실행 시간 확률분포를 고려한 실시간 시스템 은닉 타이밍 채널

곽재헌 ○, 이진규‡
성균관대학교 소프트웨어학과
0jaehunny0@gmail.com, jinkyu.lee@skku.edu

Covert Timing Channel Considering Execution Time Distribution in Real-Time Systems

Jaeheon Kwak○, Jinkyu Lee‡
College of Software, Sungkyunkwan University

요 약

실시간 시스템(real-time system)에서는 모든 태스크(Task)가 데드라인(deadline) 이전에 끝남을 보장해야 한다. 이를 위해 실시간 시스템에선 최악의 경우에도 태스크들이 데드라인을 지킬 수 있도록 태스크의 WCET(Worst Case Execution Time)을 기반으로 태스크들을 스케줄링 한다. 하지만 실제로는 태스크들의 실행 시간은 WCET와 차이가 있다. 본 논문에서는 이를 고려하여 실시간 시스템의 태스크들끼리 실행 시간을 조절하여 정보를 전달하는 은닉 타이밍 채널을 만드는 방법과, 태스크들의 실행 시간 확률분포를 통하여 은닉 타이밍 채널의 정확도를 향상 시킬 수 있는 방법을 제시한다.

1. 서 론

실시간 시스템(real-time system)은 어떤 태스크(Task)라도 데드라인(deadline) 이전에 끝나지 않으면 치명적인 손실을 입을 수 있는 환경에서 사용하는 시스템이다. 이 때문에 실시간 시스템에선 모든 태스크들이 각각의 데드라인을 지킬 수 있도록 태스크들을 스케줄링 한다. 이런 태스크 스케줄링 방법에는 RM(Rate Monotonic), EDF(Earliest Deadline First), LLF(Least Laxity First) [1,2] 등의 스케줄링이 있다. 이러한 태스크 스케줄링에선 스케줄러가 태스크들의 실행 시간(Execution Time)이 모두 길어지는 최악의 경우에도 모든 태스크들이 데드라인을 지킬 수 있는 것을 보장 해야 하는데, 이 때문에 스케줄러는 아예 태스크가 WCET(Worst Case Execution Time) 만큼의 실행 시간을 가진다 가정하고 스케줄링을 한다.

실시간 시스템의 스케줄러가 모든 태스크가 항상 WCET만큼의 실행 시간을 가질 것이라고 가정하고 스케줄링을 하는 반면, 실제 태스크들의 실행 시간은 WCET과는 큰 차이를 보이게 된다. 실시간 시스템에서 사용되는 태스크들의 실행 시간을 분석한 연구[3]는 태스크들의 실제 실행 시간을 측정했을 때 WCET의 절반 정도의 시간을 가졌고, WCET 만큼의 실행 시간을 가질 확률은 극히 적다는 것을 밝혔다.

본 논문에선 앞서 설명한 실시간 시스템의 특징들을 이용한 새로운 은닉 채널(Covert Channel) 공격법을 제시하려 한다. 기존에 제시된 실시간 시스템의 특징을 이용한 은닉 타이밍 채널 (Covert Timing Channel) 공격법[4]은 멀티 - 레벨 보안 시스템(Multi - Level Security Systems)에서 RM 스케줄 방법일 경우에 사전에 공모한 태스크들의 실행 시간과 응답 시간(Response Time)을 이용하여 보안 레벨이 높은 태스크의 정보를 보안 레벨이 낮은 태스크로 전달할 수 있는 은닉 타이밍 채널을 만들었다. 본 논문에선 해당 연구 결과를 발전시켜 스케줄링 방법에 제약이 없고, 태스크의 실제 실행 시간의 확률분포를 고려하여 보다 더 에러율이 적은 은닉 타이밍 채널을 만드는 방법을 새로이 제시한다.

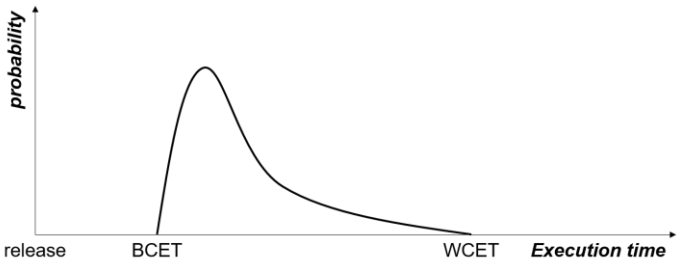

그림 1. 태스크의 실제 실행 시간 확률 분포

2. 배경 지식

2-1. WCET와 실제 실행 시간 확률분포

실시간 시스템에선 최악의 경우에도 데드라인을 보장하기 위해, 태스크들이 WCET만큼 실행 시간을 가질 것이라 가정하고 스케줄링을 한다. 하지만 태스크들이 항상 WCET만큼의 실행 시간을 가지는 것이 아니기 때문에 실제 태스크들은 이보다 적은 실행 시간을 가진다. 이와 관련해 태스크들의 실행 시간을

WCET와 비교하여 분석한 연구[3]는 태스크의 실행 시간이 대체로 그림 1과 같은 형태의 확률분포를 따르며 WCET와는 크게 다른 것을 밝혔다.

만약 태스크의 실행 시간이 특정 개형의 확률분포를 따른다 가정하면, 우리는 태스크의 실행 시간을 일부를 수집하여 실행 시간의 확률분포 모형을 추정하고, 향후 발생 가능한 실행 시간의 확률을 계산하고 앞으로의 실행 시간 확률분포를 구할 수 있다. 이러한 방법은 3장에서 더 자세히 다룰 예정이다.

2-2. 은닉 타이밍 채널

은닉 채널은 멀티 레벨 보안 시스템에서 사용되는 공격법이다. 멀티 레벨 보안 시스템에선 공개된 채널을 통해 보안 레벨이 높은 사용자의 정보가 보안 레벨이 낮은 사용자에게 흘러 가는 것이 금지되어 있는데, 은닉 채널은 시스템의 허점을 이용하여 노출되지 않은 통신 채널을 새로 구성하여 정보를 유출해낸다.

은닉 채널을 구성하는 방법에는 여러가지가 있는데, 그 중 이 논문에서는 은닉 타이밍 채널을 사용하려 한다. 은닉 타이밍 채널이란 특정 프로세스나 태스크가 할당 받은 시스템 자원을 조정해서 정보를 전달하는 방법이다. [5]

선행 연구[4]에선 태스크들의 실행 시간을 조정하여 은닉 타이밍 채널을 설계하였는데, 보안 레벨과 스케줄 우선순위(Priority)가 더 높은 사용자(전송자) T_H 가 태스크의 실행 시간을 조정하면, 보안 레벨과 스케줄 우선 순위가 낮은 사용자(수신자) T_L 의 태스크의 응답 시간에 영향 주게 된다. 이 때, 전송자와 수신자 사이에 공모를 통해 전송자가 실행 시간을 늘려 수신자의 응답 시간이 늘리면 '1' 비트를 송수신하고, 실행 시간을 줄여 수신자의 응답 시간이 줄이면 '0' 비트를 송수신한다고 사전에 약속을 하면 은닉 타이밍 채널을 구현할 수 있다. 본 논문은 T_H 와 T_L 의 우선순위에 제약을 없애고, T_H 의 작업(Job)이 T_L 의 작업보다 먼저 릴리즈 된 경우만 비트를 송수신하여 스케줄링 방식과 상관 없이 은닉 타이밍 채널을 구성할 수 있다.

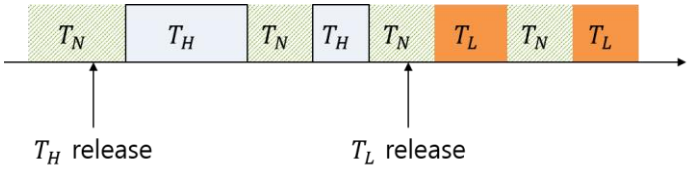

그림 2. 실시간 시스템에서 태스크들의 스케줄 예시

3. 은닉 타이밍 채널 설계

3-1. 실시간 시스템 특성 기반 은닉 타이밍 채널 구성

은닉 타이밍 채널을 구성하기 전, 채널을 구성할 T_H, T_L의 능력을 정의 내리면 아래와 같다.

T_H: 해당 태스크가 릴리스(Release)할 작업들의 실행 시간을 WCET 미만의 범위 안에서 개별적으로 조절하는 것이 가능하고 이미 릴리즈 된 작업의 실행 시간을 조절할 수는 없다. 또한, T_L 의 작업들이 릴리스 되는 주기(Period)와 데드라인을 안다.

T_L: 해당 태스크가 릴리스 할 작업들의 실행 시간을 WCET 미만의 범위 안에서 개별적으로 조절하는 것이 가능하고, 이미 릴리즈 된 작업의 실행 시간을 조절할 수는 없다. 또한, T_H 의 작업들이 릴리스 되는 주기와 데드라인을 안다.

그리고 T_H, T_L 두 태스크들 외에, 그림 2처럼 다른 태스크들에 의한 작업들의 집합을 T_N 이라 하며, 이 작업들에 대해선 어떠한 정보도 알 수 없고, 실행 시간들도 통제할 수 없다. 태스크의 종류에 따라 T_N 의 작업 중 일부는 T_H보다 우선순위가 높기도 하고, T_L보다 낮기도 하다. 이 T_N 때문에 2-2장에서 구현한 은닉 타이밍 채널 방식대로 T_H의 실행 시간을 조절하여 T_L에 정보를 전송할 경우 손실이 발생할 수 있다. T_N 에 의해서도 T_L 의 응답시간이 변하는데, 수신자 입장에서 길어진 응답시간이 T_H, T_N 중 어느 것에 의한 것인지 구분할 수 없기 때문이다. 따라서 T_N을 예측하여 은닉 타이밍 채널의 손실율을 줄일 필요가 있다.

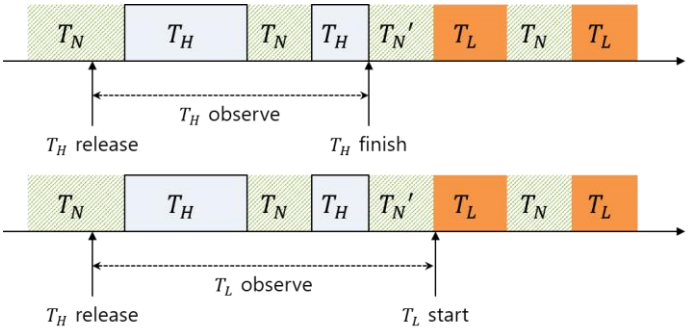

그림 3. T_H과 T_L이 관찰하는 T_N 실행 시간의 범위

3-2. 실행 시간 확률분포를 통한 추론

T_N 의 실행 시간은 T_H 이나 T_L 레벨에선 바꿀 수 없기에 통제가 불가능하다. 따라서 본 논문에선 2-1장에서 다룬 태스크의 실행 시간 확률분포를 통해 T_N 의 실행 시간을 예측하고, 이를 통해 전송 손실을 줄이는 방법을 제시하고자 한다. 이 방법을 단계 별로 구체적으로 기술하면 다음과 같다.

1) T_H, T_L은 k초 만큼 T_H의 실행 시간을 바꾸어 가며 T_N의 실행 시간 변화를 관찰한다.

T_H과 T_L은 서로 상대 태스크의 작업들이 정확히 언제 시작하고 언제 끝나는지는 알 수 없기 때문에 관찰할 수 있는 T_N 범위에 차이가 생기게 된다. 이 때문에 정확성을 어느 정도 포기하고 실질적으로 관찰 가능한 구간을 정하는 타협이 필요하다. 본 논문에선 T_H 의 경우 그림 3의 T_N' 구간의 직접 구하지 않는 방법을 사용하였다. T_H 는 T_H 작업의 릴리스 시간부터 완료 시간까지 구간을, T_L 은 T_H 작업의 릴리스 시간부터 T_L 작업의 시작 시간까지 구간을 관찰한다. T_N에 의해 T_L의

작업이 지연되는 시간을 X, 태스크가 릴리스 된 시점을 R, 실행 시간을 E, 작업이 시작된 시점을 S, 작업이 끝난 시점을 F, 데드라인을 D라고 하면, T_H는 $(F_H - R_H) - E_H$를 통해 자신의 작업이 T_N의 실행 시간에 의해 지연된 시간을 계산할 수 있고, 측정할 수 없는 T_N' 구간에 의한 T_L 이 지연될 시간은 $(R_L - F_H) * \frac{(F_H - R_H) - E_H}{(D_H - R_H) - E_H}$를 통해 어느 정도 추론해 낼 수 있고, T_H가 예측하는 X 값은 $X_H = (R_L - F_H) * \frac{(F_H - R_H) - E_H}{(D_H - R_H) - E_H}$ 이 된다. T_L의 경우엔 T_N' 포함한 구간을 관찰함으로, 바로 $X_L = S_L - R_H - E_H$ 를 통해 작업이 T_N의 실행 시간에 의해 지연된 시간을 계산할 수 있다.

2) 관찰을 통하여 T_N의 주기성과 T_H의 영향을 분석해 T_N에 확률분포를 구한다.

T_N을 관찰한 k초 동안 T_H과 T_L 각자 T_N의 주기성과 T_H 실행 시간이 길 때와 짧았을 때에 대해 경우를 나눌 수 있을 것이다. 그 후 위에서 구한 X_H, X_L들을 하나의 태스크로 여기고, 일반적인 실행 시간 분포를 따른다 가정하면 각 경우에 따른 T_N이 T_L을 지연 시킬 시간의 확률분포를 구할 수 있다.

3) T_L의 응답시간이 지연될 확률분포를 계산하고, 은닉 타이밍 채널을 통해 비트를 송수신한다.

비트를 송수신할 때마다 $E_H + X_H$, $E_H + X_L$을 계산해 T_L의 응답시간의 확률분포를 구하여 T_H은 예상 전송 손실율을 구하고, T_L은 응답시간에 따른 예상 비트를 추론해내어 전송 손실을 줄일 수 있게 된다.

4. 은닉 타이밍 채널 전송 용량 분석

T_H과 T_L은 이상적인 경우엔 T_H의 작업이 T_L의 작업보다 앞에 존재할 때마다 한 비트씩 전송을 할 수 있다. 이 경우 전송 용량은 둘 중 주기가 큰 작업에 의해 결정된다. 주기가 큰 작업의 주기 τ, 전송 용량을 C라 한다면 $C = \frac{1}{\tau}$이다. 하지만 T_N에 의해서 전송 손실이 일어날 경우 전송 용량이 줄어드는데, 2)에서 나눈 경우 중 i번째 경우 T_N에 의해 전송 손실이 일어날 확률을 P_i라 하면, 전송 손실은 $\sum P_i * \tau_i$이고, 전송 용량은 $C = \frac{1}{\tau} - \sum P_i * \tau_i$로 구할 수 있다.

5. 방어 방법

1) 스케줄러가 보안을 위해 작업이 WCET보다 일찍 끝난 경우에도 새 작업을 실행하는 대신 WCET 시간만큼 아이들(IDLE)하게 유지 시키는 경우 해당 은닉 타이밍 채널을 생성할 수 없다.

2) TaskShuffler[6] 같이 작업들을 무작위로 섞는 방어 기법이 스케줄러에 적용될 경우 은닉 타이밍 채널의 에러율이 급증할 수 있다.

3) 태스크들의 실행 시간을 분석하여 일반적인 확률분포를 따르지 않는 인위적인 태스크들을 선별해내 은닉 타이밍 채널을 생성하는지 모니터링할 수 있다.

6. 결론 및 향후 연구

본 논문은 실시간 시스템에서 태스크들의 주기성을 이용한 은닉 타이밍 채널 공격법을 제안하고, WCET와 실제 실행 시간의 차이점을 이용하여 전송 손실을 줄일 수 있는 방법을 제시하였다. 또한, 주기적으로 릴리스 되는 태스크의 작업들의 특성을 이용하여 전송 용량을 확인하였으며, 해당 공격법을 방어할 수 있는 방안을 3가지 제시하였다.

향후 연구 계획은 해당 이론을 바탕으로 실제 실시간 시스템에서 은닉 타이밍 채널을 구현하여 전송 손실을 얼마나 줄일 수 있나 확인하고, 손실을 막지 못한 비트들을 parity bit를 이용해 복구하여 최적의 전송 용량을 가지는 은닉 타이밍 채널을 연구할 것이다.

Acknowledgement

이 논문은 정부(미래창조과학부, 교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2017R1A2B2002458, NRF-2016R1D1A1B03930580). 이 논문은 또한 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음(2015-0-00914).

참 고 문 헌

[1] Liu, Chung Laung, and James W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," Journal of the ACM (JACM) 20.1, 1973.

[2] Jane W. S. Liu, “Real-Time Systems,” Prentice Hall, 2000.

[3] Hansen, Jeffery, Scott A. Hissam, and Gabriel A. Moreno. "Statistical-based wcet estimation and validation." Proceedings of the 9th Intl. Workshop on Worst-Case Execution Time (WCET) Analysis. 2009.

[4] Son, Joon. "Covert timing channel analysis of rate monotonic real-time scheduling algorithm in mls systems." Information Assurance Workshop, 2006 IEEE. IEEE, 2006.

[5] Wray, John C. "An analysis of covert timing channels." Journal of Computer Security 1.3-4, 1992.

[6] Yoon, Man-Ki, et al. "TaskShuffler: A schedule randomization protocol for obfuscation against timing inference attacks in real-time systems." Real-Time and Embedded Technology and Applications Symposium (RTAS), 2016 IEEE. IEEE, 2016.