세 가지 관점에서의 네트워크 트레이스 기반 및 실행 트레이스 기반 프로토콜 리버스 엔지니어링 차이점 분석

구영훈, Baraka D. Sija, 이성호, 김명섭
고려대학교
{gyh0808, sijabarakajia25, gaek5, tmskim}@korea.ac.kr

Analyzing the Differences Between Network Trace-based and Execution Trace-based Protocol Reverse Engineering in Three Perspectives

Young-Hoon Goo, Baraka D. Sija, Sung-Ho Lee, Myung-Sup Kim
Korea Univ.

요 약

급진적으로 발전하는 오늘날의 인터넷 환경 하에 복잡하고 다양한 비공개 프로토콜이 발생하고 있다. 효율적인 네트워크 관리 및 보안을 위해서 비공개 프로토콜에 대한 깊이 있는 이해가 필요하며 이에 대한 분석은 프로토콜 리버스 엔지니어링이라는 학문으로 꾸준히 연구되어왔다. 프로토콜 리버스 엔지니어링은 알려지지 않은 프로토콜의 사양을 추출하는 것으로 입력에 따라 크게 네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법으로 나눌 수 있다. 기존의 많은 선행 연구에서 각 분석 방법으로 저마다의 알고리즘을 구현하였지만, 아직까지 표준화된 방법론은 없으며 사용하기에 각각의 장단점이 존재한다. 이에 본 논문에서는 네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법의 차이점을 크게 일반적, 메커니즘적, 확장성 3가지 관점에서 상세하게 분석한다.

Ⅰ. 서론

오늘날의 인터넷은 전송 속도 증가와 대용량화로 인해 다양한 응용 및 악성행위가 출현하고 있으며 이에 따라 복잡 다양한 비공개 프로토콜이 발생하고 있다. 이러한 환경 하에 효율적인 네트워크 운용 및 관리와 보안을 위해서는 비공개 프로토콜에 대한 깊이 있는 이해가 필요하다. 이러한 비공개 프로토콜에 대한 구조 분석은 프로토콜 리버스 엔지니어링이라는 학문으로 꾸준히 연구되어왔다. 프로토콜 리버스 엔지니어링은 알 수 없거나 최소한으로 문서화되어 있는 네트워크 프로토콜의 사양을 추출하는 것으로 네트워크 관리 및 보안을 위해 점차 중요해지고 있다.

프로토콜 리버스 엔지니어링의 핵심은 네트워크 메시지의 구조를 분석하는 것으로 프로토콜의 메시지 형식과 의미, 메시지의 전환 및 순서를 포함하여 가능한 상세한 구조를 추출하는 것을 목표로 하고 있다. 이를 위한 방법으로는 네트워크를 통한 프로토콜의 메시지 교환을 모니터링하여 메시지를 분석하는 네트워크 트레이스 기반 분석 방법과 통신 종단점이 네트워크 프로토콜의 메시지를 처리하는 방식을 분석하는 실행 트레이스 기반 분석 방법이 있다.

이전의 많은 연구[1,2]에서 각 분석 방법을 토대로 한 다양한 프로토콜 리버스 엔지니어링 시스템을 개발하였으나 사용하기에 각 분석 방법론마다 각각의 장단점이 존재한다. 이에 본 논문에서는 네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법 각각의 차이점을 크게 3가지 관점에서 상세하게 비교 분석한다.

본 논문의 구성은 본 장의 서론에 이어 2장에서 네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법의 3가지 관점에서의 차이점을 기술하고 마지막으로 3장에서 결론 및 향후 연구에 대해 기술한다.

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 과학기술인문융합연구사업(No.NRF-2016M3C1B6929228) 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2015R1D1A3A01018057).

Ⅱ. 네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법의 차이점 분석

본 장에서는 프로토콜 리버스 엔지니어링의 네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법을 크게 일반적 차이, 메커니즘적 차이, 확장성 차이로 나누어서 기술한다.

1. 일반적 차이

네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법의 가장 큰 차이점은 프로토콜 리버스 엔지니어링의 입력으로 네트워크 트레이스를 사용하는지 실행 트레이스를 사용하는지이다. 네트워크 트레이스란 네트워크상의 해당 프로토콜을 따르는 패킷을 모니터링하여 수집한 하나의 파일을 의미하며 실행 트레이스란 해당 프로토콜을 따르는 프로그램 바이너리(Program Binary)의 실행을 로깅한 하나의 파일이다.

네트워크 트레이스 기반 분석 방법은 해당 프로토콜의 네트워크 패킷을 모니터링하여 캡쳐한 각 네트워크 트레이스들을 입력으로 분석하는 방법이다. 이 방법은 타겟 네트워크와 외부 네트워크를 연결하는 최 앞단 라우터에서 발생하는 트래픽을 캡쳐하여 클라이언트와 서버 간의 송수신 메시지를 모두 분석할 수 있는 방법이다.

그림 1에서와 같이 네트워크 트레이스 기반 분석은 네트워크 트래픽을 캡쳐하여 Flow단위로 재구성하고 분석하고자 하는 프로토콜의 트래픽만을 수집한 정답지 트래픽을 생성한다. 이를 기반으로 단일 프로토콜의 메시지가 기록되어 있는 각 트레이스를 프로토콜 리버스 엔지니어링의 입력으로 적용한다.

실행 트레이스 기반 분석 방법은 해당 프로토콜을 따르는 프로그램 바이너리를 모니터링하여 실행 명령, 메모리 사용, 시스템 콜, 특정 파일 시스템 접근 등을 기반으로 로깅한 각 실행 트레이스를 입력으로 분석하는 방법이다. 이 방법은 해당 프로토콜을 구현하는 프로그램 바이너리를 사용할 수 있는 환경에서만 분석이 가능하다. 그러나 비공개 프로토콜을 구현하는 프로그램 바이너리에 접근하는 것은 현실적으로 어려운 경우가 많

다. 예를 들어, 악성 봇넷의 명령 및 제어를 하는 서버의 프로그램 바이너리는 타겟 네트워크가 아닌 외부 네트워크에 존재할 가능성이 크며 해당 서버는 은닉하고 있기 때문이다. 또한, 일반적으로 입력 메시지를 처리하는 동안 클라이언트의 프로그램 바이너리를 관찰하여 분석하기 때문에 프로토콜의 메시지를 수신하는 호스트의 수신된 메시지만을 분석한다. 완벽한 프로토콜의 구조분석을 위해서는 서버의 프로그램 바이너리의 실행 또한 관찰하여야 하지만 대부분의 경우 서버의 프로그램 바이너리는 외부 네트워크에 존재하므로 현실적으로 불가능하다.

그림 1에서와 같이 실행 트레이스 기반 분석 방법은 타겟 네트워크에서 해당 비공개 프로토콜을 발생시키는 호스트에 프로그램 바이너리의 실행을 모니터링하는 실행 모니터링 시스템을 구축하고 각 호스트의 여러 실행 트레이스를 수집하여 프로토콜을 분석한다.

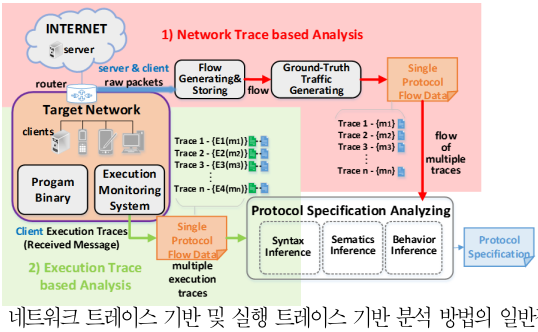

그림 1. 네트워크 트레이스 기반 및 실행 트레이스 기반 분석 방법의 일반적 차이

2. 메커니즘적 차이

네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법의 메커니즘적 차이점은 프로토콜의 구문, 의미, 프로토콜의 상태 머신을 위한 행위 추론 단계에서 각 단계별로 수직적 분석과 수평적 분석이 가능한가이다. 수직적 분석이란 수집한 여러 트레이스의 메시지를 종합하여 분석하는 방법이고, 수평적 분석이란 한 트레이스에서 시간 순으로 나열되어 있는 메시지를 종합하여 분석하는 방법이다.

그림 2에서 m은 각 트레이스에 기록되어 있는 메시지 집합을 의미하며 E는 한 호스트에서 바이너리 프로그램을 1회 실행한 실행 트레이스를 의미한다.

네트워크 트레이스 기반 분석 방법의 경우 모든 트레이스의 패킷들을 모아 m_1, m_2, ..., m_n을 하나의 메시지 집합인 M으로 통합하여 구문 추론, 의미 추론, 행위 추론의 모든 단계에서 수평적 분석과 수직적 분석이 가능하다. 이와 대조적으로 실행 트레이스 기반 분석 방법의 경우 네트워크 메시지를 분석하기 위해서는 프로그램 바이너리를 실행한 호스트의 실행을 로깅한 하나의 실행 트레이스에서 메모리 접근 및 CPU 명령어 등을 분석해야하므로 각 m은 각 E에 종속적이다. 따라서 실행 트레이스와 그에 해당하는 메시지 집합을 항상 한 쌍으로 생각하여 메시지를 분석해야하므로 구문 추론 및 의미 추론 단계에서 수직적 분석이 불가능하다. 그러나 행위 추론의 경우에는 구문 추론 및 의미 추론의 결과로 모든 트레이스에 대한 종합적인 메시지 유형별 클러스터들을 구축할 수 있으므로 수직적 분석이 가능하다.

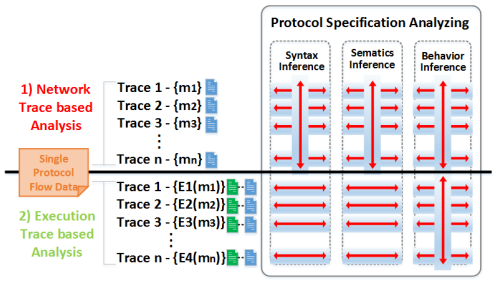

그림 2. 네트워크 트레이스 기반 및 실행 트레이스 기반 분석 방법의 메커니즘적 차이

3. 확장성 차이

네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법의 차이점 중 하나는 확장성이다. 네트워크 트레이스 기반 분석 방법 같은 경우 패킷 수집 및 프로토콜별 분류의 자동화가 가능하기 때문에 그림 3의 path-2와 같은 추가적인 모듈의 구축이 가능하다.

path-2의 Known Protocol Eliminating 모듈에서는 헤더 시그니쳐 기반, 페이로드 시그니쳐 기반, 통계 시그니쳐 기반, 행위 시그니쳐 기반 등의 분류 방법을 통하여 선행적으로 다량의 알려진 프로토콜의 트래픽을 분류하여 필터링한다. Unknown Protocol Clustering 모듈에서는 앞서 필터링한 트래픽에 한해서 단일 프로토콜 트래픽으로 그룹화하기 위하여 머신러닝 기법을 통해 유사한 통계 정보 패턴을 가지는 트래픽들을 클러스터링한다.

이를 통해 정답지 단일 프로토콜 수집의 자동화가 가능하며 서로 다른 단일 프로토콜 트래픽 클러스터들을 한 번에 여러 개를 생성할 수 있으므로 프로토콜 리버스 엔지니어링의 다양한 입력을 신속하게 확보할 수 있다. 또한, 실시간 네트워크 모니터링 분야와 연동하여 프로토콜 구조분석의 출력을 DPI 기반 악성 행위 탐지나 침입탐지시스템의 시그니쳐 추출 시스템에 신속하게 전달이 가능하다.

반면에, 실행 트레이스 기반 분석 방법의 경우 새로운 프로토콜을 분석할 때마다 해당 프로토콜을 따르는 프로그램 바이너리의 입수가 필요하기 때문에 다수의 다양한 단일 프로토콜 수집의 자동화가 불가능하다.

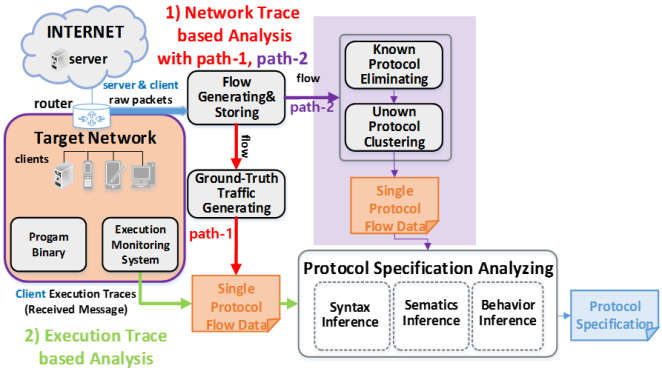

그림 3. 네트워크 트레이스 기반 및 실행 트레이스 기반 분석 방법의 확장성 차이

Ⅲ. 결론 및 향후 연구

본 논문에서는 프로토콜 리버스 엔지니어링의 네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석 방법을 일반적, 메커니즘적, 확장성 3가지 측면에서 상세하게 분석하였다. 향후 연구로는 기 연구된 프로토콜 리버스 엔지니어링의 한계점 및 고려사항을 분석하여 새로운 시스템을 개발할 계획이다.

참 고 문 헌

[1] Juan Caballero, Dawn Song "Automatic protocol reverse-engineering: Message format extraction and field semantics inference", International Journal of Computer and Telecommunications Networking, Vol. 57, Issue. 2, pp.451-474 (2012)

[2] Jian-Zhen Luo, Shun-Zheng Yu "Position-based automatic reverse engineering of network protocols", Jounal of Network and Computer Applications Vol. 36, Issue. 3, pp. 1070-1077 (2013)