
웹 해킹 기초

사이버보안학과 학술소학회  WhoIs

한광석
2017.11.15

CONTENTS

SQL injection	03
---------------	----


XSS	07
-----	----

■ SQL : 데이터를 관리하기 위해 설계된 특수 목적 언어

명령어 종류	명령어	설명
데이터 조작어 (DML : Data Manipulation Language)	SELECT	데이터베이스에 들어 있는 데이터를 조회하거나 검색하기 위한 명령어를 말하는 것으로 RETRIEVE 라고도 한다.
	INSERT	데이터베이스의 테이블에 들어 있는 데이터에 변형을 가하는 종류의 명령어들을 말한다. 데이터 삽입, 수정, 삭제
	UPDATE	
	DELETE	
데이터 정의어 (DDL : Data Definition Language)	CREATE	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어들로 그러한 구조를 생성하거나 변경하거나 삭제하거나 이름을 바꾸는 데이터 구조와 관련된 명령어들을 DDL이라고 부른다.
	ALTER	
	DROP	
	RENAME	
데이터 제어어 (DCL : Data Control Language)	TRUNCATE	데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 명령어를 DCL이라고 한다.
	GRANT	
트랜잭션 제어어 (TCL : Transaction Control Language)	REVOKE	논리적인 작업의 단위를 묶어서 DML에 의해 조작된 결과를 작업단위(트랜잭션) 별로 제어하는 명령어를 말한다.
	COMMIT	
	ROLLBACK	
	SAVEPOINT	

SQL injection





LAB: SQL Injection

Java Source Solution Lesson Plan Hints Restart Lesson

Stage 1
Stage 1: Use String SQL Injection to bypass authentication. Use SQL injection to log in as the boss ('Neville') without using the correct password. Verify that Neville's profile can be viewed and that all functions are available (including Search, Create, and Delete).

Larry Stooge (employee)

Password

Login

Cookies / Parameters

Cookie/s

name	JSESSIONID
value	DA2B06D22D989D3E06B18278FF0994B1
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	312
menu	1100
stage	1
num	

SQL injection




```
try{  
  
    String query = "SELECT * FROM employee WHERE userid = " + userId  
+ " and password = '" + password + "'";  
  
    // System.out.println("Query:" + query);  
}
```

‘or’=‘

SQL injection



 **WEBGOAT**

Database Backdoors

[Java Source](#) [Solution](#) [Lesson Plan](#) [Hints](#) [Restart Lesson](#)

Stage 1: Use String SQL Injection to execute more than one SQL Statement. The first stage of this lesson is to teach you how to use a vulnerable field to create two SQL statements. The first is the system's while the second is totally yours. Your account ID is 101. This page allows you to see your password, ssn and salary. Try to inject another update to update salary to something higher

User ID:

select userid, password, ssn, salary, email from employee where userid=

Cookies / Parameters

Cookie/s

name	JSESSIONID
value	DA2B06D22D989D3E06B18278FF0994B1
comment	
domain	
maxAge	-1
path	
secure	false
version	0
httpOnly	false

Parameters

scr	274
menu	1100
stage	
num	

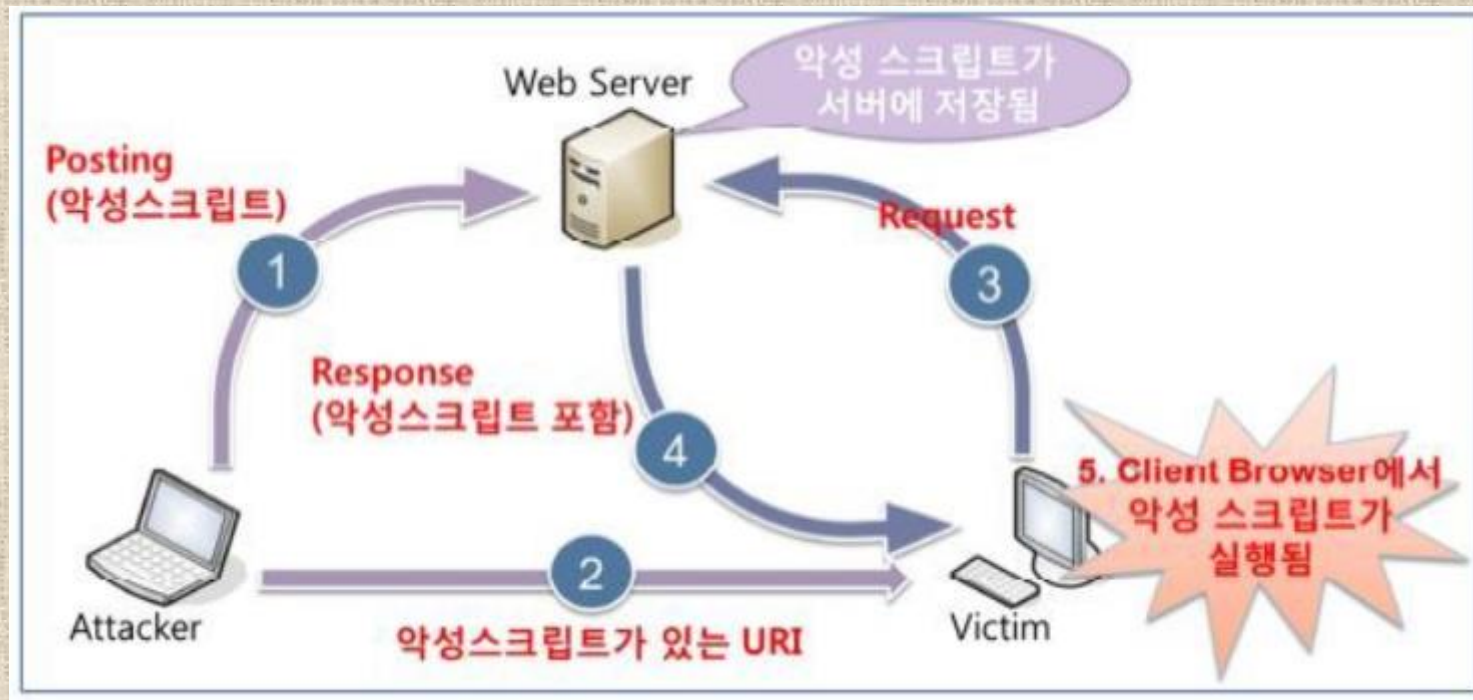
101;update employee Set salary = '70000' where userid = 101₆

XSS



- Stored XSS
- Reflect XSS
- CSRF

Stored XSS



Stored XSS

Stored XSS Attacks

[Java Source](#) [Solution](#) [Lesson Plan](#) [Hints](#) [Restart Lesson](#)

It is always a good practice to scrub all input, especially those inputs that will later be used as parameters to OS commands, scripts, and database queries. It is particularly important for content that will be permanently stored somewhere in the application. Users should not be able to create message content that could cause another user to load an undesirable page or undesirable content when the user's message is retrieved.

Title:

Message:

Submit

Message List

Stored XSS



Title:

Message:

[Java Source](#)[Solution](#)[Lesson Plan](#)[Hints](#)[Res](#)

localhost:8080 내용:

script

확인

It is always a good practice to scrub all input for dangerous commands, scripts, and database queries. It is important to ensure that somewhere in the application. Users should not be able to create message content that could cause another user an undesirable page or undesirable content when the user's message is retrieved.

*** Congratulations. You have successfully completed this lesson.**

Title:

Message:

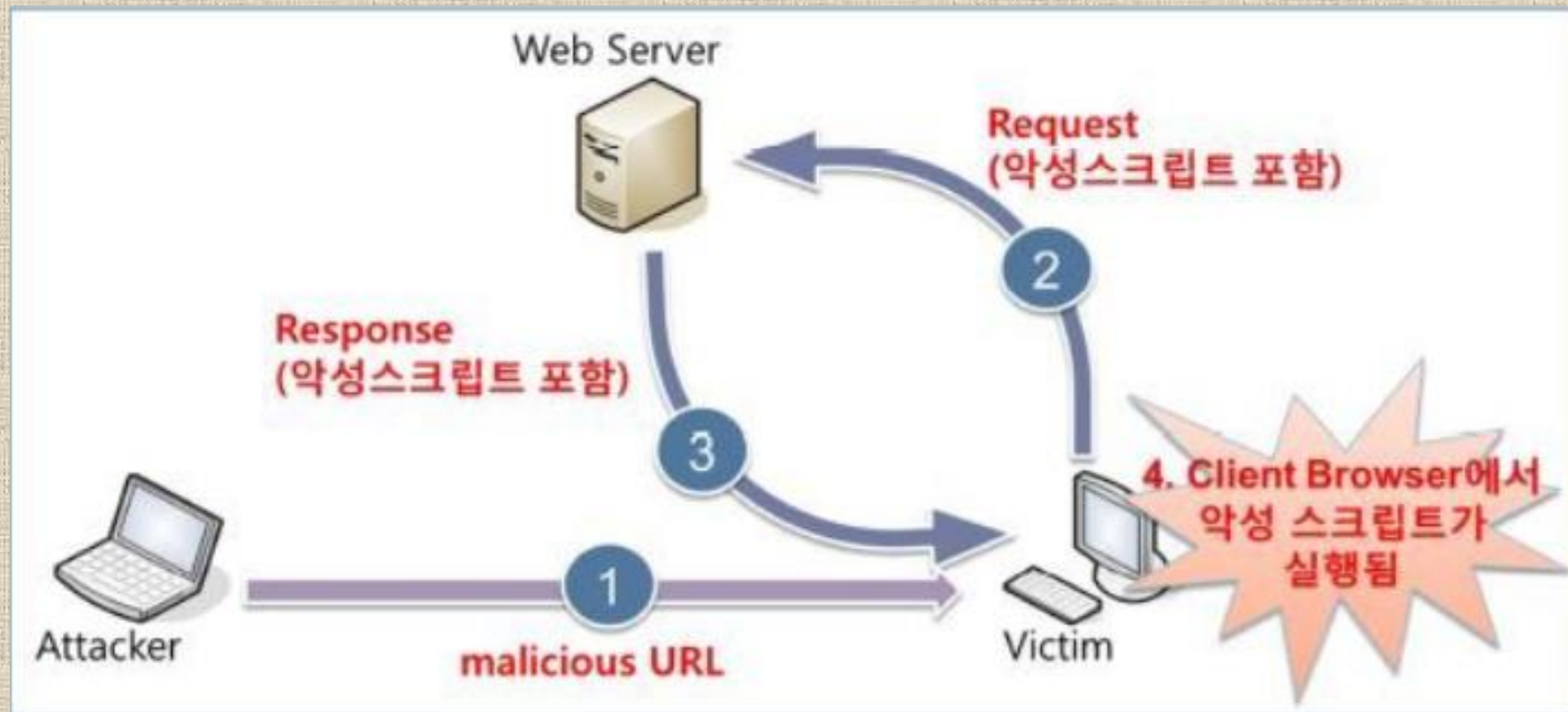
Message Contents For: h

Title: h

Message: hi

Posted by: guest

Reflect XSS



Reflect XSS

Reflected XSS Attacks

[Java Source](#) [Solution](#) [Lesson Plan](#) [Hints](#) [Restart Lesson](#)

It is always a good practice to validate all input on the server side. XSS can occur when unvalidated user input is used in an HTTP response. In a reflected XSS attack, an attacker can craft a URL with the attack script and post it to another website, email it, or otherwise get a victim to click on it.

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	<input type="text" value="1"/>	\$0.00
Dynex - Traditional Notebook Case	27.99	<input type="text" value="1"/>	\$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	<input type="text" value="1"/>	\$0.00
3 - Year Performance Service Plan \$1000 and Over	299.99	<input type="text" value="1"/>	\$0.00

The total charged to your credit card: \$0.00

[UpdateCart](#)

Enter your credit card number:

Enter your three digit access code:

[Purchase](#)

Reflect XSS

Reflected XSS

localhost:8080 내용:
bang

확인

[Java Source](#) [Solution](#) [Lesson Plan](#) [Hints](#) [Res](#)

It is always a good practice to validate all input on the server side. XSS can occur when unvalidated user input is used in an HTTP response. In a reflected XSS attack, an attacker can craft a URL with the attack script and post it to another website, email it, or otherwise get a victim to click on it.

Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tilting Surface - Cherry	69.99	<input type="text" value="1"/>	\$0.00
Dynex - Traditional Notebook Case	27.99	<input type="text" value="1"/>	\$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	<input type="text" value="1"/>	\$0.00
3 - Year Performance Service Plan \$1000 and Over	299.99	<input type="text" value="1"/>	\$0.00

The total charged to your credit card: \$0.00 [UpdateCart](#)

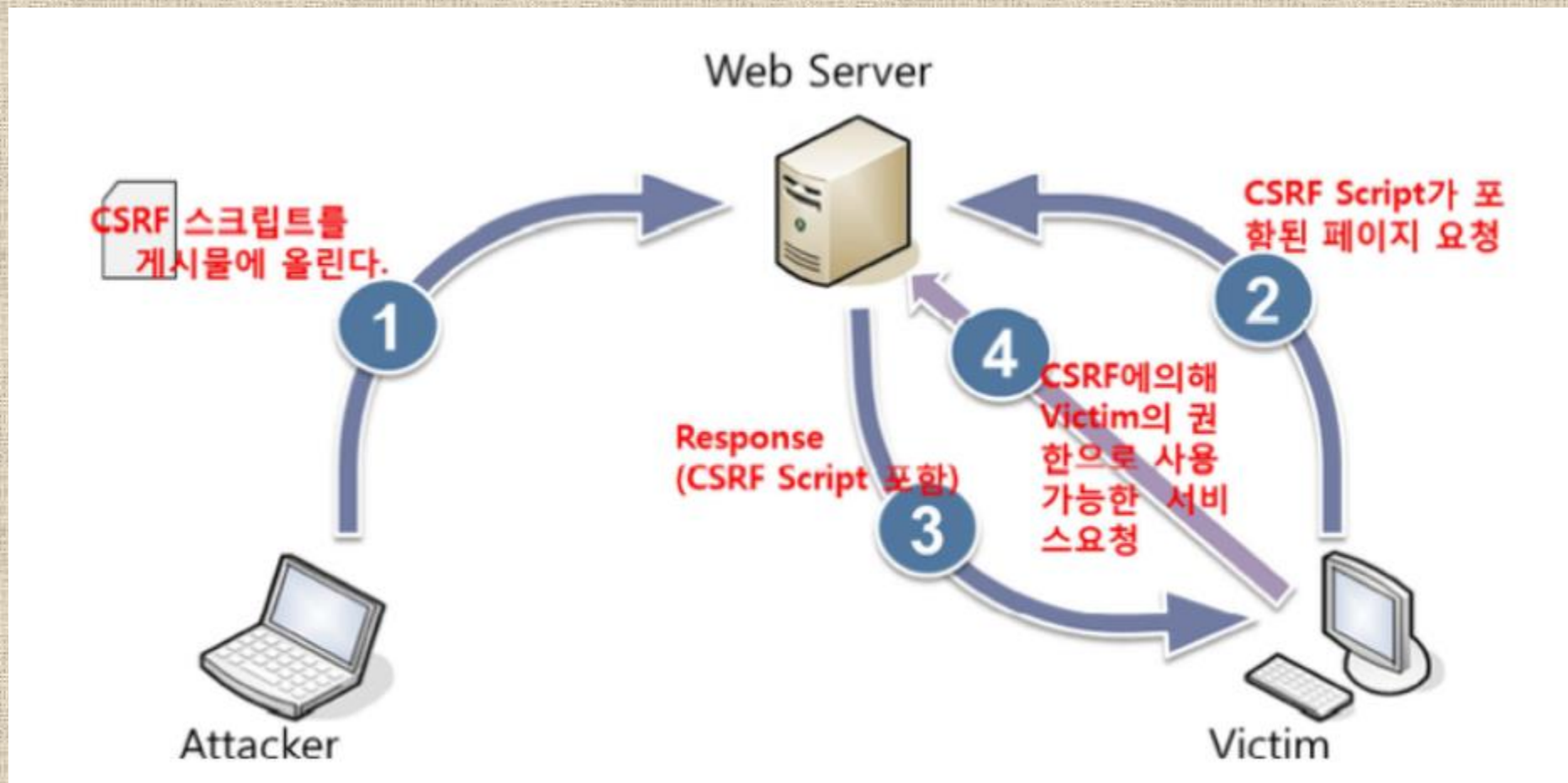
Enter your credit card number:

Enter your three digit access code:

[Purchase](#)

111'><script>alert('bang')</script><font size = '4

CSRF



CSRF



Cross Site Request Forgery (CSRF)

[Java Source](#)[Solution](#)[Lesson Plan](#)[Hints](#)[Restart Lesson](#)

Your goal is to send an email to a newsgroup. The email contains an image whose URL is pointing to a malicious request. In this lesson the URL should point to the "attack" servlet with the lesson's "Screen" and "menu" parameters and an extra parameter "transferFunds" having an arbitrary numeric value such as 5000. You can construct the link by finding the "Screen" and "menu" values in the Parameters inset on the right. Recipients of CSRF emails that happen to be authenticated at that time will have their funds transferred. When this lesson's attack succeeds, a green checkmark appears beside the lesson name in the menu on the left.

Title:

Message:

Submit

``

XSS 와 CSRF

	XSS	CSRF
공격 수행 지점	클라이언트	서버
기능 구현	공격자가 Script를 이용하여 직접 구현	서버에서 제공하는 기능을 도용
Script 사용여부	반드시 Script가 사용가능해야함	Script를 사용할 수 없어도 공격 가능
공격 시 준비사항	XSS 취약점만 발견 후 즉시 사용 가능	공격하고자 하는 Request/Response의 로직을 분석해야 함
공격감지 가능여부	Stored / Reflective 모두 감지 가능	구분할 수 없음

Q & A

