[WHA-S] ExploitTech: Blind SQL Injection Advanced

1. 들어가며

기초 과정에서 Blind SQL Injection에 대해 간략하게 다루었습니다.
Blind SQL Injection 공격은 데이터베이스에서 특정 데이터를 알아내기 위해 다수의 쿼리를 전송하는 방식입니다.
그러나 방화벽에 의해 IP가 차단될 수 있으며, 데이터의 길이가 길수록 공격 시간이 증가하는 문제점이 있습니다.

이번 강의에서는 **효율적으로 데이터베이스의 내용을 알아내는 방법**을 소개하며,
공격을 수행하기 위한 알고리즘 및 공격 쿼리 작성법을 학습합니다.

---

2. Binary Search (이진 탐색)

이진 탐색은 정렬된 리스트에서 특정 값을 찾기 위한 효율적인 알고리즘입니다.

☑ 이진 탐색 과정

1. 검색 범위를 설정 (예: 0~100)
2. 중간 값(50)을 기준으로 비교
3. 목표 값이 크면 우측 범위(51~100)로 이동, 작으면 좌측 범위(0~49)로 이동
4. 이 과정을 반복하여 목표 값을 찾음

📌 이진 탐색을 활용한 Blind SQL Injection

- `SUBSTRING()` 함수를 활용하여 데이터베이스 값을 한 글자씩 확인 가능
- 예제 쿼리:

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

- 이진 탐색을 반복하면 빠르게 비밀번호 값을 추출할 수 있음

---

3. Bit 연산을 활용한 Blind SQL Injection

ASCII 값은 7비트(0~127)로 표현되므로,
각 비트를 하나씩 확인하는 방식으로 데이터를 추출할 수도 있습니다.

☑ Bit 연산을 활용한 공격 과정

1. **문자의 비트 값을 확인**

```
SELECT bin(ASCII('A'));
```

결과: `1000001` (A의 ASCII 값은 65)

2. **비트 단위로 SQL Injection 수행**

```
SELECT * FROM users
WHERE username='admin'
AND SUBSTRING(BIN(ASCII(password)), 1, 1) = 1;
```

   - 위 쿼리는 비밀번호 첫 글자의 첫 번째 비트가 `1`인지 확인하는 방식
   - 이를 7번 반복하면 한 글자의 모든 비트를 추출할 수 있음

---

4. 마치며

이번 강의에서는 **Blind SQL Injection을 더욱 효율적으로 수행하는 방법**을 학습했습니다.

- **이진 탐색 (Binary Search)**: 쿼리 횟수를 줄여 빠르게 데이터 유출 가능
- **비트 연산 (Bit-wise Search)**: ASCII 값을 비트 단위로 비교하여 정보를 추출

이러한 기법을 사용하면 **기존 Blind SQL Injection보다 훨씬 적은 쿼리 수로 데이터를 유출**할 수 있습니다.
배운 내용을 바탕으로 실습을 진행해보시기 바랍니다. 🚀