인터넷 패킷 워터마크 검출 시스템 구현

최병철[0] 서동일
한국전자통신연구원 사이버테러기술분석팀
{corea[0], bluesea}@etri.re.kr

Anti-CyberTerror Team, ETRI

Byeong-Cheol Choi[0] Dong-Il Seo
Anti-CyberTerror Team, ETRI

요 약

본 연구에서는 TCP Connection을 유지하는 Stepping Stone 형태의 공격을 역추적하기 위해 사용되는 패킷 워터마크를 검출하는 시스템의 구현에 대해서 기술하고 있다. 본 연구에서 사용한 패킷 워터마크는 Sleepy Watermark Tracing(SWT)에서 사용한 virtual null string 형태의 워터마크를 사용하였으며, 이러한 워터마크의 효율적인 탐지 및 분석하는 방법을 설계 구현하였다. 본 연구의 의의는 패킷 워터마크 검출 시스템의 실제 구현을 통하여 TCP Connection Traceback 형태의 침입자 역추적 시스템에 활용할 수 있다는 것이다.

1. 서 론

최근에 침입자 역추적 기술은 분산 서비스 거부 공격(DDoS) 형태의 스푸핑된 IP를 추적하기 위한 IP Traceback 기술과 stepping stone 형태의 여러 시스템을 경유하는 공격 형태의 실제 TCP Connection을 추적하기 위한 TCP Connection Traceback 기술로 분류가 된다. 본 연구에서 사용된 패킷 워터마크 기법은 TCP Connection Traceback 시스템에서 사용되는 기법 중 SWT(Sleepy Watermark Tracing)에서 사용된 virtual null string 형태를 사용하였으며, 워터마크 전체 구조는 본 연구의 목적에 맞추어 새롭게 구성하였다.[1, 2, 3]

본 연구에서 언급하고 있는 패킷 워터마킹 기술은 디지털 워터마킹 기술과는 차이가 있으며, 워터마크란 원래의 개념적인 응용이다. 패킷 워터마킹 기술은 SWT 논문에서 사용되었으며, 본 논문에서 사용한 워터마크도 virtual null string 형태를 사용하였다. 본 연구에서 언급하고 있는 패킷 워터마킹 기술은 패킷 마킹 기술과는 차이가 있다. 둘 다 역추적 시스템 개발을 위해서 나온 기술이지만, 어떠한 역추적 시스템에 적용할 것인가는 다르다. 패킷 마킹 기술은 DoS나 DDoS와 같이 IP를 속이며 공격을 하는 형태의 공격의 변조되지 않은 IP를 추적하는 기술에 사용되며, 패킷 워터마킹 기술은 응답 패킷(reply packet) 에 해커의 시스템에서 보이지 않도록 패킷의 데이터 영역에 일종의 control 문자나 null string을 사용하여 stepping stone 형태의 connection을 유지하는 공격에 대한 역추적을 위해서 사용되는 것이다.

본 연구에서 개발한 인터넷 패킷 워터마크 검출 시스템은 현재망에서의 침입자 역추적 시스템에 사용될 수 있다. 실험 및 결과에서 본 연구와 관련하여 개발된 침입자 역추적 시스템에 적용하여 워터마크의 검출 성공 상황에 대한 실험 결과를 보여준다.

2. 패킷 마킹 & 패킷 워터마킹

1) 패킷 마킹 기술

패킷 마킹 기술은 DoS 또는 DDoS 형태의 IP가 속여진 형태의 공격에 대하여 공격자의 IP를 역추적 하기 위한 요소 기술이다. 즉, IP Traceback에 사용되는 기술이다. 그림 1은 IP Traceback 형태의 기술 중에서의 하나의 예로서, "Advanced and Authenticated Marking Schemes for IP Traceback"에서 사용한 IP Marking 기법에 대한 그림이다.[11]

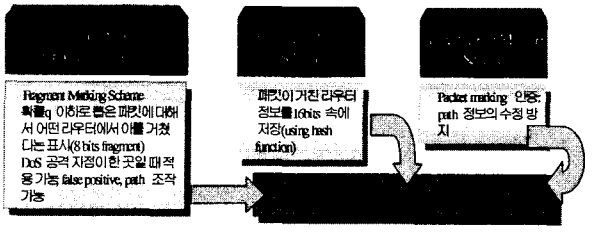

그림 1. IP Marking Scheme

2) 패킷 워터마킹 기술

패킷 워터마킹 기술은 stepping stone 형태의 공격자

가 여러 시스템을 경유하여 자신의 시스템 IP를 공개하지 않으려는 목적의 공격을 역추적에 사용되는 요소 기술이다. 즉, TCP Connection Traceback에 사용되는 기술이다. 그림 2는 SWT(Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Frame -work)에서 사용한 패킷 워터마킹 기법에 대한 것이다.[12, 31]

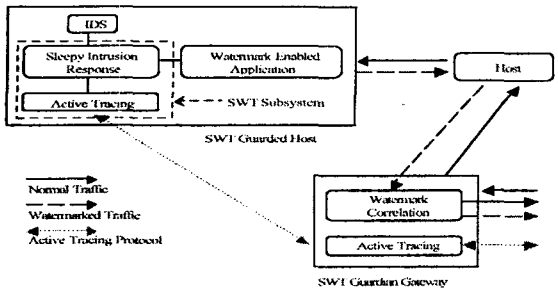

그림 2. Sleepy Watermark Tracing (SWT)

3. 워터마크 검출 시스템 구현

본 연구에서 사용한 패킷 워터마크는 Sleepy Watermark Tracing (SWT)에서 사용한 virtual null string 형태를 사용하였다. 실제 패킷에 어떠한 워터마크를 삽입했는지의 자세한 설명은 그림 3에 표시하였다.

워터마크의 정보는 다음과 같으며, 패킷에 워터마크가 있는지 여부를 판단하기 위한 워터마크 판단신호, 최초 패킷에 워터마크를 삽입한 시스템의 IP 주소인 워터마크 생성 IP 주소, 그리고 최초 생성 패킷의 데이터 영역의 신호를 ID화한 워터마크 ID로 구성되어 있다.

- 워터마크 판단 신호
- 워터마크 생성 IP 주소
- 워터마크 ID

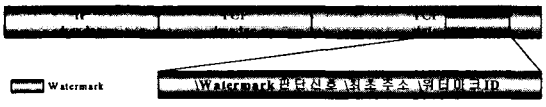

그림 3. 패킷 워터마크 구조

워터마크 검출 시스템 (WES: Watermark Extraction System)의 전체구조는 그림4와 같다. 네트워크 상의 패킷을 모니터링하는 패킷 모니터링 모듈(PMM)과 워터마크 유무를 판단하는 패킷 필터링 모듈(PFM), 패킷에 워터마크 정보가 존재할 때 워터마크 정보를 추출하고 signature 형태로 변경하는 워터마크 추출 모듈(WEM), WEM에서 추출된 signature를 최초 패킷에 워터마크를 삽입한 WES로 전송을 위한 워터마크 결과 전송 모듈(WRM), 그리고 워터마크 정보 및 signature 정보를 로그로 저장하도록 하는 워터마크 로그기록 모듈(WLM)로 구성되어 있다.

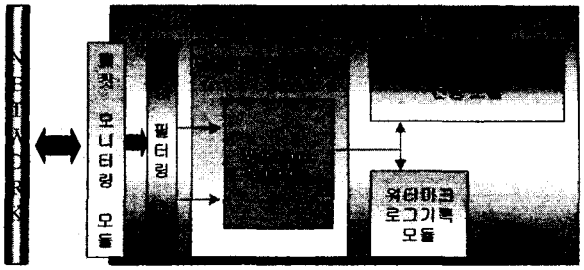

그림 4. 패킷 워터마크 검출 시스템 구조

워터마크 검출 시스템 (WES)의 동작은 지속적인 packet monitoring을 수행하면서 모든 incoming connection, outgoing connection 의 검사를 수행한다. 이때 패킷의 손실을 방지하기 위해서 네트워크 부하에 적절한 버퍼링을 하면서 패킷의 워터마크 존재 유무를 판단한다. 만약 워터마크가 존재하면 패킷의 IP 정보와 Watermark 정보로부터 최초의 워터마크 삽입 시스템의 주소를 판단하고 Signature를 전송하도록 구현하였다.

4. 실험 및 결과

본 연구의 실험에서 사용한 라이버러리 및 워터마크의 구조는 다음과 같다.

- 패킷 캡쳐 라이버러리 : libpcap을 사용하고, eth0에서 패킷을 수집 (스위칭 허브는 포터미러링된 상태임)
- 사용된 워터마크 : WM+4바이트(unsigned long 형태의 IP)+4바이트(unsigned long 형태의 ID)

워터마크 검출 시스템은 다음과 같은 동작 시나리오를 고려하여야 한다.

(1)워터마크 검출 시스템은 패킷 모니터링 모듈을 통해 지속적으로 네트워크 상에 송수신되는 패킷을 감시한다.

(2)워터마크 추출 모듈은 모니터링된 패킷에 워터마크가 삽입되어 있는지를 확인하고, 워터마크가 삽입되어 있으면 해당 워터마크를 추출하여 워터마크에 포함되어 있는 패킷의 정보를 수집한다.

(3)워터마크 추출 결과 전송 모듈은 추출된 워터마크 및 패킷 정보로부터 최초 워터마크를 패킷에 삽입한 시스템의 IP를 확인하고, 최초 IP로 워

터마크 및 패킷 정보를 전송한다.

(4) 워터마크 추출 결과를 로그로 저장한다.
- 추출된 워터마크
- 워터마크로부터 추출한 정보
- 워터마크가 삽입된 패킷 관련 정보
- 워터마크가 삽입된 패킷의 송신자 및 수신자 정보
- 워터마크가 삽입된 패킷의 확인 시간
- 각종 에러 로그

그림 5는 워터마크 검출 실험을 위한 테스트베드 구축에 관한 것이다. Linux(system 1)에서 다른 네트워크에 존재하는 Solaris(system 2)로 워터마크가 삽입된 패킷을 전송하였을 때, 미러링된 포트로 WES가 네트워크의 모든 패킷을 감시하고 있다가, 워터마크가 삽입된 패킷을 검출하고 관련된 정보를 저장하도록 구성하였다.

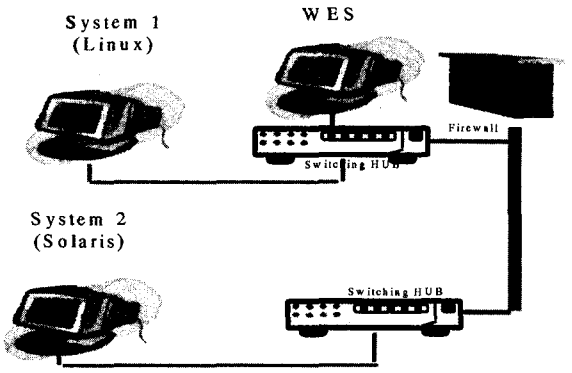

그림 5. 테스트베드 구성

그림 6은 그림 5에서 구성한 테스트베드를 기반으로 하여 워터마크된 패킷의 검출을 실험한 것이다. 본 연구에서 개발한 WES를 스위칭 허버의 미러링된 포터에 연결된 시스템에서 구동하고, Linux(system 1)에서 다른 네트워크에 존재하는 Solaris(system 2)로 워터마크된 패킷을 전송하고 실시간으로 패킷을 검출하였다. 본 실험에서는 실제 TCP Connection Traceback 기법을 사용한 현재 개발중인 침입자 역추적 시스템을 이용하여 실험하였다.

5. 결론

본 연구는 인터넷 패킷 워터마크 검출 시스템의 구현에 대한 것이다. 구현된 워터마크 검출 시스템은 TCP Connection Traceback 기법을 사용하는 침입자 역추적 시스템에 적용될 수 있다.

실험에서는 실제 TCP Connection Traceback 시스템으로 개발 중인 침입자 역추적 시스템에 적용하여 실험을 하였다. 사용된 워터마크도 실제 본 연구에 맞게 새롭게 구성된 것이다.

앞으로의 과제는 암호화된 TCP Connection을 고려한 새로운 워터마크 검출 시스템을 구현하는 것이다.

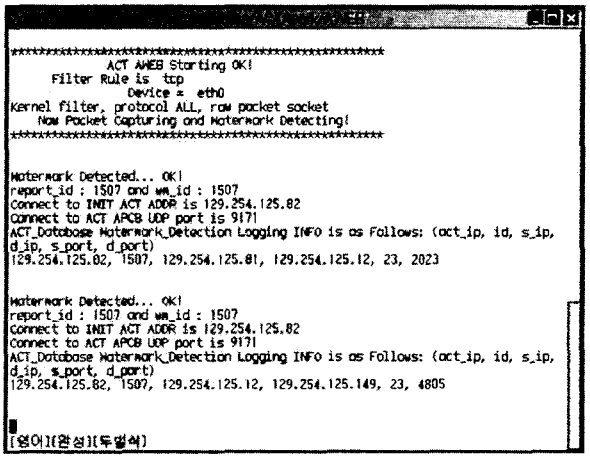

그림 6. 워터마크 검출 실험 - 침입자 역추적 시스템에 적용

참고문헌

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

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

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

[4] W. R. Stevens, "Unix Network Programming - Networking APIs", 1998 Prentice-Hall PTR

[5] W.R Stevens, "Advanced Programming in the Unix Environment", 1997 Addison-Wesley

[6] S. Northcutt and J. Novak, "Network Intrusion Detection an Analyst's Handbook", 2001 New Riders