레이스 컨디션을 활용한 난수 생성 모듈

서지운[1], 박재열[2], 이경현[3]
[1]부경대학교 컴퓨터인공지능공학부 학부생
[2]부산대학교 정보컴퓨터공학부 학부생
[3]부경대학교 컴퓨터인공지능공학부 교수
jiun20042023@pukyong.ac.kr, woduf0628@pusan.ac.kr, khrhee@pknu.ac.kr

An Exploring of Random Number Generation Using Race Condition

Jiun Seo[1], Jaeyeol Park[2], Kyung-Hyune Rhee[3]
[1]Dept. of Computer and AI Engineering, Pukyong National University
[2]Dept. of Computer Science and Engineering, Pusan National University
[3]Division of Computer and AI Engineering, Pukyong National University

요 약

오늘날 운영체제나 응용프로그램에서 레이스 컨디션으로 인한 문제가 발생하여 공격에 이용하거나 레이스 컨디션을 기반으로 한 공격을 막기 위한 연구가 진행되고 있다. 그러나 레이스 컨디션이 발생할 때 스레드가 자원에 접근하는 매커니즘을 응용한 보호기법과 관련된 연구는 미흡하다. 이에 본 논문에서는 레이스 컨디션이 발생할 때 스레드가 무작위 순서로 자원에 접근하는 점을 이용해 새로운 난수 생성 방식을 제안한다. 또한 이를 난수 생성 알고리즘을 사용하는 랜덤 모듈과 비교하여 더 안정적인 난수 생성 모듈을 개발할 수 있는 가능성에 대해 알아봤다.

1. 서론

제한된 컴퓨팅 자원에 여러 스레드나 프로세서가 한 번에 접근하는 것을 레이스 컨디션이라고 한다. Dirty Cow[1]와 같은 레이스 컨디션을 이용한 공격 연구나 세마포어[2]와 같이 레이스 컨디션을 막기 연구는 활발히 진행됐지만, 레이스 컨디션을 방어 기법에 활용하는 연구는 아직 미흡하다. 이에 레이스 컨디션을 활용하여 기존의 랜덤 모듈을 대체하는 난수 생성 방식을 제안한다.

현재 사용하는 랜덤 모듈은 알고리즘을 통해 난수를 생성하여 시드 값을 알게 되면 난수의 패턴을 구할 수 있다는 문제점이 있다[1]. 이는 시드 값만 알게 되면 무차별 대입을 통해 난수를 쉽게 구할 수 있다는 것을 의미한다. 이에 레이스 컨디션 발생 시 스레드가 자원에 불규칙하게 접근한다는 특성을 이용해 새로운 난수 생성 모듈의 개발 가능성을 제시한다.

2. 레이스 컨디션 난수 생성기

레이스 컨디션 난수 생성기는 파이썬으로 작성되었으며 멀티스레드 프로그래밍이 지원되는 어떤 언어로도 개발이 가능하다.

```
time, threading 모듈 가져오기

함수 num0():                    # 함수 정의
    0부터 500까지 반복:
        0.05초 동안 대기
        숫자 출력

t0 = 스레드(target=num0)     # 스레드 생성
t0.start()                   # 스레드 시작
```

<표 1> 레이스 컨디션 실행 의사 코드

<표 1>의 코드는 0부터 9까지의 숫자 중 하나를 출력하는 함수를 만들고, 이 함수를 별도로 생성된 스레드를 이용해서 레이스 컨디션을 발생시켜 난수를 얻는다. 얻은 난수를 저장하여 이를 원하는 자리수의 암호로 사용한다.

3. 실험

3.1 난수 중복 값 비교

<표 2>은 레이스 컨디션 난수 생성기와 랜덤 모듈에서 생성되는 6자리 암호의 총 개수와 중복 값을 정리한 것이다.

1) https://python.flowdas.com/library/random.html

실행 횟수	중복된 OTP 개수		생성된 OTP 개수
	레이스 컨디션	랜덤 모듈	
1번	14개	0개	833개
10번	399개	70개	8,333개
100번	49,242개	6,658개	83,333개
1000번	729,728개	459,552개	833,333개

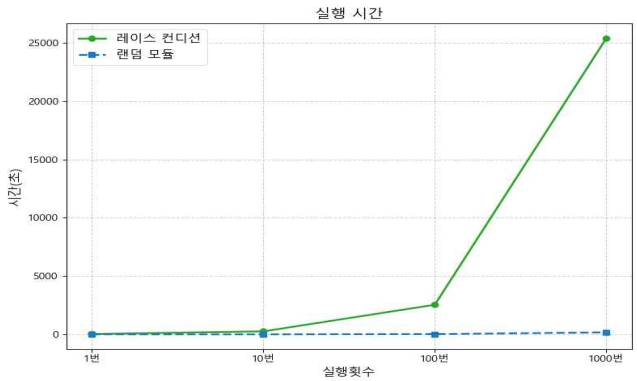
<표 2> 레이스 컨디션과 랜덤 모듈의 결과 값

결과 값에 따르면 코드를 1번 실행하면 5000개의 난수를 얻을 수 있다. 이를 6자리 OTP로 만들면 833개의 값을 얻을 수 있다. 그 중, 레이스 컨디션의 중복된 값을 살펴보면 1번 실행 시, 14개(1.68%)가 생성한다. 추가적으로 10번 실행 시, 399개(4.78%), 100번 실행은 49,242개(59.09%), 그리고 1000번은 729,728개(87.56%)가 생성된다. 이에 비해 랜덤 모듈의 중복된 값은, 1번 실행 시 0개(0%), 10번 실행 시 70개(0.84%), 100번 실행 시 6658개(7.98%), 1000번 실행 시 459,552개(55.14%) 생성한다.

두 결과 값을 비교해보면 1번과 10번 실행 시에서는 레이스 컨디션과 랜덤 모듈이 근소한 차이를 보인다. 하지만 100번과 1000번 실행에서 랜덤 모듈이 레이스 컨디션의 결과 값보다 훨신 적은 개수로 중복값이 발생한다. 하지만 1000번 실행에서 그 차가 100번에 비해 줄어드므로 더 많이 실행하면 더 적은 결과 값 차를 보여줄 것으로 예상한다.

3.2 난수 생성 시간 비교

<그림 1>는 레이스 컨디션 난수 생성기에서 난수를 생성할 때 걸리는 시간과 랜덤 모듈을 활용했을 때를 비교한 그래프이다.



<그림 1> 레이스 컨디션 모듈과 랜덤 모듈의 실행 시간 그래프

레이스 컨디션을 사용 시 발생하는 가장 큰 문제점은 생성 시간이다. 레이스 컨디션은 1번 실행 시 25.391초, 10번은 253.775초, 100번은 2536.132초, 1000번은 25398.850초가 소요된다. 이에 비해 랜덤 모듈을 사용할 때는, 1번 실행 시에는 0.131초, 10번은 0.319초, 100번은 20.260초, 1000번은 167.267초이다. 이는 레이스 컨디션을 통해 난수를 생성했을 때 적어도 125.17배에서 많으면 795.53배 더 시간이 걸린다. 이렇듯 랜덤 모듈에 비해 레이스 컨디션을 사용하는 방식은 실행 시간에 있어 큰 오버헤드가 있는 것이 단점이다.

그러나 이는 스레드의 수를 늘리는 것으로 실행 시간을 단축할 수가 있어 적절한 스레드를 할당하는 것으로 해결이 가능한 문제이다.

4. 결론 및 향후 연구

알고리즘을 통해 난수를 생성하던 기존의 방식과 비교하여, 레이스 컨디션을 통해 불규칙하게 난수를 생성하는 것이 예측이 불가능하여 안정성 측면에서 기존보다 좋은 기술로 보인다.

하지만 레이스 컨디션을 이용하여 난수를 발생시킬 때 발생 되는 중복 값이 기존 모듈로 생성할 때보다 많고, 실행 시간에 있어서 극단적인 차이를 보이는 단점이 있다. 그러나 이는 향후 연구를 통해 중복 값을 줄이고, 최적화를 진행하여 실행 시간을 보다 줄이는 것으로 개선이 충분히 가능하다.

사사(Acknowledgment)

본 연구는 과학기술정보통신부 및 연구개발특구진흥재단의 연구개발특구육성(R&D)-R&D혁신밸리 육성사업의 연구결과로 수행되었음 (2023-DD-RD-0152).

참고문헌

[1] 나성훈 and 석호식, "학부생논문 Dirtycow 취약점 기반 vold 권한탈취를 이용한 안드로이드 사용자 데이터 초기화," in 한국정보과학회 학술발표논문집, 개최지, pp.2037-2039, 2017.

[2] 양희권, 윤기현, 성영락, and 이철훈, "우선순위 역전을 해결하기 위한 세마포어의 구현," in 한국정보과학회 학술발표논문집, 개최지, pp.199-201, 2003.