제1회 한국사이버테러정보전학회 춘계학술발표대회 (2004.5)

# 시스템 프로세스 구조에 기반을 둔 침입자 추적 메커니즘

강형우*, 김강산*, 홍순좌*
* 국가보안기술연구소

## 요 약

본 논문에서는 현재 네트워크 환경에서 Stepping Stones을 이용한 경유지 우회 공격에 대한 침입자 추적 메커니즘을 제안한다. 침입자는 피해시스템에서 공격자의 IP주소 노출을 피하기 위하여 피해시스템을 직접 공격하지 않고 Stepping stone을 이용하여 경유지 우회 공격을 수행한다. 우리는 이와 같은 경유지 우회 공격 발생 시 공격자의 IP주소의 추적을 목적으로 한다. 침입자 추적은 크게 두 가지 분류로 나뉘어 진다. 첫째는 IP Packet traceback, 둘째는 Connection traceback 이다. 본 논문에서는 Connection traceback에 공격을 다루며, 운영체제의 프로세스 구조를 이용하여 공격자 또는 Stepping stone(경유지)을 구분하여 침입자의 위치를 추적한다.

# A New Intruder Traceback Mechanism based on System Process Structure

Hyungwoo Kang*, Kangsan Kim*, Soonjwa Hong*
* National Security Research Institute

## ABSTRACT

In this paper, we describe a defense mechanism to cope with stepping stones attacks in high-speed networks. (Stepping stones : Attacker launches attacks not from their own computer but from intermediary hosts that they previously compromised.) We aim at tracing origin hacker system, which attack target system via stepping stones. There are two kind of traceback technology : IP packet traceback, or connection traceback. We are concerned with connection traceback in this paper. We propose a new host-based traceback. The purpose of this paper is that distinguish between origin hacker system and stepping stones by using process structure of OS(Operating System).

## 1. 서 론

최근 인터넷 사용자가 급증하면서 인터넷을 이용한 각종 해킹 및 사이버 범죄가 크게 증가되고 있으며 그와 동시에 침입 탐지 기술도 급속도로 발전하고 있다. 이와 같은 기술은 침입자가 피해시스템에 직접 연결해서 공격하는 공격에 대해서는 매우 유용하게 사용될 수 있다. 하지만 침입자가 경유지를 이용하는 경유지 우회 공격을 시도할 경우, 침입탐지시스템은 침입자의 근원지 정보를 알아내지 못하며 바로 전 단계 경유지의 정보만을 얻을 수 있다. 이런 이유로 경유지 우회 공격이 발생 시 공격 근원지를 추적하는 것은 상당히 어려운 일이다. 따라서 수동적인 대응 기법에서 나아가 적극적인 대응이 필요한데 이를 위해 공격자 및 공격자 시스템에 대한 실시간 추적과 즉각적인 공격 대응이 필요하다.

본 논문에서는 적극적인 공격 대응에 해당되는 공격자에 대한 실시간 추적이 가능한 침입자 역추적 메커니즘을 제안한다. 2장에서는 역추적의 정의 및 그 문제의 어려움에 대해서 살펴보고, 3장에서는 근원지 추적을 위한 기존의 연구를 살펴보며 4장에서는 시스템 운영체제(예, UNIX)의 구조를 이용하여 해킹의 근원지를 실시간으로 추적하는 새로운 추적 메커니즘을 제안한다. 마지막으로 5장에서는 결론을 내린다.

## 2. 역추적 정의 및 문제점

### 2.1 용어 정의

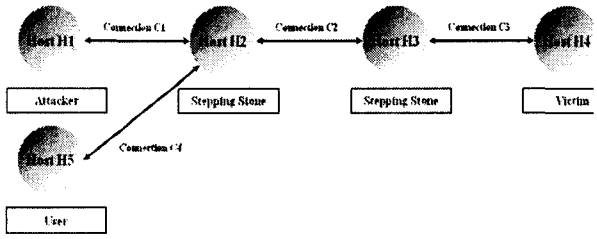
[정의 1] 역추적 : 사이버 범죄를 시도하는 공격자의 네트워크 상의실제 위치를 탐색하는 기술

[정의 2] Connection pair 과 stepping stone의 정의 : 어떤 사용자가 한 시스템에 접속하였다가 다른 시스템에 접속하였다면 접속한 sequence를 connection chain이라 한다. 이때 connection chain 상의 중간 호스트를 stepping stone이라 한다. 어떤 네트워크 connection pair가 connection chain의 일부이면 stepping stone connection pair라 한다.

가령 호스트 h1, h2, h3가 양방향 connection을 맺었다고 가정하고 h1 ↔ h2의 connection을 C1, h2 ↔ h3의 connection을 C2라 표현한다면 호스트 h2에서 C1과 C2는 stepping stone connection pair이다.

### 2.2 근원지 판단 문제 정의

에이전트 기반의 근원지 추적 문제를 정의하기 위해 <그림 1>을 살펴본다.




<그림 1> connection chain

호스트가 4대 있고 각각의 호스트를 H1, H2, H3, H4라 하고, 호스트 H1의 공격자 Attacker가 Host H4의 Victim 컴퓨터를 공격하기 위해서 먼저 Connection C1을 통해 호스트 H2에 접속한다. 그 후 다시 호스트 H3에 Connection C2를 통해 접속하고 다시 Connection C3를 통해 목표 호스트인 H4에 접속하여 공격을 수행하였다고 가정하자.

이 경우 Connection Chain은 [C1, C2, C3]이고, Stepping Stone은 H2와 H3가 된다. 이때 Stepping Stone Connection Pair는 [C1, C2]와

[C2, C3]이다.

만약 호스트 H3에서 호스트 H4로 공격한 connection C3를 탐지하고 [C1, C2, C3]와 같은 connection chain이 존재한다면, stepping stone인 호스트 H3에서 근원지 추적은 C3에 매칭되는 inbound connection C2를 찾는 문제로 정의할 수 있다.

마찬가지로 호스트 H2에서 호스트 H3로 공격한 connection C2를 알 수 있다면 stepping stone인 H2에서 근원지 추적은 C2에 매칭되는 inbound connection C1을 찾는 문제이다.

따라서 공격이 이루어진 connection Cn을 안다고 가정하면 호스트 Hn에서 근원지 추적은 Cn의 connection pair인 Cn-1을 찾는 것으로 정의 할 수 있다.

만약 호스트 Hn에서 Cn에 매칭하는 connection pair인 Cn-1이 존재한다면 이 호스트는 stepping stone이고 그렇지 않다면 이 호스트는 근원지로 판단할 수 있다.

### 2.3 Connection Pair 매칭

위의 <그림 1>은 정상적인 사용자 User와 공격자 Attacker가 Host H2에 접속한 모습을 표현한 그림이다. H2에서는 Connection C2에 해당하는 connection pair를 찾기 위해 inbound connection C1과 C4 중에서 적합한 connection을 판단해야한다.

본 논문에서는 H2의 시스템 프로세스의 성질을 이용하여 inbound connection인 C1과 C4 중에서 Attacker와의 연결인 C1을 찾아내는 추적 메커니즘을 제안한다.

## 3. 기존의 연구

지금까지 근원지 추적에 관한 연구를 살펴보면 다음과 같이 크게 세 가지로 나눌 수 있다.

- 호스트 기반 추적 (Host-based Traceback)
- 네트워크 기반 추적 (Network-based Traceback)
- 동적 네트워크 기반 추적 (Active Network based Traceback)

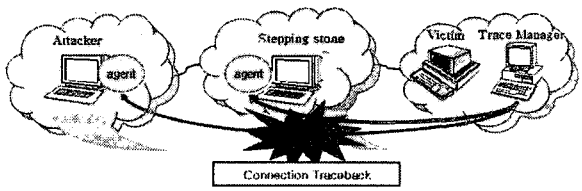
호스트 기반의 추적 메커니즘은 에이전트 기반 시스템으로 모든 호스트에 추적 모듈을 설치하거나 경유지로 사용된 호스트에 역추적 모듈을 설치하여 침입자를 추적한다. 이때 에이전트에서의 추적 목표는 해당 에이전트가 설치된 시스템에서 공격이 시작된 것인지 다른 시스템에서의 접속을 통해 중간 경유지로 사용되었는지 판단하는 것이다.

네트워크 기반의 추적 메커니즘은 네트워크를 구성하는 장치인 라우터나 방화벽에 추적 및 침입 탐지 모듈을 설치하여 네트워크 단위로 추적이 가능하도록 하는 방법이다.

Active Network 기반의 추적 메커니즘은 active 네트워크의 기능을 이용하여 추적 및 대응 모듈을 라우터 단위에서 수행할 수 있도록 하는 방법이다.

## 4. 제안하는 추적 메커니즘

에이전트 기반 실시간 근원지 추적 메커니즘은 다음 <그림 2>와 같은 경유지 우회 공격에 대해 근원지 추적을 하며 실시간 추적 및 사후 추적이 가능하다.




<그림 2> 경유지 우회 공격에 대한 추적

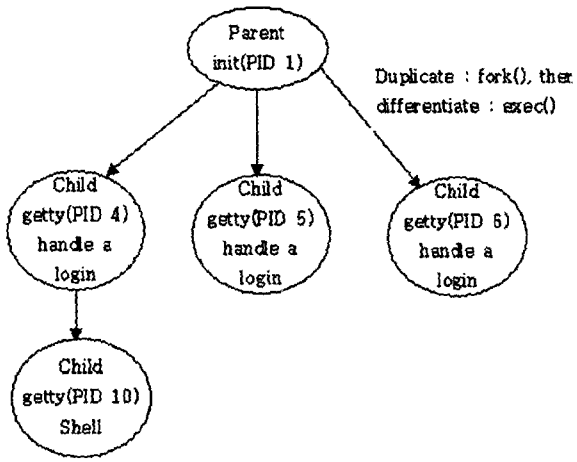
경유지 우회 공격에서 경유지로 사용된 stepping stone을 탐지하기 위해서는 해당 시스템에 에이전트를 설치하여 경유지/근원지를 식별한다. 만약 중간 경유지로 사용되었다면 inbound connection과 outbound connection이 존재할 것이며 두 개의 connection은 밀접한 상관관계를 가질 것이다. 본 논문에서 적용된 추적 메커니즘에서는 inbound connection과 outbound connection에 연결된 프로세스의 연관성을 통해 connection pair를 찾아낸다. 본 장에서는 UNIX 시스템 프로세스 구조의 특성을 이용하여 해당 시스템이 stepping stone인지 아니면 근원지 시스템인지를 판별하고 stepping stone일 경우 근원지를 추적하기 위하여 inbound connection system의 IP Number와 Port Number를 찾아내는 알고리즘을 제안한다.

### 4.1 UNIX 시스템 운영체제의 프로세스 구조

유닉스 시스템의 모든 프로세스는 아래의 요소를 가지고 있다.

- data
- code
- stack
- unique id number(PID)

유닉스 시스템이 처음 시작되면 "init"라는 한 개의 프로세스만이 생성되며, 이후의 모든 프로세스는 init 프로세스의 복사에 의해서 생성 된다. 즉, init 프로세스는 모든 프로세스의 ancestor 가 된다. 프로세스가 복사될 때 parent 프로세스와 child 프로세스는 PID를 제외하고는 동일하며 같은 code를 실행한다. 하지만 다른 실행코드를 사용하여 실행하는 code를 변경할 수 있다. 이와 같은 방식으로 유닉스의 프로세스는 생성되고 실행된다.

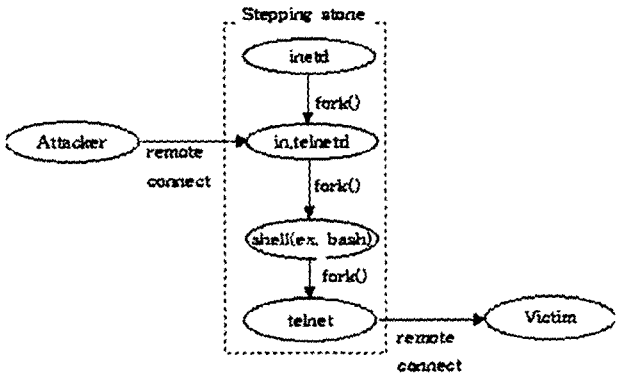



<그림 3> 유닉스시스템 프로세스 구조

### 4.2 connection chain이 발생 시 프로세스 구조

경유지가 유닉스 시스템인 환경에서 Connection pair(chain)가 발생되는 유형은 아래 3가지로 형태로 볼 수 있다.

#### 4.2.1 Telnet, SSH, rlogin 등 접속(접속 유형 1)



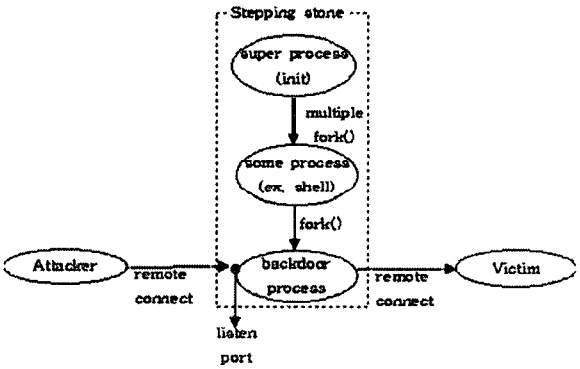

<그림 4> Connection-chain 접속유형 1

가장 일반적인 접속 유형으로서 경유지 시스템 네트워크 서비스 데몬인 inetd 데몬으로 telnet 접속하면in.telnetd 데몬을 통해 shell이 뜨

게 된다. telnet으로 통해 shell로 들어왔을 경우 작업을 하다가 다시 telnet 명령어를 이용하여 피해 시스템으로 접속하여 피해 시스템을 공격하는 유형이다.

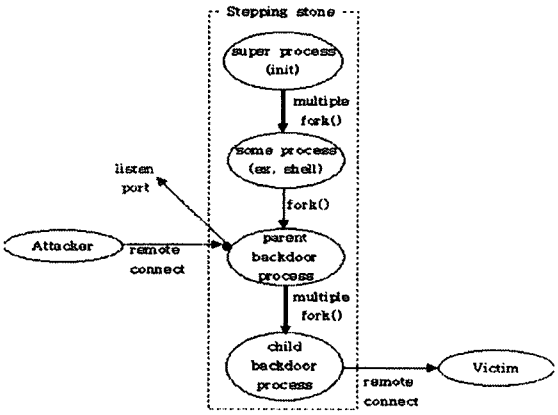
#### 4.2.2 직접 접속형 백도어(접속 유형 2)

설치된 백도어가 port를 열고 listen하고 있으며 근원지에서 원격 접속 시 프로세스 생성 없이 직접적으로 피해 시스템으로 접속하는 유형으로서 원격 DOS(Deny of Service) 등의 일부 백도어 형태의 공격이 이에 해당된다.




<그림 5> Connection-chain 접속유형 2

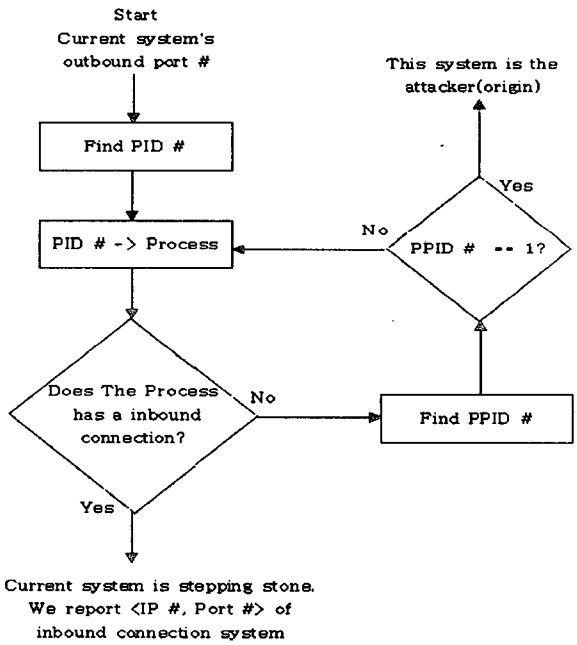
#### 4.2.3 프로세스 fork형 백도어(접속 유형 3)




<그림 6> Connection-chain 접속유형 3

위의 접속 유형 2와 유사하지만 Attacker에서 stepping stone의 백도어에 접속을 한 다음 바로 피해 시스템에 접속하지 않고, child 프로세스를 fork 한 후에 피해시스템에 접속하는 유형이다. 여기서 child 프로세스를 fork는 한번만 일어날 수도 있고 여러 번에 거쳐서 일어날 수 있다. port redirect 백도어나 기타 대부분의 백도어가 이와 같은 유형을 가진다.

### 4.3 stepping stone 찾는 알고리즘




<그림 7> Connection pair matching algorithm

공격에 추적의 입력 값에 해당되는 경유지의 source 포트가 주어졌을 때 그 포트와 연결되어 있는 프로세스 ID를 찾고 PID에 해당되는 프로세스를 찾아서 그 프로세스에 inbound connection이 있는지 확인한다. 만약 inbound connection이 존재할 경우 그 inbound

connection의 IP와 포트가 근원지의 정보가 된다. 만약 inbound connection이 존재하지 않을 경우 그 프로세스의 부모 프로세스의 ID를 찾게 된다. PPID 값이 1(super process)이면 현재 시스템이 근원지라고 판단하고 그렇지 않을 경우에는 다시 PPID에 해당되는 프로세스를 찾아서 inbound connection을 찾는 일을 반복한다.

### 4.4 제안된 방법의 장점

- 거의 100% 오류 없이 근원지 추적이 가능함
- 네트워크 단위의 추적 메커니즘의 경우 라우터나 방화벽에 모듈을 설치해야 하는데 이에 비해 개발 및 설치비용이 줄어듦
- 텔넷과 같은 대화형(interactive) 모드의 connection 뿐만이 아니라 거의 대부분의 connection 을 찾아냄
- 기존의 connection pair matching 기법[8]에 비하여 경유지의 양쪽의 데이터가 다르거나 한쪽이 암호화 되어 있어도 잘 찾아냄
- 기존의 connection pair matching 기법[8]에 비하여 TCP connection 뿐만 아니라 connectionless 의 프로토콜도 추적 가능함
- 시스템 프로세스를 이용하므로 기존의 호스트기반[4,8]의 추적보다 효율적임
- inbound 와 outbound connection의 프로세스 기록을 로그로 남기면 사후 추적 등 포렌식 도구로 사용가능

## 5. 결론

컴퓨터와 인터넷의 보급으로 인하여 우리 생활에 있어서 많은 혜택이 있음과 동시에 컴퓨터 해킹 및 바이러스와 같은 정보화에 대한 역기능이 발생하였다. 침입 발생 시 그것을 탐지하는 IDS는 지금까지 매우 유용하게 사용되어 왔지만 경유지를 이용하는 징검다리 공격에 대해서는 경유지의 정보만을 제공할 뿐 실제 근원지인 공격자의 정보를 얻어 올 수 없어서 시스템 해킹에 대한 대응책으로서 그 한계를 나타내고 있다. 이런 이유로 말미암아 해킹에 대한 능동적인 대응 방안의 일환으로 침입자의 근원지까지 추적하는 새로운 역추적 메커니즘을 개발하여 제안하였다. 4장에서 보인 새로운 역추적 메커니즘은 기존의 연구에 비교하여 4.4절과 같은 장점을 가지고 있다. 본 논문의 연구는 유닉스 시스템에서 새로운 침입자 추적 메커니즘 설계, connection pair matching 알고리즘을 설계하였으며, 향후 이와 같은 아이디어를 기반으로 윈도우 시스템에서도 적용 및 실험할 수 있도록 지속적인 연구가 필요하다.

## 참고문헌

[1] X. Wang, D. Reeves, S. F. Wu, and J. Yuill, "Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework", Proceedings of IFIP Conference on Security, Mar. 2001

[2] Buchholz, Thomas E. Daniels, Benjamin Kuperman, Clay Shields, "Packet Tracker Final Report," CERIAS Technical Report 2000-23, Purdue University, 2000.

[3] K. Yoda and H. Etoh, "Finding a Connection Chain for Tracing Intruders,"In F. Guppens, Y. Deswarte, D. Gollamann, and M. Waidner, editors, 6th European Symposisum on Research in Computer Security - ESORICS 2000 LNCS -1985, Toulouse, France, Oct. 2000.

[4] H.T. Jung et al. "Caller Identification System in the Internet Environment.," Proceedings of the 4th Usenix Security Symposium, 1993.

[5] Chaeho Lim, "Semi-Auto Intruder Retracing Using Autonomous Intrusion Analysis Agent,"FIRST Conference on Computer Security Incident Handling & Response 1999.

[6] Steven R. Snapp, James Brentano, Gihan V. Dias, "DIDS(Distributed Intrusion Detection System) Motivation, Architecture, and An Early Prototype," Proceedings of the 14th National Computer Security Conference, 1991.

[7] S. Staniford-Chen and L.T. Heberlein. "Holding Intruders Accountable on the Internet," In Proceedings of the 1995 IEEE Symposium on Security and Privacy, 1995.

[8] Y. Zhang and V. Paxson, "Detecting Stepping stones," Proceedings of 9th USENIX Security Symposium, Aug. 2000.

[9] D. Schnackenberg, K. Djahandari, and D. Sterene, "Infrastructure for Intrusion Detection and Response," Proceedings of DISCEX, Jan. 2000.

[10] D. Schnackenberg, K. Djahandary, and D Strene, "Cooperative Intrusion Traceback and Response Architecture(CITRA)," Proceedings of the 2nd DARPA Information Survivability Conference and Exposition(DISCEXII), June 2001.

[11] Dawn X. Song and Adrian Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," Proceedings of InfoCom 2001.

[12] Stefan Savage, David Wetherall, Anna Karlin "Practical Network Support for IP Traceback," Proceedings of the 2000 ACM SIGCOMM Conference, Stockholm, Sweden, Aug. 2000, pp295-306.

[13] Graham Glass, "UNIX for Programmers and Users : A Complete Guide", Prentice Hall, 1993.

[14] Dong-il Seo, "Trend & Technique of Intruder Traceback", ITU-T Workshop on Security, may 2002.

[15] Wright Stevens, "Unix Network Programming," Prentice Hall, 1998.

[16] W.R. Stevens. TCP/IP Illustrated, Vol.1, Addison Wesley, 1994.

강 형 우

1997년 고려대학교 전산학과(이학사)
1999년 고려대학교 전산학과(이학석사)
1999년 고려대학교 정보보호대학원 박사과정 재학 중
2000년 ~ 현재 국가보안기술연구소 선임연구원

김 강 산

2002년 경북대학교 전자공학과(공학사)
2004년 한국과학기술원 전자전산학과(공학석사)
2004년 ~ 현재 국가보안기술연 구소 연구원

홍 순 좌

1989년 숭실대학교 전산학과(공학사)
1991년 숭실대학교 전산학과(공학석사)
1991년~2000년 국방과학연구소 선임연구원
2000년~현재 국가보안기술연구소
선임연구원/팀장