Binary Search를 이용한 Blind SQL Injection 정리

1. 개요

Binary Search(이진 탐색)는 정렬된 리스트에서 원하는 값을 빠르게 찾는 알고리즘으로, **Blind SQL Injection** 공격 시 패스워드 등의 값을 효율적으로 알아내는 데 활용할 수 있다.

---

2. Binary Search 방식으로 비밀번호 추출

2.1 원리

1. 탐색 범위 설정 (예: ASCII 코드 32~126)
2. 중간값을 기준으로 참/거짓 비교
3. 비교 결과에 따라 탐색 범위를 절반으로 줄여 나감
4. 반복적으로 수행하여 정확한 값을 찾아냄

2.2 예시 데이터베이스

username	password
admin	P@ssword

2.3 쿼리 예시

```
SELECT * FROM users
WHERE username='admin' AND ascii(substr(password, 1, 1)) > 79;
```

- 결과 있음 → 패스워드 첫 글자(P)는 80 → 참
- 다음 단계에서 80~126의 중간값인 103으로 반복 진행

이 과정을 반복하여 전체 문자열을 추출 가능

---

3. Bit 연산을 통한 Blind SQLI

3.1 원리

- ASCII 한 문자는 7비트 → 각 비트를 0/1로 비교하여 알아냄
- 총 7개의 쿼리로 한 문자를 유추 가능
- `bin`, `ord`, `substr` 함수 활용

3.2 함수 설명

- `ord('A')` → 문자 'A'의 ASCII 값 반환 (65)
- `bin(65)` → 2진수 변환 ('1000001')
- `substr()` → 특정 위치의 문자 추출

3.3 예시 쿼리

```
SELECT bin(ord('A'));
-- 결과: '1000001'
```

3.4 비트 단위 Blind SQLI 쿼리 예시

```
SELECT * FROM users
WHERE username='admin' AND substr(bin(ord(password)), 1, 1) = 1;

SELECT * FROM users
WHERE username='admin' AND substr(bin(ord(password)), 2, 1) = 1;

... (계속)
```

3.5 결과 해석

비트 위치	결과
1	참
2	거짓
3	참
4~7	거짓

→ 2진수: `1010000` → 10진수: `80` → 문자: `'P'`

---

4. 마무리

- 이진 탐색 및 비트 비교 알고리즘을 Blind SQLI에 응용하여 빠르고 정확하게 데이터 추출 가능
- 알고리즘을 공격에 적용하면 효율적인 정보 추출 가능

키워드 요약

- **Binary Search**: 탐색 범위를 반씩 줄여 값을 찾는 알고리즘
- **Blind SQL Injection**: 쿼리 결과를 직접 볼 수 없을 때, 조건문의 참/거짓 여부로 정보를 추론
- **bin**, **ord**, **substr**: 문자열의 비트를 추출하거나 비교하기 위한 함수
- **Bit 연산**: 각 비트를 비교하여 한 문자의 값을 알아냄