CSRF(Cross-Site Request Forgery)

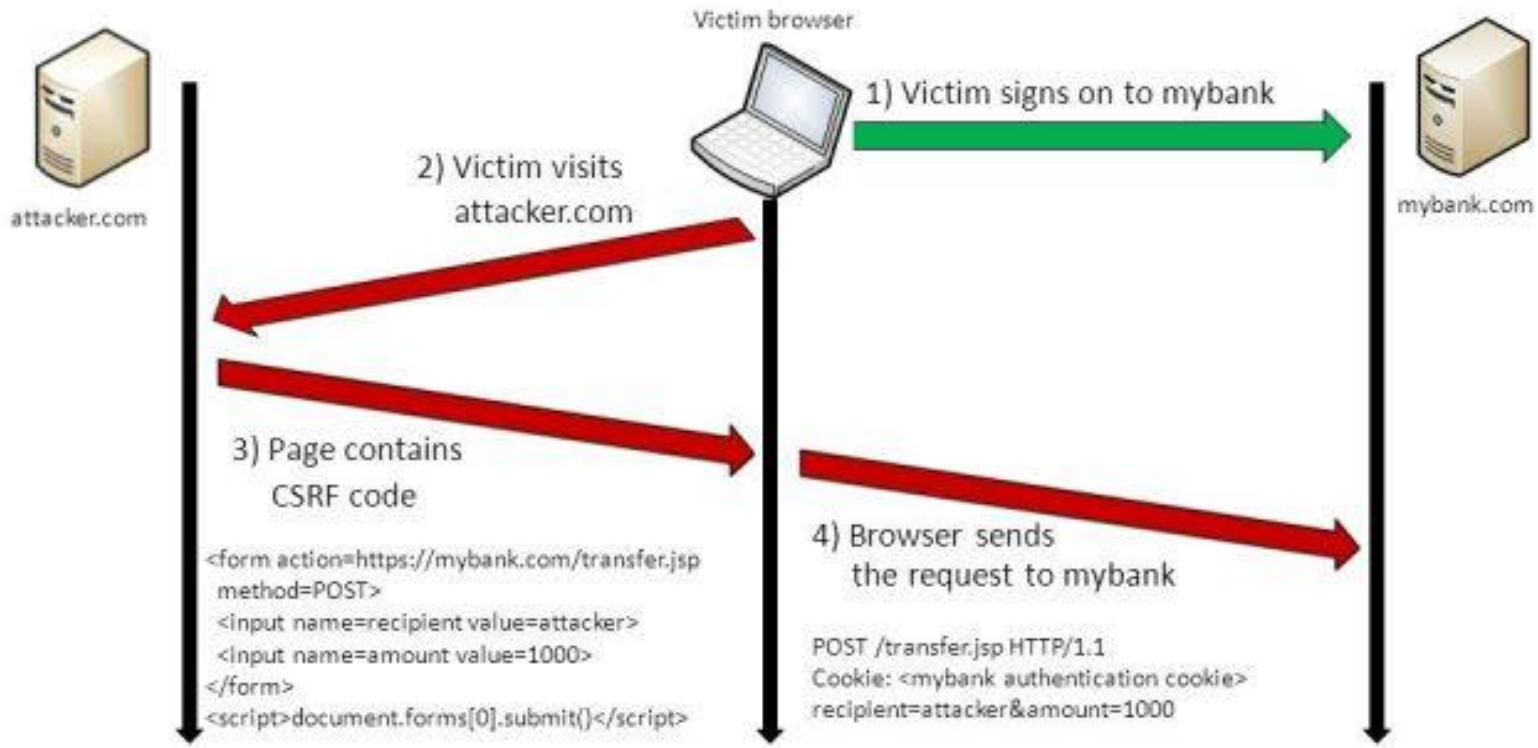
- 사이트 간 요청 위조

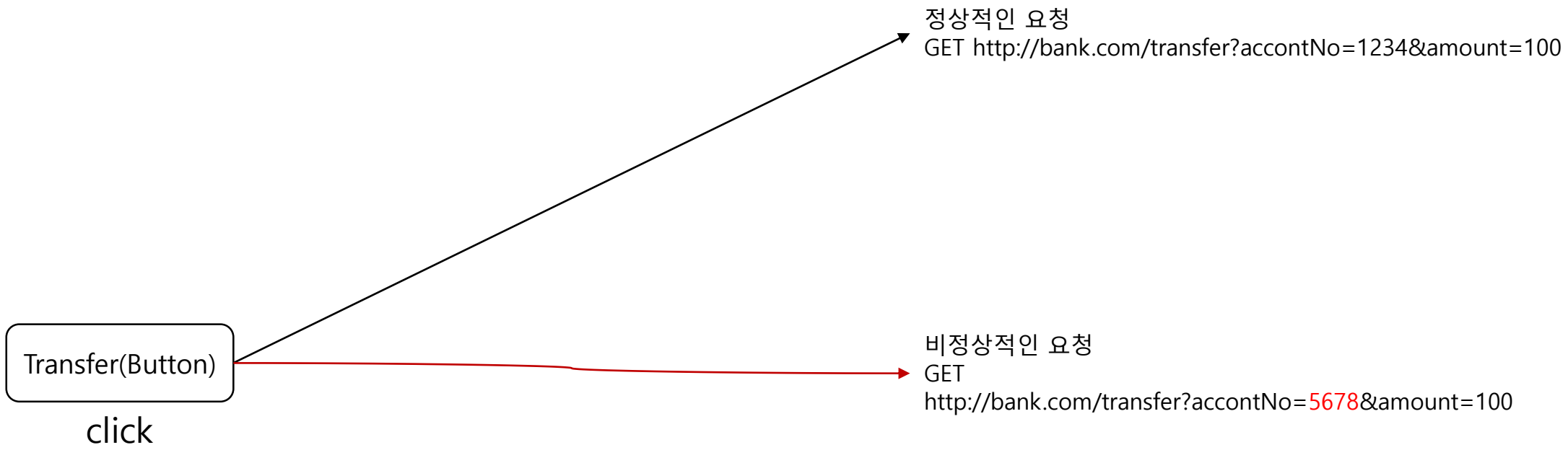
1) 공격자가 악의적인 의도가 담긴 게시글을 올린다.

2) 이미 인증된 사용자가 공격자의 게시글을 읽을 때, 공격자가 의도한 요청을 인증된 사용자의 브라우저에서 보내게 된다.

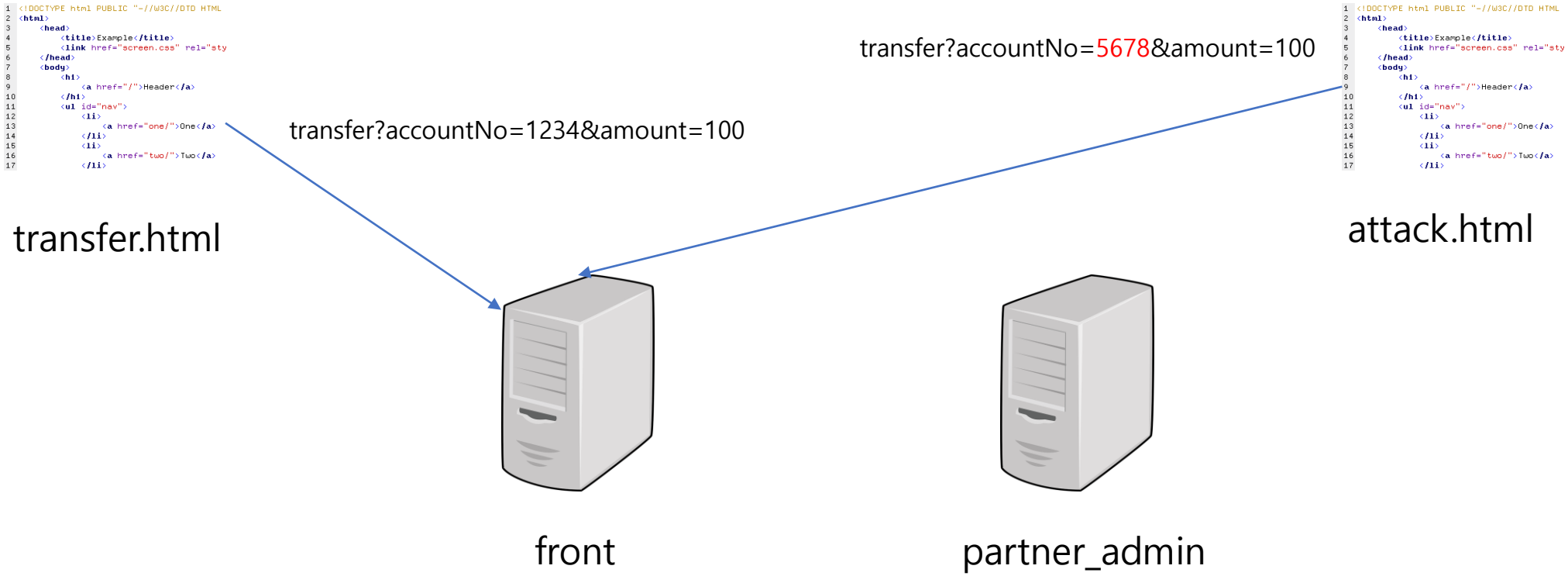
3) 이미 인증된 사용자가 보낸 요청이기 때문에 서버는 이 요청을 수락하고 실행하게 된다.

Cross-Site Request Forgery (CSRF)




정상적인 요청
GET http://bank.com/transfer?accontNo=1234&amount=100
Transfer(Button)
click
비정상적인 요청
GET
http://bank.com/transfer?accontNo=5678&amount=100

test


transfer?accountNo=1234&amount=100
transfer?accountNo=5678&amount=100
transfer.html
attack.html
front
partner_admin

1)frontServer

```java
@Slf4j
@RestController

public class BankController {
    @RequestMapping(value = "/transfer",method = RequestMethod.GET)
    public String tranfer(@RequestParam int accountNo, @RequestParam int amount){
        log.info("accountNo : {},amount : {}",accountNo,amount);
        return "OK";
    }

    @RequestMapping(value = "/transfer",method = RequestMethod.POST)
    public String transfer2( int accountNo, int amount){
        log.info("accountNo : {},amount : {}",accountNo,amount);
        return "OK";
    }

}
```

BankController.java

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <title>TEST</title>
</head>
<body>
    <h1>CSRF test on Origin</h1>
    <a href="transfer?accountNo=1234&amount=100">Transfer Money to Friend</a>

    <form action="/transfer" method="post">
        <label>Account Number</label>
        <input name="accountNo" type="number"/>
        <label>Account</label>
        <input name="amount" type="number"/>

        <input type="submit">
    </form>
</body>
</html>
```

transfer.html

CSRF test on Origin

Transfer Money to Friend

Account Number 1234 Account 100 제출

http://localhost:8080/transfer.html

```
2018-09-23 09:56:49.841  INFO 1540 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : FrameworkServlet 'dispatcherServlet': initialization started
2018-09-23 09:56:49.865  INFO 1540 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : FrameworkServlet 'dispatcherServlet': initialization completed in 24 ms
2018-09-23 09:56:56.164  INFO 1540 --- [nio-8080-exec-2] c.study.front.controller.BankController  : accountNo : 1234,amount : 100
```

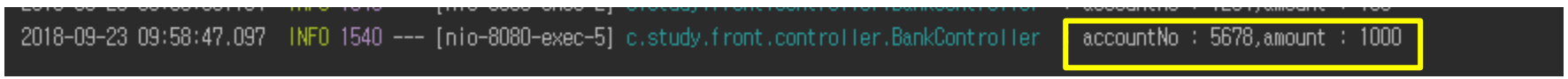
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <title>Title</title>
</head>
<body>
    <a href="http://localhost:8080/transfer?accountNo=5678&amount=100">Transfer Money to Friend</a>

    <form action="http://localhost:8080/transfer" method="POST">
        <input name="accountNo" type="hidden" value="5678"/>
        <input name="amount" type="hidden" value="1000"/>
        <input name="accountNo1" type="text" value="1234"/>
        <input name="amount1" type="text" value="1000"/>
        <input type="submit" value="Show Kittens Picture">
    </form>
</body>
</html>
```

attack.html

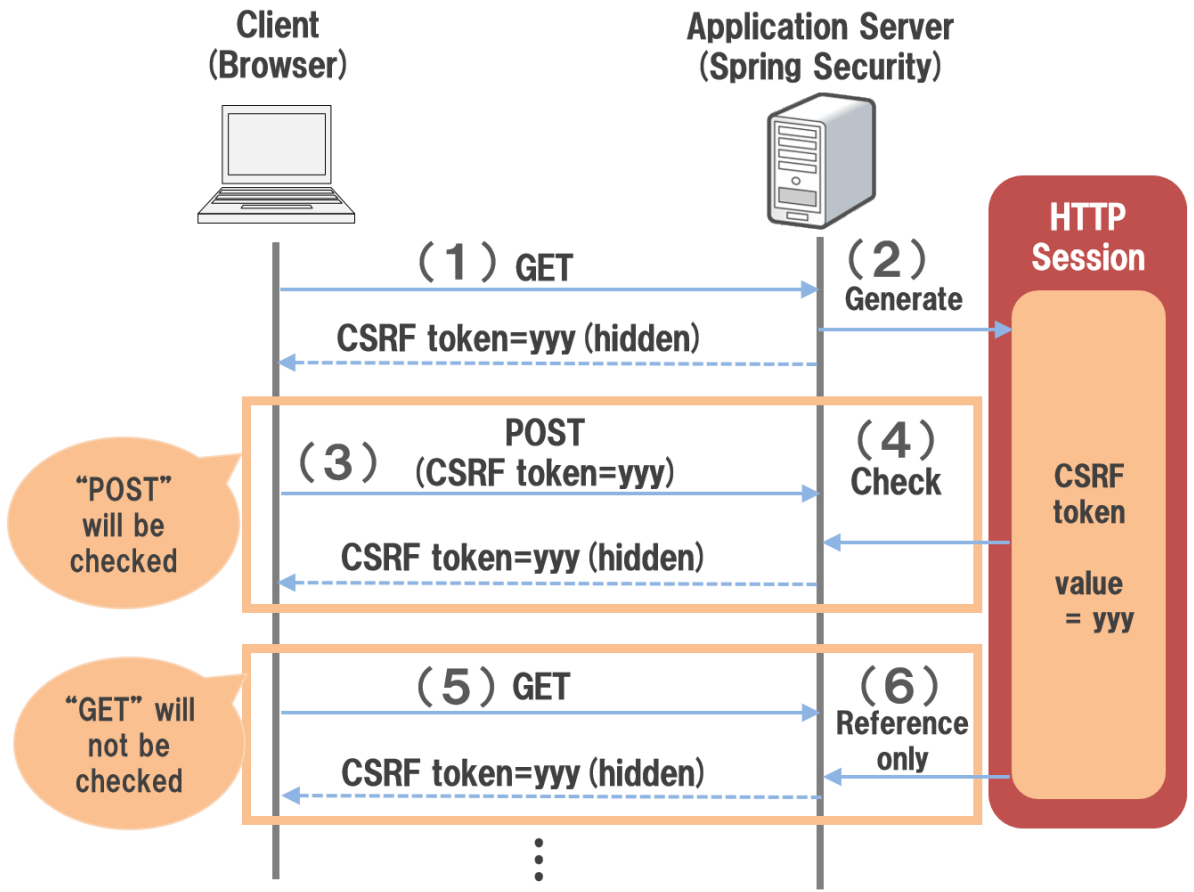
Transfer Money to Friend

1234 | 1000 | Show Kittens Picture

http://localhost:7075/attack.html



csrf 방지법 – 일회성 토큰 발급



http://localhost:8080/transfer?accountNo=5678&amount=500 +CSRF token

- 어제 설정했던 http.csrf().disable()을 -> http.csrf() 라고 설정
- html파일에서는 일회성 토큰의 값을 받아줄 수 있는 태그 추가

An easier approach is to use the csrfInput tag from the Spring Security JSP tag library.

If you are using Spring MVC `<form:form>` tag or Thymeleaf 2.1+ and are using `@EnableWebSecurity`, the `CsrfToken` is automatically included for you (using the `CsrfRequestDataValueProcessor`).

```
// 이 부분 알아보기!!
http.csrf();
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"></head>
    <title>Title</title>
</head>
<body>
login
<form action="/api/process/login" method="post">
    id : <input type="text" name="id"> <br>
    pw : <input type="text" name="pw"> <br>
    토큰 :<input type="hidden" th:name="${_csrf.parameterName}" th:value="${_csrf.token}" />
    <button type="submit">전송</button>
</form>
</body>
</html>
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"></head>
    <title>Title</title>
</head>
<body>
login
<form action="/api/process/login" method="post">
    id : <input type="text" name="id"> <br>
    pw : <input type="text" name="pw"> <br>
    토큰 :<input type="hidden" name="_csrf" value="8acb9173-180a-44d3-914f-650adeb5b13b" />
    <button type="submit">전송</button>
</form>
</body>
</html>
```

login.html