제 19 회 한국정보처리학회 춘계학술발표대회 논문집 제 10 권 제 1 호 (2003. 5)

침입자 역추적을 위한 워터마크 패킷 생성 시스템 설계 및 구현

한승완 O, 서동일
한국전자통신연구원 사이버테러기술분석팀
e-mail : {hansw,bluesea}@etri.re.kr

Design and Implementation of Watermarked Packet Creation System for the Intruder Traceback

Seung-Wan Han O, Dong-Il Seo
Anti-CyberTerror Team, ETRI

요 약

역추적 기술은 해킹이 발생했을 때 해커의 실제적인 위치를 추적하는 기술이다. 해커의 위치를 추적하기 위해 현재 가장 널리 사용되는 방법은 시스템 관리자의 시스템 로그 분석과 시스템 관리자 사이의 상호 정보 교환을 통한 수동적인 역추적 방법이다. 그러나 수동적인 역추적 방법은 수작업으로 수행되는 시스템 로그 분석과 관리자의 개입으로 인하여 많은 비용이 요구되고 추적 시간의 지연이 발생하기 때문에 이러한 수동적인 방법으로는 해커의 위치를 실시간으로 추적할 수 없다. 만약, 네트워크의 특정 패킷을 식별 가능한 형태로 가공할 수 있다면, 특정 패킷의 이동 경로를 효과적으로 추적할 수 있어 공격자의 위치를 실시간 역추적하는데 활용될 수 있다. 본 논문에서는 네트워크의 특정 패킷에 의미적인 워터마크를 삽입하여 워터마크 패킷을 생성하는 워터마크 패킷 생성 시스템을 설계하고 구현한다. 이 시스템은 중간 경유지를 활용하여 우회 공격하는 공격자의 실제 위치를 실시간 역추적하는 ACT 역추적 기법의 구현에 활용되었다.

1. 서론

네트워크 인프라의 확충으로 인하여 누구나 쉽게 언제 어디서나 인터넷을 사용할 수 있게 되었다. 그리고 인터넷은 업무 처리의 신속성, 편리함, 효율성을 제공하는 도구로써 많은 사람들에게 각광을 받게 되었다. 반면에, 인터넷의 오용으로 인한 각종 침해 사고 역시 크게 증가하고 있다. 이러한 증가 추세는 해킹을 악의적인 크래커뿐만 아니라, 해킹에 호기심을 갖은 많은 스크립트 키즈들도 쉽게 해킹 도구를 사용하여 수행할 수 있기 때문으로 분석할 수 있다.

해킹으로부터 시스템을 보호하기 위해 현재 가장 널리 사용되는 대응책으로는 침입차단시스템(Firewall), 침입탐지시스템(Intrusion Detection System), 침입방지시스템(Intrusion Prevention System) 등과 같은 보안 솔루션들이 있다. 시스템 또는 네트워크에 보안 솔루션을 설치함으로써 많은 해킹을 방지하고 탐지할 수 있다. 그러나 이러한 보안 솔루션들을 우회하거나 동작을 방해하는 새로운 해킹 기법과 도구들이 계속적으로 출현하고 있으므로 기존의 보안 솔루션만을 가지고 해킹을 방어하기에는 많은 어려움이 있다. 그러므로 해킹을 미연에 방지할 수 있는 억지력을 갖는 기술적, 사회적, 윤리적, 법적인 장치 마련이 시급하다. 본 논문에서는 해킹을 미연에 방지할 수 있는 억지력을 제공하는 기술적 주제를 다루려고 한다.

만약 해킹이 발생되었을 때 해킹을 시도하는 해커의 실제적인 위치를 발견할 수 있는 기술이 있다면, 해커를 붙잡을 수 있을 뿐만 아니라, 이러한 기술은 호기심 많은 스크립트 키드들의 호기심을 억제하는데도 활용될 수 있을 것이다.

역추적 기술은 해킹이 발생했을 때 해커의 실제적인 위치를 추적하는 기술이다. 해커의 위치를 추적하기 위해 현재 가장 널리 사용되는 방법은 시스템 관리자의 시스템 로그 분석과 시스템 관리자 사이의 상호 정보 교환을 통한 수동적인 역추적 방법이다.

그러나 수동적인 역추적 방법은 수작업으로 수행되

는 시스템 로그 분석과 관리자의 개입으로 인하여 많은 비용이 요구되고 추적 시간의 지연이 발생하기 때문에 이러한 수동적인 방법으로는 해커의 위치를 실시간으로 추적할 수 없다. 그러므로 실시간에 해커의 실제 위치를 추적하기 위해서는 자동화되고 지능을 갖춘 보다 향상된 역추적 방법이 필요하다.

본 논문에서는 실시간 역추적에 활용될 수 있는 워터마크 패킷 생성 시스템을 제안하고자 한다. 워터마크 패킷 생성 시스템은 연결의 응답 패킷에 식별 가능한 의미적인 워터마크를 삽입하여 워터마크 패킷을 생성한다. 이렇게 생성된 워터마크 패킷은 해커의 위치를 역추적하는데 사용할 수 있다.

해킹은 크게 Dos/DDos 와 같이 패킷 수준에서 행해지는 해킹과 telnet 이나 rlogin 을 이용하는 연결을 갖는 해킹 등으로 나누어 볼 수 있다. 언급한 바와 같이, 본 논문에서 제안될 워터마크 패킷 생성 시스템은 앞에서 살펴 본 두 부류의 해킹 중에서 연결을 갖고 stepping stone 을 활용하여 우회적으로 공격하는 해킹을 역추적하는데 사용될 수 있다.

논문의 구성은 다음과 같다. 먼저 2 장에서는 침입자 역추적 기술에 대한 기존 연구들에 대해 살펴본다. 3 장은 본 논문에서 제안하는 워터마크 패킷 생성 시스템의 구조 및 동작 개요를 설명하고, 4 장에서는 시스템의 구현에 대해 서술한다. 마지막으로 5 장에서 결론을 맺는다.

2. 침입자 역추적 기술

침입자 역추적 기술은 수행되는 해킹의 종류에 따라 그림 1 과 같이 분류할 수 있다.

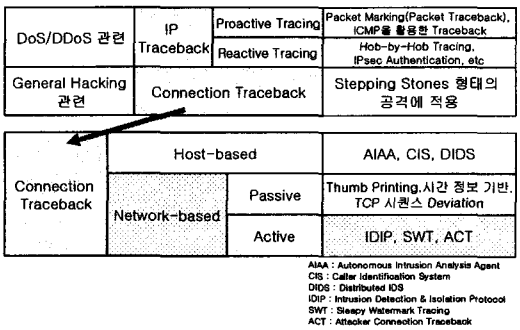

그림 1 침입자 역추적 기술의 분류

이 중에서 우회 경로를 통하여 수행되는 연결 기반 해킹에 대한 역추적 기술은 역추적이 수행되는 위치에 따라 다시 호스트 기반 역추적 기술과 네트워크 기반 역추적 기술로 나눌 수 있다.

먼저 호스트 기반 역추적 기술은 AIAA, CIS, DIDS 등이 있는데 각각에 대해 살펴보면 다음과 같다. CIS[1]는 H. T. Jung 에 의해 1993 년 제안된 시스템으로써, 사용자가 특정 시스템에 접속 하고자 할 때 해당 시스템은 접속을 시도하는 사용자가 그 이전에 거쳐왔던 모든 시스템에 대한 시스템 목록과 로그인 ID 등의 정보를 요구한다. 그리고 요구에 따라 이전의 경유 시스템 목록을 입력 받게 되면, 모든 경유 시스템과의 통신을 통해 각 시스템에 대해 입력된 시스템 및 로그인 ID 목록이 정당한 것인지를 확인하게 되고, 이러한 목록이 유효할 때만 접속을 허락한다. 이 방법은 사용자의 접속 지연, 인증을 위한 네트워크 부하, 인증 메시지의 무결성 등의 문제점을 갖는다. 또 다른 호스트 기반 역추적 기술인 AIAA 시스템[2]은 침해를 당한 서버의 해킹 피해 분석 및 로그 분석을 agent 를 이용해 자동화한 역추적 시스템이다. AIAA 시스템은 침입자가 거쳐온 경유 시스템의 관리자의 도움을 받아 AIAA 를 설치하고, 이 시스템에서 바로 이전의 침입경로와 해킹 흔적을 분석하고 다시 이전의 침입시스템으로 분석을 옮겨가서 최종 경유지 서버까지 거슬러 간다. 그러므로 이 시스템은 역추적에 많은 시간이 소요되고, 타 시스템 관리자와의 협조가 필수적인 단점이 있다.

앞에서 살펴본 호스트 기반 역추적 기술의 단점을 극복하기 위해 등장한 것이 네트워크 기반 역추적 기술이다. 네트워크 기반 역추적 기술은 침입자 역추적을 수행하기 위해 네트워크 트래픽 정보를 분석하고 이를 활용한다. 널리 알려진 네트워크 기반 역추적 기술에는 Thumb Printing 기법[3], 시간 정보 기반 기법[4], TCP 시퀀스 Deviation 기법[5] 등이 있다. Thumb Printing 기법은 피해 시스템으로부터 침입자가 위치하고 있는 시스템까지의 연결 사슬(Connection Chain)를 통해 송수신되는 데이터는 동일할 것이라는 아이디어를 이용한다. 먼저 침입자의 공격에 사용된 연결(connection)의 송수신 데이터에 대한 해쉬값인 thumbprints 을 계산한다. 그리고 네트워크 상에 존재하는 모든 연결의 송수신 데이터에 대한 해쉬값인 thumbprints 을 계산하여, thumbprints 값이 일정 수준 이상 동일한 연결들을 이용하여 하나의 연결 사슬을 구성하고 이를 통하여 해커의 위치를 추적한다. 이 방법은 네트워크 상의 모든 연결에 대해 thumbprints 을 계산하고 그 결과값을 저장해야 하는 단점이 있다. 또한 같은 연결 사슬에 속하지 않는 연결의 송수신 데이터의 thumbprints 가 우연히 동일하게 되는 경우, 같은 연결 사슬로 잘못 분류될 수 있다.

네트워크 트래픽의 내용 분석에 기반한 Thumb Printing 기법에서 나타난 문제점을 극복하기 위해 네트워크 트래픽의 다른 특성, 즉 시간이나 패킷의 시퀀스 번호 등을 이용하는 방법들이 제안되었다. 시간 정보 기반 역추적 기법은 해커가 입력하는 keystroke 에 의해 발생하는 데이터의 송신 간격은 프로그램에 의해 송신되는 데이터의 송신 간격에 비해 매우 크기 때문에 이를 쉽게 파악할 수 있고, 만약 같은 연결 사슬에 속한다면 그 간격이 매우 유사할 것이라는 점을 이용한다. 이 기법에서는 ON Period 와 OFF Period 를 이용하여 각각의 상태가 변화하는 시점과 한 상태를 유지하는 시간 간격을 분석하여 같은 연결 사슬에 속하는지 여부를 판단하게 된다. 네트워크 트래픽의 또 다른 특성인 패킷의 TCP 시퀀스 번호의 증가 정도를 이용하는 기법은 비록 송수신되는 데이터가 암호화되더라도 데이터의 양은 크게 변하지 않는다는 점에

착안하여 시퀀스 번호의 증가 정도를 변동 폭의 조정을 통해 비교하여 연결 사슬을 구성하는 알고리즘이다.

연결 기반 해킹에 대한 다른 부류의 역추적 기술인 SWT 와, ACT 는 침입자의 연결에 대한 응답 패킷에 표식자를 삽입하여 역추적을 수행하는 기법이다. 이러한 기법은 비교적 간단하고 효율적으로 역추적을 수행할 수 있는 장점을 갖는다. 특히 ACT 는 본 논문에서 제안된 워터마크 패킷 생성 시스템을 사용하여 현재 인터넷 환경에 적용할 수 있도록 구현된 역추적 시스템이다.

3. 워터마크 패킷 생성 시스템

워터마크 패킷 생성 시스템은 네트워크의 특정 패킷에 의미적인 워터마크를 삽입하여 네트워크에서 그 패킷을 식별 가능하게 하는 시스템이다. 시스템의 구성은 그림 2 와 같다.

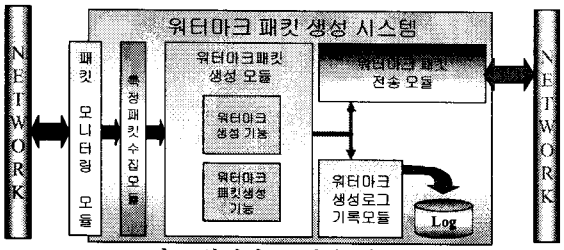

그림 2 워터마크 패킷 생성 시스템

워터마크 패킷 생성 시스템을 구성하고 있는 각각의 모듈의 기능을 살펴보면 다음과 같다.

- 패킷 모니터링 모듈
  - 네트워크 상에 송수신되는 패킷 관찰
- 특정 패킷 수집 모듈
  - 모니터링 패킷 중에서 사용자에 의해 지정된 패킷 수집
- 워터마크 패킷 생성 모듈
  - 수집 패킷에 대한 워터마크 생성
  - 생성된 워터마크를 패킷에 삽입하여 워터마크 패킷 생성
- 워터마크 패킷 전송 모듈
  - 워터마크 패킷를 시스템의 네트워크 인터페이스 카드(NIC)를 통하여 목적지로 전송
- 워터마크 생성 로그 기록 모듈
  - 수집 패킷과 생성된 워터마크에 대한 로그 기록

워터마크 패킷 생성 시스템에서는 특정 패킷을 식별 가능하도록 만들기 위해 의미적인 워터마크 정보를 사용한다. 여기에서 사용되는 워터마크 정보는 직관적으로 식별 불가능한 디지털 워터마크와는 달리 패킷 캡처를 통한 분석에서 쉽게 식별되는 정보다. 그러나 응용 프로그램의 사용자인 수신자에게는 보이지 않는 의미적인 워터마크 정보이다. 패킷 워터마크 생성 시스템에서 사용하는 의미적인 워터마크는 그림 3 과 같이 워터마크 판단 신호, 워터마크 생성 호스트의 주소, 워터마크 식별자(ID), 그리고 여러 개의 백스페이스 문자로 구성된다. 여기에서 백스페이스 문자는 수신되는 곳에서 워터마크 정보를 지우도록 함으로써 수신자에게 워터마크 정보가 들어나지 않도록 하는 역할을 한다.

그림 3 워터마크 정보

워터마크 패킷 생성 시스템의 핵심 모듈은 수집된 패킷에 의미적인 워터마크를 삽입하여 워터마크 팻킷을 만드는 워터마크 패킷 생성 모듈이다. 워터마크 패킷 생성 모듈은 패킷이 네트워크의 여러 노드를 경유하더라도 패킷이 식별 가능하도록 워터마크 정보를 그림 4 와 같이 패킷의 TCP 데이터 부분(payload)에 포함시켜 워터마크 패킷을 생성한다.

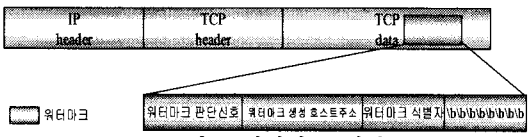

그림 4 워터마크 패킷

워터마크 패킷 생성 모듈에 의해 생성된 워터마크 패킷은 워터마크 패킷 전송 모듈에 의해 목적지로 전송된다. 그리고 워터마크 정보와 관련된 정보 및 전송 결과는 워터마크 생성 로그 기록 모듈에 의해 로그 저장소에 기록된다.

워터마크 생성 로그 기록 모듈에 의해 저장되는 로그 정보는 다음과 같다.

- 대상 패킷 정보
- 생성된 워터마크 정보
- 워터마크 패킷의 송신 성공 여부
- 각종 에러 로그

이상에서 살펴본 워터마크 패킷 생성 시스템의 동작 과정은 다음과 같다.

1) 패킷 수집 모듈은 모니터링된 패킷들로부터 조건에 맞는 특정 패킷을 수집한다.
2) 패킷 수집 모듈이 특정 패킷을 수집한 후, 워터마크 패킷 생성 모듈은 특정 패킷에 대한 워터마크를 생성한다.
3) 워터마크를 생성한 후, 워터마크 패킷 생성 모듈은 특정 패킷의 TCP 데이터 부분(payload)에 워터마크를 삽입하여 워터마크 패킷을 생성한다.
4) 워터마크 삽입이 완료되면, 워터마크 패킷 전송 모듈은 워터마크 패킷을 패킷의 목적지로 전송한다.
5) 마지막으로 워터마크 생성 블록은 워터마크

와 관련된 로그를 작성하여 로그 저장소에 저장한다.

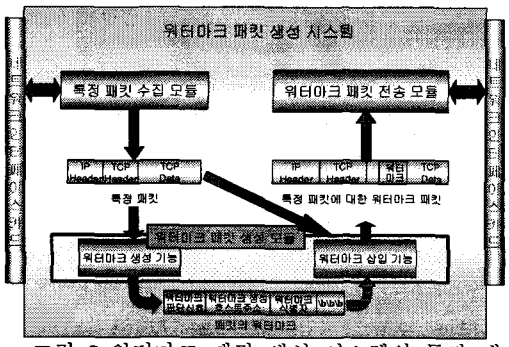

그림 5 워터마크 패킷 생성 시스템의 동작 개요

워터마크 패킷 생성 시스템을 사용하여 해커에 의해 공격을 입은 피해 시스템의 응답 패킷에 대한 패킷 워터마크를 생성한 후 삽입하여, 그 워터마크 패킷에 대한 이동 경로 추적을 추적함으로써 해커의 위치를 쉽게 역추적할 수 있다. 이때 피해 시스템의 응답 패킷에 대한 패킷 워터마크로 워터마크 식별자, 워터마크를 생성한 위치 정보, 해커의 공격 정보 등을 포함하면, 응답 패킷을 생성한 근원지 식별을 좀 더 쉽게 할 수 한다. 또한 패킷 워터마크를 패킷의 헤더가 아닌 패킷의 TCP 데이터 부분(payload)에 삽입하는 워터마크 패킷 생성 시스템의 특징은 응답 패킷의 워터마크가 여러 단계의 연결을 거치더라도 효력을 갖게 한다. 그 결과 여러 중간 호스트를 경유하는 해킹 시도 시에도 해커의 실제적인 위치에 대한 역추적을 가능하게 한다.

4. 구현

워터마크 패킷 생성 시스템은 리눅스에서 멀티쓰레드를 사용하여 구현되었다. 그림 6 은 워터마크 패킷 생성 시스템을 구성하는 멀티쓰레드와 할당된 메모리 사이의 상관 관계를 나타낸다.

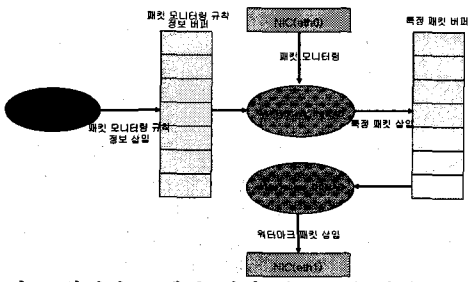

그림 6 워터마크 패킷 생성 시스템의 멀티쓰레드와 메모리의 상관 관계

워터마크 패킷 생성 시스템은 네트워크의 패킷을 실시간으로 모니터링하기 위해 Libpcap 라이브러리를 사용하는데, 그림 6 은 워터마크 패킷 생성 시스템에서 Libpcap 라이브러리를 이용하여 패킷 모니터링을 수행하는 코드의 일부분을 나타낸다.

그림 6 워 Libpcap 라이브러리를 이용한 패킷 모니터링

또한 워터마크 패킷 생성 시스템에서는 수집된 패킷에 워터마크를 삽입하여 생성된 워터마크 패킷을 목적지로 송신하기 위해 Libnet 라이브러리를 사용하였다. Libnet 라이브러리는 워터마크 패킷 전송 모듈 내에서 이용되고, 다음과 같은 6 단계를 거쳐 하나의 워터마크 패킷을 전송한다.

[워터마크 패킷 전송 단계]

단계 1: 네트워크 링크 초기화
단계 2: 패킷 삽입을 위한 메모리 할당
단계 3: 패킷 생성
- Ethernet Frame Header 생성
- IP Header 생성
- TCP Header 및 Payload 생성

단계 4: 패킷 체크섬 계산 (IP Header 및 TCP Header)
단계 5: 네트워크에 패킷 주입
단계 6: 메모리 free 및 네트워크 링크 close

5. 결론

본 논문에서는 네트워크의 특정 패킷을 식별 가능하게 만드는 워터마크 패킷 생성 시스템을 설계하고 구현하였다. 이 시스템은 우회 경로를 가지고 특정 호스트를 공격하는 공격자의 실제 위치를 실시간 역추적하는 기법에 활용되었다.

참고문헌

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

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

[3] 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.

[4] Y. Zhang and V. Paxson, "Detecting Stepping Stones", Proceedings of 9th USENIX Security Symposium, August 2000.

[5] 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.