

2019 한국정보보호학회 하계학술대회

CISC-S'19

Conference on Information Security and Cryptography-Summer 2019

2019. 6. 20(목)~22(토)

부산 동명대학교 (제1정보통신관, 중앙도서관)

주최



한국정보보호학회
Korea Institute of Information Security & Cryptology

주관



동명대학교
TONGMYONG UNIVERSITY

후원



국가정보원
National Intelligence Service Korea



과학기술정보통신부



행정안전부



KISA 한국인터넷진흥원

ETRI

한국전자통신연구원
Electronic and Telecommunications
Research Institute

NSR

국가보안기술연구소
National Security Technology Research Institute

IITP

정보통신기획평가원

AhnLab

TRACK 5 - 311호 소강의실

세션	발표논문
A5 국방보안 좌장 : 박기웅 교수 (세종대)	국방 IT 환경 변화에 따른 내부자 위협 완화 방안 원경수, 김승주 (고려대학교)
	국방 SW 운영 및 유지보수 단계 공급망 공격 대응 이민호* 박창욱** 김완주* 임재성* (*아주대학교, **세종대학교)
	위협 모델링을 이용한 정찰용 무인항공기 위협 분석 이규원, 윤주범 (세종대학교)
	사이버전 훈련시스템에서의 행위 자동화 기법 연구 배근우, 천종희, 이동환, 김용현, 안명길 (국방과학연구소)
	사이버 위협 인텔리전스 기반 사이버 킬 체인 모델 공성현, 이창훈 (서울과학기술대학교)

B5 스마트그리드/ 스마트시티 좌장 : 강동주 교수 (고려대)	6월 21일(금) 14:40 ~ 16:10	OIML D 31 규격 적합성을 위한 전자식 계량기의 정보보증 요구사항 주승환, 송광재, 송상훈, 이시우 (한국산업기술시험원)
		에너지 인터넷 환경에서의 보안 설계 및 실험 한주홍, 오진혁, 이재훈, 윤승환, 이민호, 이옥연 (국민대학교)
		스마트팩토리 보안위협 분석 및 대응방안 제시 이주찬, 김장훈, 서정택 (순천향대학교)
		산업제어시스템 내 무선통신기술 활용을 위한 보안 요구사항 연구 김장훈, 이주찬, 서정택 (순천향대학교)

C5 네트워크보안 좌장 : 조효진 교수 (한림대)	6월 22일(토) 10:00 ~ 11:30	DTLS 기반의 안전한 MQTT-SN 프로토콜 남혜민, 송지은, 이길희, 박창섭 (단국대학교)
		SNI 차단 회피 기법 연구 봉기정, 진혜윤, 김지수, 박수진, 엄홍열 (순천향대학교)
		JavaScript를 활용한 웹 사이트 Spectre 취약점 분석 이민경, 조정석, 박진 (아주대학교)
		WPA3 표준의 키 생성 과정 성능 향상에 관한 연구 권송희, 최형기 (성균관대학교)
		NS-3 시뮬레이션을 이용한 무선 센서 네트워크 FRP 프로토콜 구현 실험 연구 김종신, 윤혜민, 권태경 (연세대학교)

JavaScript를 활용한 웹 사이트 Spectre 취약점 분석

이민경*, 조정석**, 곽진***

*,***아주대학교 사이버보안학과

**아주대학교 컴퓨터공학과, 정보보호응용및보증연구실

Analysis of Spectre Vulnerabilities in a Website using JavaScript

MinKyung Lee*, Jeong-Seok Jo**, Jin Kwak***

*,***Department of Cyber Security, Ajou University.

**ISAA Lab., Department of Computer Engineering, Ajou University.

요 약

부채널 공격은 하드웨어 정보를 이용한 공격이다. 최근 CPU 성능 향상을 위한 기법을 활용하는 Spectre 취약점에서 부채널 공격을 시도할 수 있으며, 민감한 데이터로의 접근을 허용하게 함으로써, 데이터 유출을 가능하게 한다. 이를 이용한 공격은 파일을 직접 훔쳐가는 것이 아니라 데이터 정보를 읽어내는 것이기 때문에 공격이 발생할 경우 탐지가 어렵다는 문제가 존재하며, 이와 유사한 형태의 취약점들이 지속적으로 등장하고 있다.

따라서, 본 논문에서는 Spectre 취약점에 대해 분석하고 JavaScript를 활용하여 Google 크롬 62.0.3202버전에서 발생 가능한 Spectre 취약점 예시의 동작 과정을 단계별로 분석하였다.

I. 서론

부채널 공격은 크게 CPU가 사용하는 전력량을 분석하여 비밀키를 알아내는 전력 분석 공격 및 암호화 알고리즘 혹은 명령어를 처리할 때 발생하는 시간을 측정하여 데이터를 알아낼 수 있는 시간차 분석 공격 등이 존재한다[1]. Jann Horn 등 Google의 보안연구원들은 CPU 구조를 이용한 Spectre와 Meltdown으로 명명된 CPU 취약점을 공개하였으며, 이를 활용하여 부채널 공격이 가능하다[2]. 또한, Google Project Zero에 따르면, Spectre와 Meltdown의 발표 이후 Spectre와 Meltdown의 변종 공격기법들이 지속적으로 생겨나고 있으며, CPU 구조를 이용한 새로운 형태의 부채널 공격이 등장하고

있음을 발표하였다. 실제로 CPU 구조를 이용한 부채널 공격은 난이도가 높은 공격이지만, 스마트폰과 같은 IoT 기기의 확산과 이를 활용하는 다양한 서비스가 등장하면서 새로운 형태의 부채널 공격이 새로운 위협으로 발생할 가능성이 매우 높을 것으로 예상된다.

이에 따라, 본 논문은 Spectre 취약점에 대해 분석하고 JavaScript를 활용하여 Google 크롬 62.0.3202버전에서 발생 가능한 Spectre 취약점 예시[3]에 대해 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 CPU 구조인 파이프라인(Pipeline)과 분기 예측 기법(branch prediction), 그리고 flush+reload 공격에 대해 설명하고, 3장에는 Spectre 취약점 및 취약점 예시에 대해 분석한다. 마지막으로 4장에서 결론을 맺는다.

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2017R1E1A1A01075110).

II. 관련 연구

2.1 파이프라인(Pipeline)

프로세서는 하나의 명령에 대한 처리가 끝난 후 다른 명령을 받기 전까지는 어떠한 작업도 수행하지 않는다. 즉, 메모리 주소 이동을 위한 어드레스 버스를 통해 해당 메모리의 내용을 차례대로 읽어서 실행하는 구조이다. 하지만, 파이프라인(Pipeline)은 명령어를 효율적으로 수행하기 위한 메커니즘으로 명령어 실행을 메모리에서 읽기, 해석, 실행, 결과 쓰기 등의 작은 단위 작업으로 나누고, 이런 작업을 각각 분담하는 유닛을 따로 두어 수행하는 구조이다.

(그림 1)은 여러 개의 명령어가 수행되는 파이프라인 구조이다.



(그림 1) 파이프라인구조

Clock Cycle 2에서, 1 instruction이 Fetch됨과 동시에 2 instruction이 Decode된다. 즉, 명령어가 수행되는 동안 다음에 수행될 명령어가 해석되고 그 다음에 해석될 명령어가 레지스터에 배치될 수 있으며, 이 과정을 통해 프로세스의 동작 속도가 향상된다[4].

파이프라인의 4단계 과정은 다음과 같다[5].

□ Stage 1: Fetch

프로그램 메모리에서 명령어를 로드한다.

□ Stage 2: Decode

Instruction Register의 명령어를 해석한다.

□ Stage 3: Execute

명령어를 처리한다.

□ Stage 4: Write-back

처리된 결과를 레지스터 또는 데이터 메모리에 저장한다.

2.2 분기 예측 기법(Branch Prediction)

기존의 CPU는 조건문의 명령어가 들어올 경우, 파이프라인 구조의 Execution 단계에서 값을 계산하여 조건이 맞는지 판단한다. 하지만, 분기 예측 기법은 값을 계산하지 않고 Decode 단계에서 여러 가지 데이터에 근거하여 조건문의 판단 결과를 추측한 후 다음 명령을 파이프라인에 적재한다. 추측이 맞으면 명령을 수행하고, 그렇지 않으면 명령을 취소한다. 즉, CPU 분기가 맞는지 안 맞는지 판단하기 전에 분기 내부에 포함된 코드를 실행시킴으로써 CPU의 효율이 향상되는 것이다[3].

2.3 flush+reload 공격

flush+reload 공격은 하드웨어의 정보를 이용한 부채널 공격의 일종으로서, 데이터를 불러오는 시간을 측정하여 데이터를 알아낼 수 있다. 프로세스가 메인 메모리에 접근하기 위해서는 그 주소에 해당하는 데이터가 캐시 메모리에 존재하는지 확인한다. 만약 캐시 메모리에 존재한다면 캐시 메모리에서 읽고 캐시 메모리에 존재하지 않는다면 메인 메모리에 직접 접근한다. 캐시 메모리는 메인 메모리에 비해 처리속도가 빠르기 때문에 메인 메모리에서 자주 사용되는 위치의 데이터를 갖고 있다. 따라서, 캐시 메모리와 메인 메모리 접근시간 차이를 분석함으로써 캐시 메모리 데이터 내용을 유추할 수 있다.

이는, 사용자와 공격자가 같은 물리 메모리 주소에 접근할 수 있다면, 두 프로세스가 공유하는 메모리 영역을 대상으로 부채널 공격이 가능함을 보여준다[6, 7].

III. Spectre 취약점

3.1 Spectre 취약점 분석

운영체제는 메모리 보호를 위해 다른 프로그램에 속한 데이터에 접근할 수 없으며

접근하기 위해선 접근 권한이 있어야 한다. 하지만, Spectre(CVE-2017-5753, CVE-2017-5715)는 공격자의 응용 프로그램이 메모리의 임의의 위치에 접근할 수 있도록 하여 접근하지 못하는 메모리 영역을 접근할 수 있도록 하는 취약점이다.

Spectre는 분기 예측 기법을 이용한 취약점으로, 다음 명령어에 수행될 명령어를 추측하여 실행시킬 수 있다. 하지만, 해당 조건이 틀릴 경우 오실행을 막기 위해 파이프라인으로부터 명령을 취소한다. 이 과정에서 실행 취소된 명령어에서 읽은 데이터가 캐시 메모리에 적재되고 이를 flush+reload 공격을 사용하여 데이터를 읽어올 수 있다. 즉, 다른 프로그램의 메모리 정보를 읽어올 수 있으며 해당 데이터는 저장된 암호, 개인 사진, 전자 메일 등 중요한 데이터가 포함될 수 있다[3].

다음 [표 1]은 Spectre 취약점의 특징을 분석한 표이다.

[표 1] Spectre 취약점 특징

Contents	Description
Architecture	인텔, ARM, AMD
Entry	공격자의 코드 실행
Method	분기 예측 기법
Affect	다른 프로세스의 메모리 영역 접근 가능

Spectre 취약점은 인텔, ARM 및 AMD 제조사의 CPU가 영향을 받으며, 공격자의 코드 실행을 통하여 발생 가능하다. Spectre는 CPU의 성능 향상을 위해 활용되는 분기 예측

기법을 이용한 취약점으로 이를 통하여 공격자는 자신의 프로세스뿐만 아니라 다른 프로세스의 메모리 영역에 접근할 수 있다.

3.2 Spectre 취약점 예시 분석

Paul Kocher 등은 Google 크롬 62.0.3202버전에서 웹 사이트가 실행 중인 프로세스에서의 메모리를 읽어올 수 있는 공격 방법을 제시하였다. (그림 2)는 Paul Kocher 등이 제시한 Spectre 취약점이 발생 가능한 웹 사이트에 포함된 JavaScript 코드의 일부이다[3].

```

1 if (index < simpleByteArray.length) {
2   index = simpleByteArray[index | 0];
3   index = (((index * 4096) | 0) & (32*1024*1024-1)) | 0;
4   localJunk ^= probeTable[index|0]|0;
5 }
    
```

(그림 2) JavaScript 예시

해당 코드에서의 index는 공격자가 설정하는 값이며, simpleByteArray는 임의의 배열로써 공격 대상이 되는 주소에 접근하기 위한 배열이다. probeTable은 공격한 값을 읽어오기 위한 임의의 배열이다.

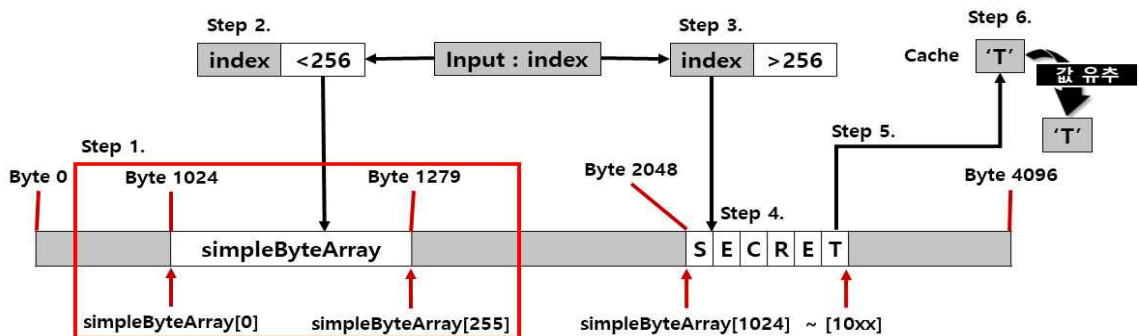
Spectre 동작 과정은 다음 (그림 3)과 같다.

Step 1. 설정

고정된 크기의 배열(simpleByteArray[256])을 준비한다.

Step 2. 조건 충족

분기 예측이 가능하도록 index 값이 항상 분기를 만족하도록 256보다 작은 값으로 설정한다.



(그림 3) Spectre 동작 과정

Step 3. 특정 값 대입

분기 예측이 실패하도록 `simpleByteArray.length` 값인 256보다 큰 값을 `index`에 대응시키며 이때의 `index` 값은 공격하고자 하는 메모리 주소 값이다.

Step 4. 분기 예측 수행

기존에는 `simpleByteArray[index]`와 이후의 명령문들은 조건문이 거짓이기 때문에 실행되지 않지만, 분기 예측으로 인해 `simpleByteArray[index]`와 함께 `probeTable[index]`와 같은 명령어들을 미리 실행한다.

Step 5. 비밀 데이터 접근

`simpleByteArray[index]`의 값이 캐시 메모리에 저장되어 있기 때문에 SECRET 내용 값인 'T'가 캐시 메모리에 저장되어 있다.

Step 6. 추론

모든 SECRET의 요소를 측정함으로써, flush+reload 공격을 통해 어떤 요소가 캐시 메모리에 적재 되었는지 알 수 있다. 예를 들어 `probeTable[simpleByteArray[1029]]`의 접근시간이 짧다면, `simpleByteArray[1029]`의 값이 캐시 메모리에 있다는 의미이고 SECRET 값의 일부인 'T'를 유추할 수 있다.

반복 수행(Step1. ~ Step 6.)

Step 1.~ Step 6.을 반복 수행하여 모든 SECRET 값을 유추하여 알아낼 수 있다.

위에서 설명한 바와 같이, Step 1. ~ Step 6.의 과정을 반복한 경우, 웹 사이트에서의 Spectre 취약점을 통해 하나의 탭에서 열려있는 웹 사이트가 다른 탭에 입력되는 비밀번호를 읽거나, 신용 카드 번호 등을 읽을 수 있다.

또한, 웹 브라우저 서비스를 제공할 수 있는 데스크톱, 태블릿과 같은 다양한 디바이스에서도 Spectre 취약점이 발생 가능하다[8].

IV. 결론

최근에는 Spectre 취약점이 등장한 이후로 이와 유사한 취약점들이 지속적으로 등장하고 있다. 이와 같은 취약점은 메모리 내에 저장된 민감한 정보를 탈취할 수 있다. 따라서, Spectre

변종 공격 기법들을 대응하기 위해서 CPU의 성능 향상을 유지하면서 안전한 CPU의 설계가 요구된다.

본 논문에서는 CPU의 성능 향상을 위한 분기 예측 기법을 활용한 Spectre 취약점에 대해 분석하였으며 JavaScript를 활용한 웹 사이트에서 발생 가능한 Spectre 취약점에 대해 분석하였다. 향후에는 Spectre와 유사한 취약점들이 사용자 공간에서 커널 메모리에 접근하거나 가상환경에서 escape를 가능하게 하기 때문에 다양한 환경에서의 부채널 공격을 분석하고자 한다.

[참고문헌]

- [1] 김주한, 오경희, 최용제, 김태성, 최두호, “부채널 분석 시스템 기술 동향”, 전자통신동향분석 제28권 제3호, Jun. 2013.
- [2] 이스트시큐리티(ESTsecurity), “인텔 CPU 취약점(Meltdown&Spectre) 분석 및 이스트시큐리티 대응상황”, <https://www.estsecurity.com/securityCenter/news/view/1420>
- [3] Paul Kocher, Anders, et al, “Spectre Attacks: Exploiting Speculative Execution”, ArXiv e-prints, Jan. 2018.
- [4] John Crawford, “The Execution Pipeline of the Intel i486 CPU”, pp.254-258, Feb. 1990.
- [5] 위키백과, “Instruction pipelining”, https://en.wikipedia.org/wiki/Instruction_pipelining
- [6] Kayaalp, Mehmet, et al, “A high-resolution side-channel attack on last-level cache.”, Proceedings of the 53rd Annual Design Automation Conference. ACM, pp.72, 2016.
- [7] Yarom, Yuval, and Katrina Falkner. “FLUSH+RELOAD: a high resolution, low noise, L3 cache side-channel attack.”, 23rd Security Symposium, pp.719-732, 2014.
- [8] Graz University of Technology, “Meltdown and Spectre”, <https://meltdownattack.com/>