韓國 OA學會 論文誌
第7卷 第1號, 2002. 3.

2002-7-1-3-1

네트워크상의 중간 노드 탐지를 위한 효과적인 탐지 알고리즘

김 효 남*

An Efficient Algorithm for Detecting Stepping Stones

Hyo-Nam Kim*

요 약

최근 인터넷 네트워크 공격자들이 익명성을 얻기 위하여 가장 보편적으로 사용하는 방법이 이른바 징검다리를 이용한 공격이다. 징검다리 방법은 공격자가 그들 고유의 컴퓨터가 아닌 이미 사전에 계정을 획득하였거나 침입에 성공한 호스트를 통하여 공격을 하는 방법으로 그 공격의 근원지를 찾기가 무척 힘들다.

따라서 본 논문에서는 이 같은 공격들을 인터넷에 접근 가능한 사이트를 모니터링 함으로서 찾아낼 수 있는 효과적인 탐지 알고리즘을 개발하고자 한다. 본 알고리즘은 네트워크 플로우의 패킷 사이즈와 전송 타이밍을 이용하여 탐지하기 때문에 네트워크 플로우의 컨텐츠들이 암호화되어 있어도 적용이 가능하다. 또한 DoS공격에 사용될 수 있는 사이트들을 탐지해 내고 그 위험 순위를 매김으로서 DoS공격에 효과적인 대응책이 될 수 있는 탐지 알고리즘을 소개한다.

Abstract

One widely-used technique by which network attackers attain anonymity and complicate their apprehension is by employing stepping stones: they launch attacks not from their own computer but from intermediary hosts that they previously compromised. We develop an efficient algorithm for detecting stepping stones by monitoring a site's Internet access link. The algorithm is based on the distinctive characteristics(packet size, timing) of interactive traffic, and not on connection contents, and hence can be used to find stepping stones even when the traffic is encrypted. We evaluate the algorithm on large Internet access traces and find that it performs quite well. However, the success of the algorithm is tempered by the discovery that large sites have many users who routinely traverse stepping stones for a variety of legitimate reasons.

* 청강문화산업대학 컴퓨터소프트웨어과

논문접수 : 2001. 12. 26
심사완료 : 2002. 2. 27

Ⅰ. 서론

최근 들어 네트워크 공격자들을 탐지해내는 가장 중요한 문제는 얼마나 공격자들이 그들의 IP를 잘 숨기고 있는가가 탐지의 효과와 속도를 좌우하고 있다.

때문에 공격자가 익명성을 얻기 위해 가장 쉽고 많이 사용하는 방법이 바로 징검다리(stepping stone[1] :공격자가 자기 자신의 컴퓨터가 아닌 네트워크 상의 타 호스트를 중간 매개체로 이용하여 공격을 하는 방법) 기법을 이용한 공격이다.

일반적으로 공격자는 그들이 이용한 호스트들의 순서를 바꿔 다양한 경로의 공격을 시도함으로써 그 침입경로의 탐지 및 추적이 어렵다.

우리는 이와 같은 DoS공격에 징검다리로 이용되는 호스트들을 탐지해 냄으로써 공격에 사용될지 모를 의심스러운 활동을 감시하고 외부 호스트로의 접속을 검열하여 내부의 공격자를 탐지해 낼 수 있다.

그리고 현재까지 공격자가 compromised 호스트를 이용하여 보내오는 패킷을 탐지하기 위한 수많은 연구가 이루어져 왔다. 이러한 연구들은 두 가지 목적으로 이루어져 왔는데, 첫째는 compromised 호스트를 찾아내는 것이며 둘째는 공격자를 역추적 해낼 수 있는 가장 효과적인 방법을 찾아내는 것이다.

이러한 연구들은 공격자의 활동을 실시간으로 탐지할 수 있는 방법에 초점을 맞추었는데[8, 9, 10] 이는 데이터 스트림에 fingerprint를 남기고 후에 각각 다른 곳에서 이를 검출하여 비교함으로써 일부 가능하게 되었다.[11,12]

또 다른 연구는 host-based 접근 방법으로써 공격자가 거쳐간 호스트들의 로그를 이용하여 추적하는 방법이다[13]. 원격 호스트에 로깅 하는 동안 원천 호스트는 원격 호스트에게 사용자의 이름과 그 동안 거쳐온 경로정보를 제공하게 된다. 목적지 호스트는 이러한 정보를 가지고 로그인 하려고 하는 사용자의 인증작업을 하게된다. 만약 인증에 성공하면 로그인을 허락하고 그 외의 경우에는 로그인이 거부되며 그 정보(trace)들은 후에 관리자에 의해 사용되어진다. 매우 엄격한 관리환경 하에서는 이러한 방법이 유용하지만 이 또한 비밀 채널이나 그 밖의 트릭으로 깨어질 가능성이 있다.

지금까지의 연구들은 대부분 실제 네트워크 환경 하에서 검증되지 않았으며 그 구조에서 많은 취약점을 드러냈다. 한가지 예로 fingerprinting 방법은 얼마나 잘못된 매치가 발생되었는가가 명확하지 않으며 또 link-based 방법, evasion방법과 유사하여 그 차이점을 확실히 구별하기가 힘들다.

이처럼 다양한 네트워크 환경 하에서 패킷과 스트림을 추적하는 것은 이제 초기 단계라고 볼 수 있다. 이미 제안되어진 추적 방법들은 아주 엄격한 통제하의 네트워크 환경에서만 그 효과를 나타낸다. 때문에 우리는 열린 네트워크 환경인 인터넷에서도 성능을 나타낼 수 있는 탐지 시스템의 개발이 절실하다.

본 논문에서는 위에 언급한 것과 같이 공격자가 자신의 신분을 속이기 위해 중간경로로 사용되어지는 호스트의 탐지 알고리즘을 개발해 보고자 한다. 이에 이어지는 내용은 알고리즘에 사용되어지는 용어 및 기호들을 정의하고 실제 탐지에 사용되어지는 알고리즘을 기술하였으며, 그리고 알고리즘과 관련한 실험 및 논의 사항을 제시한다. 마지막으로 알고리즘에 대한 결론으로 논문을 마치고자한다.

Ⅱ. 본 론

2.1 탐지 알고리즘

2.1.1 용어 정의

먼저 본 논문에서 제시한 알고리즘의 이해를 돕기위해 알고리즘에 사용된 용어의 정의를 한다.

■ 징검다리 기법

징검다리 기법이란 공격자가 자신 고유의 IP를 속이고 이전에 접근하였던 머신(machine)을 통하여 타 호스트를 공격하는 방법이다.

■ 네트워크 주기 (network periods)

본 논문에서는 특별히 TCP연결상의 키스트로크(keystroke)의 지연시간을 기준으로 네트웍의 흐름이 있었는

가 아닌가를 일정시간을 기준으로 판단하여 나눈 주기이다.

■ 징검다리쌍

징검다리(stepping stone)로 연결되어 있는 두 사이트(양방향으로 공격의 기점이 될 수 있는)를 나타낸다.

2.1.2 ON/OFF 주기

본 논문에 제시한 알고리즘에서 사용된 ON/OFF 주기에 대한 정의는 다음과 같다.

일단 연결이 이루어진 후 Tidle 시간 동안 데이터의 흐름이 없을 때 우리는 이를 OFF 주기로 간주한다. 우리는 TCP연결에서 패킷에 실린 데이터가 새로운 것일 경우에만 데이터의 흐름이 있는 것으로 간주한다.

패킷에 새로운 데이터가 실린 경우 우리는 OFF 주기가 끝나고 ON 주기가 시작된 것으로 간주하고 Tidle 시간 동안 데이터의 흐름이 없을 때까지 그 상태를 지속한다.

본 정의에 사용된 ON/OFF 주기의 구분은 파레토그래프의 고정된 값으로 잘 알려진 TCP telnet연결에서의 사용자 키스트로크 시간의 간격분포[14 ,15]를 이용하였으며 이 논문에 따라 Tidle 시간을 0.5로 설정하였다.

2.1.3 알고리즘

1. 먼저 C1과C2를 징검다리쌍이라 했을 때 C1과 C2에서 매우 유사한 횟수의 OFF 주기가 나타남을 관찰할 수 있다. 이것은 사용자의 타이핑이 먼저 C1을 통해 전송되고 곧바로 C2를 통하여 전송되거나 혹은 또 다른 쉘 프롬프트를 받을 경우 이 쉘 프롬프트가 뜰 때까지의 지연시간이 마찬가지로 두 연결에서 유사하게 나타나는 것으로 알 수 있다.
2. C1의 OFF 주기의 수와 C2의 OFF 주기의 수, 그리고 C1과 C2에서 동시에 관찰된(둘 사이의 차이가 500msec 이하인) OFF 주기의 수를 체크하여 두 연결에서 동시에 관찰된 OFF 주기의 수를 두 연결 중 OFF 주기의 수가 작은 연결의 OFF 주기의 수로 나누어 전체 OFF 주기중 몇%가 일치하였는지를 계산한다. 이때 그 결과가 30%를 기준으로 하여 30%이상일 경우 일단 이 두 연결을 예비 징검다리쌍 영역으로 저장하고 지속적인 모니터링을 한다.
3. 만일 그 결과가 30% 미만일 경우에는 이 두 연결을 징검다리쌍이 아닌 것으로 간주 [그림 1]에서 나타내는 징검다리쌍 제외 영역으로 이동 저장한다. 이후 다시 이 두 연결이 30% 미만으로 계산되어지면 징검다리쌍 제외 영역에서 삭제시킨다.
4. 예비 징검다리쌍 영역에 저장되어 있는 연결들이 만일 또다시 그 OFF 주기의 일치율 계산결과 30%를 초과하게 되면 이 두 연결을 확정 징검다리쌍 영역으로 이동 저장하고 두 사이트에 대한 접속을 제한한 후 그 두 사이트를 침입의 중간경로로 이용될 수 있는 사이트로 간주 경고 메시지를 발생시킨다.
5. 우리가 본 알고리즘에서 사용한 지연 시간은 TCP 연결의 키스트로크 시간(keystroke time)을 기준으로 하였는데 이는 매우 다양한 분포를 나타내기 때문에[19,20] 본 논문에서는 실험에 의한 평균치인 Tidle=0.5초를 사용하였다.

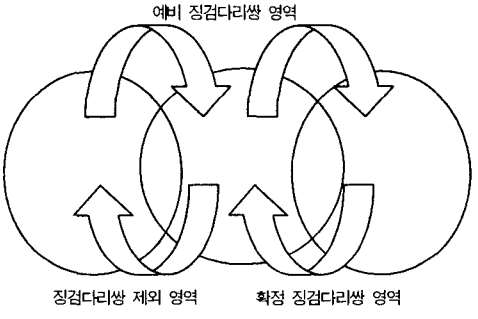

그림 1. 징검다리쌍 구분 영역

2.1.4 실험

현실적으로 국내에서 트레이스를 얻는다거나 연구실이나 학교에서 트레이스를 직접 얻을 수 있는 여건이 되지 않기 때문에 인터넷에 공개되어있는 University of California at Berkely(UCB)의 Telnet 트레이스를 이용하여 실험하였다.

* ucb-telnet.trace (390 MB. 7319 connection) - 이 트레이스를 바로 이용 하기는 곤란하므로 100MB를 떼어내어 10MB 씩 10번 실험하였다.
* 트레이스안의 모든 징검다리쌍들은 표시되어져 있다.

알고리즘에 대한 실험과정은 다음과 같다.

1. 트레이스 파일안의 모든 telent과 rlogin 출력을 추출한다.
2. 추출한 결과를 정렬한다.(sort ' uniq -c 명령 사용)
3. 정확하게 같은 패턴이 두 번 일어난 리스트를 체크

하여 나머지 리스트는 모두 삭제한다. 남은 리스트의 같은 패턴을 보이는 쌍들의 연결을 조사하여 연결이 일치하는 것은 제외시킨다.

4. 남겨진 리스트들에 대하여 알고리즘을 적용한다.

2.2.5 실험 결과

모두 100MB의 트레이스를 10MB씩 나누어 10번 실험한 결과를 아래 표와 같이 제시한다.

회수	1	2	3	4	5	6	7	8	9	10
오탐지	2	0	2	1	0	1	0	0	1	0
미탐지	1	4	3	0	3	5	1	2	0	4

그림 2. 실험결과(40%)

회수	1	2	3	4	5	6	7	8	9	10
오탐지	3	2	5	2	0	2	1	3	2	2
미탐지	0	0	0	0	0	0	0	0	0	0

그림 3. 실험결과(30%)

회수	1	2	3	4	5	6	7	8	9	10
오탐지	7	9	5	8	7	5	3	10	9	10
미탐지	0	0	0	0	0	0	0	0	0	0

그림 4. 실험결과(20%)

회수	1	2	3	4	5	6	7	8	9	10
오탐지	12	17	13	19	8	17	17	15	21	16
미탐지	0	0	0	0	0	0	0	0	0	0

그림 5. 실험결과(10%)

실험결과 30% 이상에서 징검다리쌍을 못 잡아낸 미탐지는 한 건도 없었으나 실제로 징검다리쌍이 아닌데도 탐지된 것이 다소 발견되었다.

결론적으로 30%에서 위 트레이스를 이용한 실험에서는 모든 징검다리쌍을 탐지해 내는 성능을 나타내었다.

2.2.6 평가 및 연구과제

실험결과만을 가지고 단정적으로 말한다면 본 논문에서 제시한 알고리즘은 완벽하게 그 역할을 수행하였다고 할 수 있다.

그러나 본 연구에 이용된 각종 통계적 수치들이나 파라미터들이 결코 확실히 검증된 것들이 아니고 실험방법 및 데이터들이 미흡했던 것이 사실이다.

또한 본 논문에서 인용되어진 많은 자료들이 실제로 실험에 의해 검증되었는지를 알 수 없기 때문에 본 논문에 언급된 알고리즘이 완벽하게 동작한다고는 단언할 수 없다.

마지막으로 본 논문에 제시된 알고리즘을 구현하는데 있어 그 알고리즘이 수행되는 시간이나 하드웨어적인 오버헤드를 전혀 감안하지 않았다. 이는 실제 실용단계의 구현에 있어 심각한 문제점으로 나타나리라 생각되기에 본 알고리즘을 응용한 탐지 시스템 등은 이 문제의 해결에 역점을 두어야 할 것이다.

Ⅲ. 결 론

최근 인터넷 공격의 추세는 자기의 신분을 숨긴 채 다른 사이트를 이용하여 공격하는 방법이 주류를 이루고 있다. 또한 이를 이용한 DoS공격은 그 공격의 다양성이나 용이성 등을 미루어 볼 때 아주 심각한 상황이다. 이처럼 자기의 신분을 숨긴 공격의 대표적인 것들이 바로 징검다리를 이용한 공격방법이다.

본 논문에서 우리는 이러한 징검다리 공격을 탐지할 수 있는 방법을 네트워크 플로우의 주기를 이용하여 실시간으로 탐지해 내고 또한 이를 지속적으로 관리 할 수 있는 알고리즘을 개발해보았다. 그리고 언급된 방법과 유사한 알고리즘이 Internet Relay Chat [16] 에서 사용되었으며 Distributed Denial-of-Service tools [17]에서도 유사한 방법이 언급되어있다.

본 논문은 아직 이론단계의 알고리즘을 제시한 것이기 때문에 논문에서 사용된 실험치 들과 각종 이론들이 실제로 본 논문에서 제시한 알고리즘과 부합하는지를 검증하는 단계가 필요하다. 또한 논문에서 사용된 컨트롤 파라미터들은 추정치가 아닌 실험에 의한 정확하고 통계적인 수치로 사용되어져야만 할 것이다. 그리고 본 논문의 목적인 징검다리쌍의 탐지에만 머무르지 않고 기존 에 개발되어진 침입자 역추적 시스템들을 본 논문의 알고리즘과

연관시켜 적절히 사용한다면 좀더 효과적인 역추적 시스템을 개발할 수 있으리라 기대된다.

따라서 본 연구의 최종 목표는 침입자 역추적 시스템이 될 것이다.

참고문헌

[1] ZHANG, Y., AND PAXSON, V. "Stepping Stone Detection". USENIX Security Symposium, Denver, Colorado, August 2000

[2] BELLOVIN, S. M. "Security Problems in TCP-IP Protocol Suite". Compu Communications Review 19, 2 (April 1 32-48.

[3] CA-98.01, C. A. IP Denial-of Service Attacks. http://www.cert.org/advisories/CA-98.01.smurf.html, January 1998.

[4] CA-98.13, C. A. Vulnerability in Certain TCP/IP Implementations.http://www.cert/org/advisories/CA-98-13-tcp-denial-of-service.html,December 1998.

[5] CA-96.21, C. A. TCP SYN Flooding and IP Spoofing Attacks. http://www.cert.org/advisories/CA-96.21.tcp_syn_flooding.html, September 1996

[6] ROWE, J. "Intrusion detection and isolation protocol: Automated response to attacks". Presentation st RAID'99, Sep 1999.

[7] NIPC, "Trinoo/Tribal Flood Net/tfn2k", http://www.nipc.gov/, 1999.

[8] MANSFIELD, G., OHTA, K., TAKEI, Y., KATO, N., AND NEMOTO, Y. "Towards Trapping Wily Intruders in the Large". In Proceedings of the Second Annual Workshop in Recent Advances in Intrusion Detection(RAID)(West Lafayette, IN, Sep. 1999)

[9] ZHANG, Y., AND PAXSON, V. "Stepping Stone Detection". Presentation at SIGCOMM'99, New Areas of Research, August 1999.

[10] CHANG, H., AND D.DREW. DoSTracker. This was a publically available PERL script that attempted to trace a denial-of-service attack through a series of Cisco routers. It was released into the public domain, but later withdrawn. Copies are still available on some websites., June 1997

[11] STANIFORD-CHEN, S., AND HERBERLEIN, L. "Holding Intruders Accountable on the Internet". In Proc. of the 1995 IEEE Symposium on Security and Privacy (Oakland, CA, MAY 1995), pp. 39-49

[12] STANIFORD-CHEN, S., G. "Distributed tracing of intruders". Master's thesis, University of California Davis, 1995.

[13] JUNG, H. T., KIM, H. L., SEO, Y. M., CHOE, G., MIN, S. L., KIM, C. S., AND KOH, K. Caller identification system in the internet environment. In UNIX Security Symposium IV Proceedings(1993), pp. 69-78

[14] P. DANZIG, S. JAMIN, R. CACERES, D. MITZEL, and D. STRIN, "An Empirical Workload Model for Driving Wide-area TCP/IP Network Simulations, "Internetworking: Research and Experience, 3(1),pp. 1-26, 1992.

[15] V. PAXSON and S. FLOYD, "Wide-Area Traffic: The Failure of Poisson Modelling," IEEE/ACM Transactions on Networking, 3(3),pp. 226-244, June 1995

[16] J. OIKARINEN and D. REED, "Internet Relay Chat Protocol", RFC1459,Network Information Center,DDN Network Information Center, May 1993.

[17] Computer Emergency Response Team,

"Denial-of-Service tools", CERT Advisory CA-99-17,Dec. 1999

저 자 소 개

김 효 남

1988년 홍익대학교 전자계산학과 (이공학사)

1990년 홍익대학교 전자계산학과 (이공석사)

1999년 홍익대학교 전자계산학과 (박사과정)

현재 청강문화산업대학 컴퓨터소프트웨어과 교수

관신분야 :

Programming Language, Object Oriented Programming, XML / EDI, XSLT