프로토콜 리버스 엔지니어링의 이상적인 메커니즘 정의

구영훈, Baraka D. Sija, 김명섭
고려대학교 컴퓨터정보학과

{gyh0808, sijabarakajia25, tmskim}@korea.ac.kr

Defining Ideal Mechanism of Protocol Reverse Engineering

Young-Hoon Goo, Baraka D. Sija, Myung-Sup Kim
Dept. of Computer and Information Science, Korea Univ.

요 약

오늘날의 인터넷은 네트워크 고속화 및 유비쿼터스 환경으로 인해 다양한 기능을 가지는 응용 및 악성 행위의 출현으로 복잡 다양한 비공개 프로토콜이 발생하고 있다. 효율적인 네트워크 관리 및 보안을 위하여 프로토콜 리버스 엔지니어링은 네트워크 모니터링 및 보안 분야의 많은 부분에서 필수적이다. 기존의 다양한 연구에서 프로토콜 리버스 엔지니어링을 위한 방법론을 제시하고 있지만, 현재까지 표준화된 방법론은 없으며 복잡 다양한 오늘날의 대용량 네트워크에 적용하기에 각각의 장단점이 존재한다. 이에 본 논문에서는 프로토콜 리버스 엔지니어링의 분류 방법을 설명하고 효율적인 프로토콜 리버스 엔지니어링의 이상적인 메커니즘을 정의한다.

Ⅰ. 서 론

오늘날의 인터넷은 네트워크 고속화 및 유비쿼터스 환경으로 인해 대용량의 트래픽이 발생하고 있으며 이에 따라 다양한 기능을 가지는 응용 및 악성 행위가 기하급수적으로 증가하고 있다. 이러한 환경 하에 발생하는 복잡 다양한 프로토콜 중 다수는 알려지지 않았거나 최소한으로 문서화되어 있는 비공개 프로토콜이다. 효율적인 네트워크 관리 및 보안을 위하여 비공개 프로토콜에 대한 분석은 필수적이다. 예를 들어, 알 수 없는 비공개 프로토콜의 구조를 분석함으로써 네트워크 모니터링 분야에서는 타겟 네트워크에서 발생하는 알 수 없는 트래픽에 대한 정보를 습득할 수 있으므로 정상 응용에서 발생하는 트래픽을 플로우 단위로 분류해 내고, 분류되지 않은 소량의 플로우에서 비공개 프로토콜이 발생시키는 트래픽을 분류하여 네트워크 사용 현황 파악과 확장 계획 수립, 특정 프로토콜에 대한 대역폭 조절 등의 관리에 활용이 가능하다. 네트워크 보안 분야에서는 네트워크 취약성을 분석하거나 기존에 알려지지 않은 공격에 대한 탐지 및 차단을 위한 방화벽과 침입 탐지 시스템에 유용한 정보를 제공하는 데에 도움이 될 수 있다.

기존의 다양한 연구에서 비공개 프로토콜 분석을 위한 프로토콜 리버스 엔지니어링의 방법론을 제시하고 있지만, 현재까지 표준화된 방법론은 없으며 복잡 다양한 오늘날의 대용량 네트워크에 적용하기에 각각의 장단점이 존재한다. 이에 본 논문에서는 프로토콜 리버스 엔지니어링의 분류 방법을 설명하고 효율적인 프로토콜 리버스 엔지니어링을 위한 이상적 메커니즘을 정의한다.

본 논문은 본 장의 서론에 이어 2 장에서 프로토콜 리버스 엔지니어링의 이상적인 메커니즘 정의를 기술하고 3 장에서 결론 및 향후 연구에 대해 기술한다.

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

Ⅱ. 프로토콜 리버스 엔지니어링의 이상적인 메커니즘 정의

프로토콜 리버스 엔지니어링은 자동화 여부, 프로토콜 구조분석을 위한 입력에 따라 분류할 수 있다. 본 장에서는 프로토콜 리버스 엔지니어링의 분류 방법을 기술하고 이를 토대로 한 프로토콜 리버스 엔지니어링의 이상적인 메커니즘을 정의한다.

1) 자동화 여부에 따른 분류

수동적 프로토콜 리버스 엔지니어링에는 2005 년 Borisov 가 발표한 GAPA(Generic Application-level Protocol Analyzer)를 예시로 들 수 있다. GAPA 는 손으로 작성한 문법을 통해 네트워크 프로토콜의 사양을 확인하고 파싱할 수 있는 프레임워크이다. 그러나 수동적 리버스 엔지니어링은 모든 프로토콜 요소를 정확하게 복구할 수 있지만 오류가 발생하기 쉬우며 시간이 굉장히 많이 소모되는 수작업이므로 기하급수적으로 증가하는 응용의 속도에 대처할 수 없다. 따라서 오늘날과 같은 고속의 대용량 네트워크 환경과 고도로 지능화된 다양한 악성행위에 대한 대처를 위해서는 자동적 프로토콜 리버스 엔지니어링이 필수적이다.

2) 입력에 따른 분류

프로토콜 리버스 엔지니어링은 입력으로 네트워크 트레이스를 사용하느냐 실행 트레이스를 사용하느냐에 따라 네트워크 트레이스 기반 분석 방법과 실행 트레이스 기반 분석으로 나눌 수 있다. 네트워크 트레이스 기반 분석 방법은 해당 프로토콜의 네트워크 패킷을 모니터링하여 캡쳐한 각 네트워크 트레이스들을 입력으로 분석하는 분석 방법이다. 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 트레이스들을 입력으로 분석하는 방법이다. 이 분석 방법은 실제 프로그램 바이너리의 실행을 분석하기 때문에 프로토콜 구조 분석의 정밀도가 향상될 수 있으나 비공개 프로토콜의 프로그램 바이너리 입수는 현실적으로 어렵다.

3) 프로토콜 리버스 엔지니어링의 이상적인 메커니즘

프로토콜 리버스 엔지니어링 출력으로는 크게 구문, 의미, 프로토콜 상태 머신이 있다. 프로토콜의 구문은 각 유형별 메시지를 구성하는 필드들의 형식을 말한다. 필드의 형식은 필드를 구분할 수 있는 구분자 혹은 offset 과 depth 로 표현할 수 있는 경계와 필드들이 가지는 값, 순서 등을 포함한다. 프로토콜의 의미는 메시지를 구성하는 각 필드들이 뜻하는 바를 말한다. 대다수의 기존 연구에서는 의미를 추출하기 위한 방법으로 자주 사용되는 필드의 의미 유형들을 미리 정의하고 이에 해당하는 필드를 휴리스틱한 방법으로 찾는다. 프로토콜 상태 머신은 프로토콜의 유형별 메시지들의 행위를 분석하여 발생 순서, 발생 조건, 메시지의 방향 등을 표현하기 위한 유한 상태 오토마타이다.

선행 연구의 방법론 중에는 프로토콜의 구문만을 추출하는 데에 초점을 맞춘 방법론도 있으며 유한 상태 머신만을 추출하는 데에 초점을 맞춘 방법론도 있다. 상세한 프로토콜의 구조분석을 위해서는 프로토콜 구문, 의미, 프로토콜 상태 머신을 포함하여 가능한 모든 정보를 추론하여야 하는 것이 이상적이다.

이러한 프로토콜 리버스 엔지니어링의 메커니즘의 순서로는 구문 추론(syntax inference) - 의미 추론(semantics inference) - 프로토콜 상태 머신을 위한 행위 추론(behavior inference)의 단계가 타당하다. 그 이유는 다음과 같다.

의미를 추론하기 위해서는 미리 정의한 각 유형별 의미에 해당하는 필드들을 찾아야하므로 먼저 메시지에서 필드를 구분할 수 있어야 한다. 이를 위해서는 프로토콜의 구문을 먼저 추론하여야 하는 것이 필수적이다. 그리고 프로토콜 상태 머신의 추출을 위해서는 각 노드를 정의하기 위한 메시지 클러스터링이 필수적이다. 각 메시지의 구문 및 의미를 추론하여 공통된 프로토콜의 구조를 결정하기 위해서는 최종적으로 메시지 클러스터링을 수행하여야 하며 이는 프로토콜 상태 머신 추론을 위한 전제 조건이 될 수 있다. 하지만 이와 반대의 순서인 프로토콜 상태 머신을 추론한 후 구문과 의미를 추론하는 것은 프로토콜 상태 머신 추론 시 선행적으로 노드를 정의하기 위한 메시지 클러스터링을 수행한 후 구문 및 의미를 추론할 때 최종적 프로토콜 구조를 결정하기 위하여 메시지 클러스터링을 다시 한 번 수행하여야 하므로 비효율적이다.

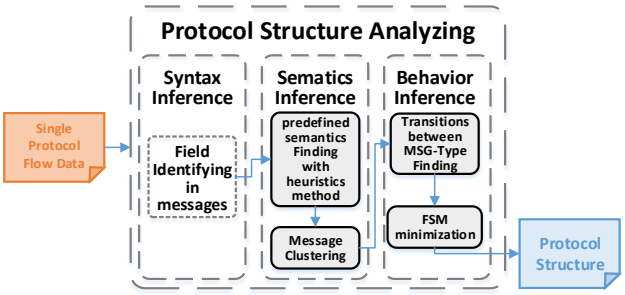

Figure 1. 프로토콜 리버스 엔지니어링의 메커니즘

Figure1 은 프로토콜 리버스 엔지니어링의 이상적인 메커니즘이다. 먼저 다양한 프로토콜 중 하나의 단일 프로토콜을 입력받는다. 이 단일 프로토콜은 다중 계층 구조의 프로토콜일 수 있다.

첫 번째 단계로 구문 추론 단계에서는 각 입력이 네트워크 트레이스인지 실행 트레이스인지에 따라 각기 다른 방법으로 메시지 내의 필드를 식별한다. 입력이 네트워크 트레이스인 경우, 여러 메시지에 대해 키워드 및 통계 분석을 통해 필드를 식별한다. 입력이 실행 트레이스인 경우, 특정 의미를 가지는 필드는 상대적으로 프로그램 바이너리에서 메모리 접근이 자주 일어난다는 점을 바탕으로 하여 CPU 명령의 피연산자 메모리 접근 빈도 등의 특성을 관찰하여 필드를 식별한다.

두 번째 단계인 의미 추론 단계에서는 몇 가지 의미의 유형을 미리 정의하고 모든 필드에 대하여 각 유형별 의미에 만족하는 필드가 있는지를 휴리스틱한 방법으로 추론한다. 그리고 최종적인 유형별 메시지의 프로토콜 구문과 의미를 결정하기 위해 메시지 클러스터링을 한다.

마지막으로 행위 추론 단계에서는 앞선 단계의 결과인 유형별 메시지 클러스터들을 유한 상태 머신의 노드로 결정하고 노드들의 변환을 확인하기 위해 메시지의 행위를 분석하여 관찰된 노드들 사이의 변환을 엣지로 연결한다. 최종적으로 유한 상태 머신을 최소화하여 프로토콜 상태 머신을 추출한다.

Ⅲ. 결론 및 향후 연구

본 논문에서는 프로토콜 리버스 엔지니어링의 분류 방법을 기술하고 이를 토대로 한 이상적인 메커니즘을 정의하였다. 향후 연구로는 정의한 이상적인 메커니즘을 토대로 한 새로운 프로토콜 리버스 엔지니어링의 알고리즘을 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, 2012, Vol. 57, Issue. 2, pp. 451- 474

[2] John Narayan, Sandeep K. Shukla, T. Charles Clancy," A Survey of Automatic Protocol Reverse Engineering Tools," Journal ACM Computing Survey, 2016, Vol. 48, Issue. 3, No. 40