 탐지하던 것으로 대부분의 침입 탐지/방지 시스템에서 적용이 가능하다. 이는 먼저 웜이 퍼지고 난 후 이들에 대한 정보를 수집해서 이런 정보를 바탕으로 웜에 대한 signature를 생성한 후 이것을 가지고 탐지하는 방식이다. (그림 2)에서와 같

<표 1> 웜 요약표

E-Mail Worm		Windows File Sharing Worm		기타 Worm	
Worm Name	Discovery Date	Worm Name	Discovery Date	Worm Name	Discovery Date
Christmas Tree	1987. 12.	Many File infecting viruses	Pre-1999.	Morris	1988. 11.
ShareFun	1997. 2.	ExploreZip	1999. 6.	ADM	1998. 5.
Melissa	1999. 3.	NetLog	2000. 2.	RAtmen	2001. 1.
ExploreZip	1999. 6.	Shorm	2001. 1.	Lion	2001. 3.
LoveLetter	2000. 3.	Nimda	2001. 9.	BoxPoison	2001. 5.
Stages	2000. 5.	Ladex	2002. 7.	Cheese	2001. 6.
Magistr	2001. 3.	Opaserv	2002. 9.	CodeRed	2001. 7.
Sircam	2001. 7.	Gaobot	2002. 10.	Walk	2001. 8.
Nimda	2001. 9.	Lioten	2002. 12.	Nimda	2001. 9.
Goner	2001. 12.	Netspree	2003. 1.	Scalper	2002. 6.
Bibrog	2003. 1.			Slammer	2003. 1.

주) 같은 이름의 웜이 여러 개 있는 경우는 이 웜이 하나의 형태만을 가진 것이 아니라 여러 개의 변종 혹은 다양한 형태의 종류를 가지고 있기 때문임



(그림 2) CodeRed 웜의 Signature

이 CodeRed 웜의 경우 ASCII 값으로 이루어진 패턴이 발생하게 된다. 그리고 이는 signature가 되어, 차후에 다른 패킷들 내에서 이러한 패턴이 발견되면 이를 CodeRed 웜으로 간주하게 되는 것이다.

이러한 방식은 웜을 false-positive가 적게 (정확하게) 탐지할 수 있는 확률이 높다는 장점이 있으나, 반대로 이미 알려진 웜에 대해서만 탐지가 가능하다는 단점이 있다. 따라서, 새로 발생하는 웜에 대해서는 탐지할 수 없고, 만약 Slammer 웜처럼 엄청난 속도로 네트워크에 연결된 호스트를 감염시키면서 네트워크를 동시에 마비시키는 웜들에 대해서는 그 대처가 한계가 있다고 할 수 있다.

이 탐지 방법은 open source 네트워크 침입 탐지 시스템인 Snort[3]를 비롯하여, 대부분의 상용 침입 탐지 혹은 방지 시스템들에서 적용이 가능하다.

2. 비정상 트래픽 특성 분석을 이용한 탐지 방법

이 방법은 알려진 웜보다는 알려지지 않은 새로운 웜의 가능성을 차단하기 위한 것으로, 네트워크 상을 오가는 트래픽의 특성을 분석하고 이들 중 웜으로 생각되는 트래픽 패턴을 찾아서 이를 웜 공격으로 간주하는 것이다. 이러한 방식은 이전에 발생하지 않은 웜 공격이라 할지라도 미리 발견하여 차단할 수 있다는 장점이 있다. 따라서, Slammer 웜과 같은 웜 역시 조기에 발견하여 사전에 차단할 수 있게 된다. 그러나, 이러한 방법은 아직 정확한 탐지가 어렵고 (오탐율이 상대적으로 높음) 그 구현이 쉽지가 않다는 단점이 있다. 그러나, 현재 알려진 웜을 탐지하는 것만으로는 네트워크를 보호할 수 없기 때문에 비정상 트래픽 특성 분석을 이용한 탐지 방법은 현재 많은 연구가 되고 있으며 앞으로도 계속 발전하리라 본다. 본 장에서는 현재 각 연구소 및 대학 등지에서 연구되고 있는 비정상 트래픽 특성 분석을 이용한 탐지 방법에 대해서 좀 더 알아보려 한다.

가. TRW(MIT)

MIT에서 이루어지고 있는 연구로 수학적인 알고리듬인 TRW(Threshold Random Walk) 방법[4]을 이용해서 트래픽 패턴을 분석해서 웜을 찾아내는 방법을 제시하고 있다. 이 TRW 방식은 TCP 프로토콜을 사용한 웜에 대한 것으로, TCP 프로토콜의 처음 연결을 위한 SYN 패킷을 스캐닝 동작으로 보고 이 정보를 바탕으로 Sequential Hypothesis Testing 방식을 이용해서 비정상 트래픽 패턴을 찾아내게 된다. 이 TRW 알고리듬은 다음과 같다.

먼저, Y라는 값을 어떤 특정 호스트의 TCP 연결의 성공과 실패를 나타내는 연속적인 값이라고 하자. 그러면, Y는 (1)과 같이 나타내어 질 수 있다.

$$\begin{aligned} Y_i &= S(0), \text{ 연결이 성공한 경우} \\ Y_i &= F(1), \text{ 연결이 실패한 경우} \end{aligned} \quad (1)$$

그러면, 우리는 (1)을 바탕으로 어떤 호스트 H에 대해서 스캐닝이 발생한 경우와 아닌 경우를 (2)와 같은 확률로 나타낼 수 있다.

$$\begin{aligned} \Pr[S \mid H_{\text{scanning}}] &< \Pr[S \mid H_{\text{benign}}] \\ \Pr[F \mid H_{\text{scanning}}] &> \Pr[F \mid H_{\text{benign}}] \end{aligned} \quad (2)$$

그리고 이에 대한 Likelihood Function을 작성하여 본다면 (3)과 같이 정의할 수 있다.

$$\begin{aligned} \phi(S) &= \frac{\Pr[S \mid H_{\text{scanning}}]}{\Pr[S \mid H_{\text{benign}}]} < 1 \\ \phi(F) &= \frac{\Pr[F \mid H_{\text{scanning}}]}{\Pr[F \mid H_{\text{benign}}]} > 1 \end{aligned} \quad (3)$$

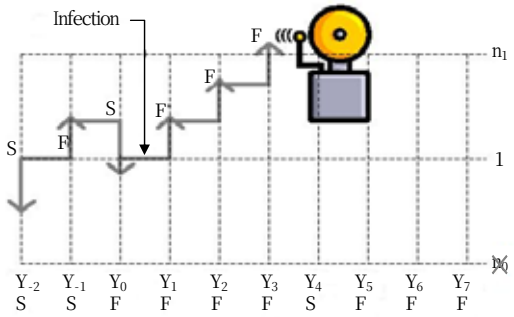
따라서, (3)을 Y에 대한 식으로 변환하면 (4)와

같이 정리할 수 있다.

$$\phi(Y_i) = \frac{\Pr[Y_i \mid H_{\text{scanning}}]}{\Pr[Y_i \mid H_{\text{benign}}]}$$
$$\Lambda(Y) = \prod_{i=1}^{n} \frac{\Pr[Y_i \mid H_{\text{scanning}}]}{\Pr[Y_i \mid H_{\text{benign}}]} = \prod_{i=1}^{n} \phi(Y_i) \quad (4)$$

이렇게 정리된 수식을 바탕으로 실제 호스트 *H*에서 TCP 연결이 발생하는 경우, 만약 연결이 성공하게 되면 위의 Λ(*Y*)는 TCP 연결이 성공하는 경우 양의 값(+)을 나타내게 되고, 반대로 연결이 실패하는 경우 음의 값(-)을 나타내게 된다. 따라서, 어떤 호스트 *H*에서 TCP 연결에 대한 SYN 패킷을 내보낼 때마다 위의 값이 변하게 되고, 호스트 *H*에서 계속 TCP 연결을 실패하는 경우 이 값이 계속 증가하게 된다(양의 값이 계속 증가하기 때문에). 이 때 이 증가하는 값이 threshold 값보다 크게 되면 (그림 3)처럼 웜 공격으로 간주하게 된다.

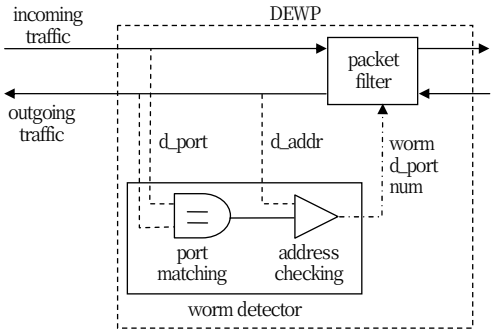
TRW 방식은 수식은 복잡하지만, 실제 구현 복잡도는 높지 않으며, 빠르게 TCP 웜을 잡아낼 수 있다는 장점이 있다. 그러나, UDP 웜에 대한 탐지 방법이 없다는 단점이 있다.



(그림 3) TRW 알고리듬을 이용한 탐지

나. DEWP(ISI)

DEWP(Detecting Early Worm Propagation through Packet Matching)[5] 방식은 ISI(Information Sciences Institute)에서 제안한 방식으로, 네트워크 상에 유입되는 트래픽들을 특정 패턴들로 분류하여 웜을 탐지하는 방식이다. 보통 정상적인 트래픽에 비해서 웜과 같은 경우 특정 네트워크 서비스 포트로 유입되는 트래픽의 양이 많게 된다. 이 방식은 이러한 성질을 이용해서 특정 포트가 다른 포트에 비해서 혹은 이전에 비해서 갑자기 많은 트래픽들을 받게 되면 이를 웜으로 의심하고 이를 탐지하는 것이다. 이에 대한 구조는 (그림 4)와 같다.



(그림 4) DEWP 구조

(그림 4)의 구조에서 알 수 있듯이 DEWP에서는 네트워크 패킷의 데이터를 볼 필요는 없다. 단지 특정 네트워크 포트로 전달된 패킷들과 그 포트에서 다시 외부로 전송된 패킷들에 대한 정보만 있으면 된다. 즉, DEWP 방식에서는 매 T 시간동안 특정 포트로 전달된 패킷의 양과, 그 포트에서 나간 양을 비교하여 EWMA(Exponential Weighted Moving Average) 알고리듬을 이용하여 변화량을 추적하게 되고 변화가 발생하면 이를 웜 공격으로 간주하게 되는 것이다. 먼저, 트래픽 양의 변화를 찾기 위하여 (5)와 같은 수식을 정의한다.

$$N' = \alpha \times N' + (1 - \alpha) \times N \quad (5)$$

이 때, N은 특정 포트로 유입되는 트래픽의 근원지 주소들의 수를 말하고 N′은 이들의 평균적인 수치를 말한다. (5)에서 만약 $N > N' \times (1+\sigma)$인 조건을 만족하게 되면 이를 웜으로 간주하게 되는 것이다. 이 때 σ는 표준 편차를 의미한다. 즉, 전체적인 알고리듬을 정리하면, 특정 네트워크 포트로 유입되는 패킷의 근원지 주소들의 수가 갑자기 많아지는 경우 이를 웜으로 간주하는 것이라 할 수 있다.

이 방식의 경우 쉽고 간편하게 웜을 찾아 낼 수 있다는 장점이 있으나, 핫스폿과 같이 갑자기 외부 접속자가 많아지는 사이트의 경우 웜과 일반 사용자

의 빈도가 비슷해지는 경우가 발생하기 때문에 이에 대한 탐지를 잘못할 경우가 많이 일어날 수 있다는 단점이 있다.

다. Statistical Intrusion Detection 방식
(University of Massachusetts, Amherst)

Statistical Intrusion Detection 방식[6]은 말 그대로 통계학적인 방식을 이용하여 웜을 탐지하는 방식이다. 이 기법에서 주로 사용한 모델은 실제 전염병 같은 것의 전파를 설명할 때 쓰이는 Epidemic Model 이다. 이 연구에서는 이 Epidemic Model을 바탕으로 웜의 전파를 설명하고 실제 네트워크 트래픽을 가지고 모델을 만들어서 웜의 모델과 일치하는 경우 이를 웜으로 판단하게 된다. 먼저 Epidemic Model을 만들기 위해서 이들은 CodeRed와 Slammer 웜에 대한 자료를 수집해서 실제 웜의 모델을 만들어 냈다. 먼저 이 연구에서 정리한 웜의 모델에 대해서 살펴 보자.

Simple Epidemic Model

$$dI(t)/dt = \beta I(t)S(t) = \beta I(t)[N(t) - I(t)] \qquad (6)$$

여기에서 $I(t)$는 시간 t에서 감염된 호스트의 수이고, N은 모집단의 크기, β는 감염률을 나타낸다. 위의 모델은 기본적인 Epidemic Model로서 실제 전염병이 퍼지는 경우 만약 어떤 호스트가 전염병에 감염이 되면 계속 감염이 되어 있다고 가정하게 된다. 따라서 한 번 감염된 호스트가 다시 치료될 수도 있다는 것을 고려하지 않는 등의 문제로 실제 환경에서 적용하기에 어려움이 있다. 따라서 이 연구에서는 이 모델을 개선한 Kermack-Mckendrick Epidemic Model을 이용하였다.

Kermack-Mckendrick Epidemic Model

$$\begin{aligned} dI(t)/dt &= \beta I(t)S(t) - \gamma I(t) \\ dU(t)/dt &= \gamma I(t) \\ N &= I(t) + U(t) + S(t) \end{aligned} \qquad (7)$$

여기에서 β는 감염률을 나타내고, γ는 제거된 감염된 호스트의 제거율, $S(t)$는 시간 t에서 취약한 호스트의 수, N은 모집단의 크기를 나타낸다.

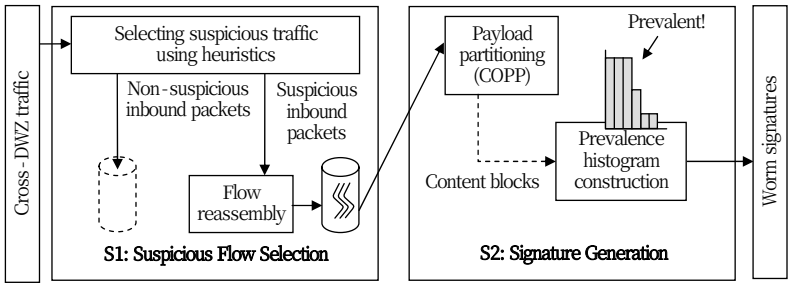
이 연구에서는 이러한 모델들을 이용해서 실제 웜 데이터를 바탕으로 이 모델들의 일치성을 확인한 후 파라미터들을 조정하여 실제 값과 유사하게 만들어 냈다. 그리고, 이러한 모델은 차후에 실제 네트워크 트래픽을 대상으로 만들어진 모델과 비교 기준이 되고, 만약 실제 네트워크 트래픽으로부터 만들어진 모델이 웜으로부터 만들어진 모델과 유사하다면 이를 웜으로 판별하게 된다.

이러한 방식은 웜을 보다 정확하게(수학적인 근거를 바탕으로) 탐지하는 것이 가능하지만, 실제 구현하기에 복잡도가 높은 단점이 있다.

라. Autograph Project(CMU)

Autograph Project는 Carnegie Mellon University[7]에서 현재 수행되고 있는 프로젝트로 비정상 트래픽 탐지 방식으로 웜을 탐지하기 위한 방법이라고 하기보다는 웜으로 추측되는 패킷들을 잡아서 자동으로 웜에 대한 signature를 생성하고자 하는 것이다. 이 프로젝트의 주 관심사가 비정상 트래픽에 의한 탐지는 아니지만, 웜에 대한 탐지 결과를 바탕으로 자동으로 signature를 생성해서 빠른 시간 안에 웜을 차단하고자 하는 근본 목표는 다른 프로젝트와 같다고 할 수 있다. 그리고 생성된 signature는 여타 다른 침입 탐지 혹은 방지 시스템에 그대로 적용될 수 있다. 비정상 트래픽 탐지 방식을 제외하고 웜에 대한 signature를 생성하는 부분에 대한 autograph의 기본 구조는 (그림 5)와 같다.

(그림 5)에서 볼 수 있듯이 autograph에서는 웜이라고 의심되는 패킷들의 데이터를 따로 모아서 COPP라는 것을 작성하게 된다. 그리고 이를 모아서 유사성을 찾는 알고리듬을 이용해서 공통 패턴을 찾아내게 되는데 이때 Rabin's Fingerprint[8] 알고리듬을 이용하게 된다. 이 알고리듬은 간단한 모듈러 연산을 이용해서 공통 부분을 찾아내게 되는데, 실제 구현 역시 매우 쉽고, 동작 또한 빠르기 때문에 많은 곳에서 활용되고 있다. 이렇게 찾아낸 패턴은 Bro[8] 네트워크 침입 탐지 시스템에서 이용할 수 있는 실제 signature 형태로 만들어 진다.



(그림 5) Autograph의 구조

마. SGS-20 프로젝트 내의 과다 트래픽 탐지 기법 (ETRI)

ETRI에서는 2003년 10Giga급 네트워크 침입 방지 시스템(SGS-10)을 제작하였고, 2004년에는 20Giga급 네트워크 침입 방지 시스템(SGS-20)을 제작하였다. 2004년도에 제작한 SGS-20 시스템은 (그림 6)과 같은 시스템으로, 그 내부에 signature를 기반으로 하는 침입 탐지 및 방지 시스템과 더불어 비정상 과다 트래픽을 탐지하는 시스템을 가지고 있다. 이 비정상 과다 트래픽 탐지 시스템은 분산 서비스 거부 공격 및 웜 공격을 탐지할 수 있으며 동시에 이를 차단하는 등의 대응이 가능하다.

SGS-20 시스템 내의 과다 트래픽 감지 기법에서 이용하는 알고리듬은 CPD(Change Point Detection) 방식이다. CPD 방식은 특정 random process의 변화를 쉽게 찾아 낼 수 있는 것으로 산업공학 및 경영 공학에서 많이 이용되는 것이다. 이 프로젝트에서는 CPD 내의 CUSUM(Cumulative Sum) 기법을 이용하였고, 동시에 SPC(Statistical Process Control) 기법에서 많이 이용되는 EWMA 기법도 같이 적용하였다.

먼저 이 시스템에서 탐지를 위하여 활용하는 파라미터는 매우 다양하다. 네트워크 트래픽을 플로(flow) 별로 정리한 후 이들의 BPS(Bandwidth Per Second)와 PPS(Packet Per Second)값을 정리하였으며, 이들을 방향성을 주어 구분하였다(ingoing vs outgoing). 그리고, TCP 세션 정보를 추가로 수집하여 탐지의 정확도를 높이려 하였다. 그럼 이 프로젝트에서 이용한 알고리듬에 대해서 알아보자. 이 프로젝트에서 제안한 CUSUM 테스트의 알고리듬

(그림 6) SGS-20 네트워크 침입 방지 시스템

은 (8), (9)와 같은 수식으로 표현된다.

$$C_i^+ = \max[0, x_i - (u + K) + C_{i-1}^+] \quad (8)$$

$$C_i^- = \max[0, (u + K) - x_i + C_{i-1}^-] \quad (9)$$

이 때 초기 값은 모두 0이라고 가정한다.

식 (8)은 upper CUSUM value를 나타내고, (9)는 lower CUSUM value를 나타낸다. 이 때 우리에게 필요한 것은 급격한 증가를 찾아내는 것이기 때문에 (8)을 이용하도록 한다. (8)에서 x 값은 시간 축에서 변화하는 BPS와 PPS의 값들을 의미하고, u 값은 타깃으로 잡은 목표 값을 의미한다. 이 테스트에서 말하는 목표 값이란, BPS와 PPS의 일반적인

값들을 말한다. 따라서, 이 값은 static하게 정해질 수도 있으며, 기타 다른 정상 트래픽 값들에 의하여 dynamic하게 정해질 수도 있다. 이 값을 크게 잡는 경우 변화가 크게 생기는 경우에만 탐지가 가능하여, 큰 변화를 잡고 false-positive가 낮아질 확률이 높아지지만, 적은 변화에 민감하게 대처하기 어렵다는 단점이 있다. 그러나 값을 작게 잡는 경우 너무 민감하게 반응할 확률이 높기 때문에 역시 주의하여야 한다. 따라서 이와 같은 가정을 바탕으로 u 값을 설정하면 (10), (11)과 같다.

$$u\ of\ BPS = (first\ BPS\ value\ of\ Flow(n)) + |\ Default\ Value - (first\ BPS\ value\ of\ Flow(n))\ |/2 \quad (10)$$

$$u\ of\ PPS = (first\ PPS\ value\ of\ Flow(n)) + |\ Default\ Value - (first\ PPS\ value\ of\ Flow(n))\ | \quad (11)$$

그리고, 이러한 변수들을 대입하여 얻어진 결과들을 바탕으로 upper CUSUM value를 얻게 된다. 이 값들을 시간 순으로 나열하여 계속 증가하는 값을 보인다면 이는 상대적으로 이전에 비하여 BPS나 PPS 값들이 증가하는 것임을 알 수 있다. 따라서 (8) 값들의 누적 값이 특정 threshold m 값보다 큰 경우 (즉, BPS나 PPS의 증가가 연속적인 m point 만큼 발생), 이를 의심스러운 플로로 간주하게 된다. 그리고 이어서 TCP 세션 정보들을 수집하여 TCP 연결이 실패하는 비율에 대한 성공하는 비율을 구하게 된다.

$$A = (TCP\ 연결\ 실패) / (TCP\ 연결\ 성공) \quad (12)$$

그리고 이러한 비율 값을 마찬가지로 CUSUM 식을 이용하여 변화를 측정하게 된다. 이 때 A 값이 계속 증가하는 패턴을 보이게 된다면 역시 의심스러운 플로로 간주하게 되고, 위의 BPS와 PPS의 의심스러운 플로와 일치하게 되면 이를 웜으로 간주하게 되는 것이다.

이 시스템은 트래픽 데이터 수집 및 관리와 같이 고성능이 필요한 부분은 대부분 하드웨어로 실제 구현하였고, 분석을 위한 부분을 소프트웨어로 구현하였다. 그리고 소프트웨어 부분 역시 성능을 고려하여 메모리 연산을 위주로 하였기 때문에 고성능에 적합하다는 장점이 있다.

Ⅳ. 결 론

본 문서에서는 현재 인터넷 보안에서 가장 큰 화제가 되고 있으며 동시에 가장 많은 문제를 일으키고 있는 인터넷 웜에 대해서 알아보았다. 그리고 인터넷 웜을 탐지하기 위한 다양한 방법에 대해서도 알아 보았다. 단순히 웜이 이미 퍼져 나간 후에 웜에 대한 signature를 생성하여 웜을 탐지하는 것은 이미 네트워크에 많은 피해가 가해진 후에 이루어 질 수 밖에 없기 때문에, 그 자체적으로 완전한 해결책이 될 수 없다. 따라서, 이러한 문제를 해결하기 위하여 웜으로 생각되는 것을 미리 찾아내서 이를 차단하는 연구가 필요하고 앞으로도 이에 대한 연구가 계속 이루어져야 할 것이다.

향후 연구로는 인터넷 웜을 탐지하기 위한 시스템들의 실제적인 구현 사례와 이들의 적용 결과에 대한 것을 조사할 것이다.

참 고 문 헌

[1] D.M. Kienzie and M.C. Elder, "Recent Worms: A Survey and Trends," *ACM WORMS'03*, Washington DC, USA, Oct. 2003.

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

[3] Snort, http://www.snort.org

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

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

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

[7] H.A. Kim and Karp Brad, "Autograph: Toward Automated, Distributed Worm Signature Detection," *13th USENIX Security Symposium*, Aug. 2004.

[8] Bro NIDS, http://www.bro-ids.org/