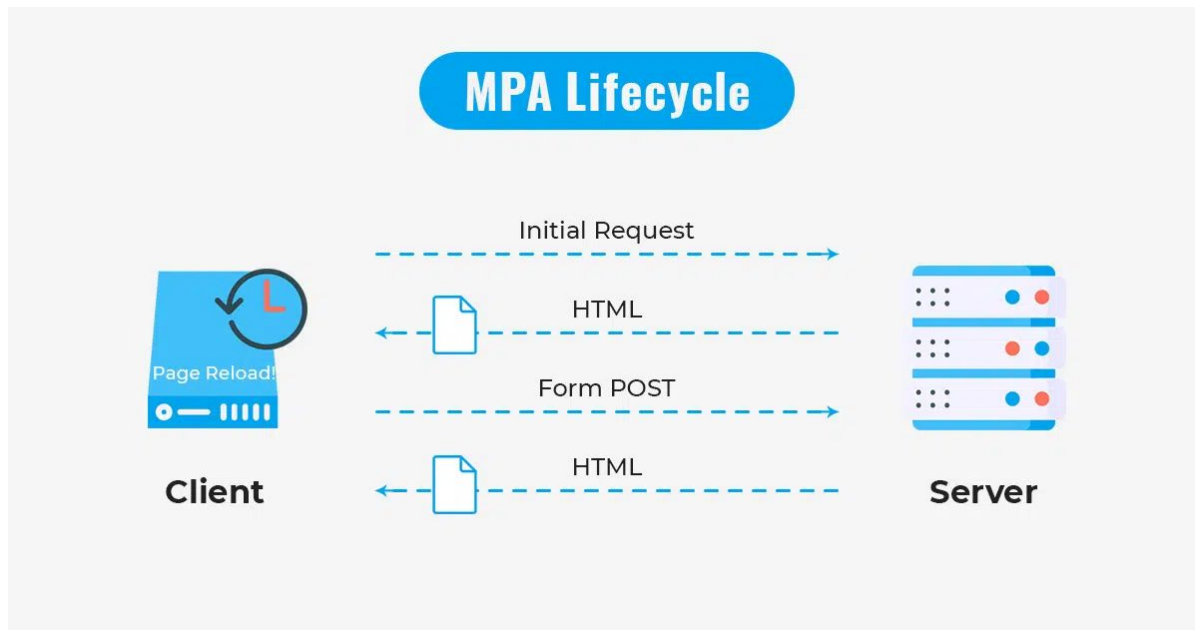




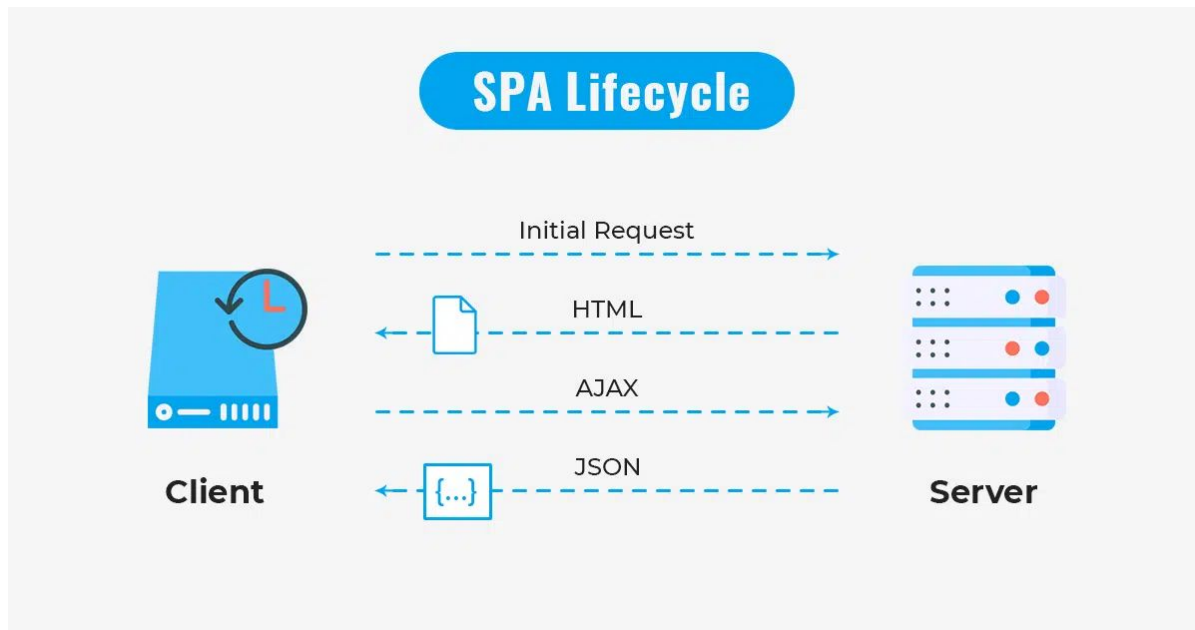
Ajax 통신 과정에서 발생가능한 취약점

bee-box를 통한 실습

MPA와 SPA - Multiple Page Application



MPA와 SPA - Single Page Application





MPA와 SPA

- MPA와 SPA의 차이는 SSR(Server side Rendering)과 CSR(Client side Rendering)의 차이
- 각 페이지를 동기 방식으로 전체 페이지를 렌더링 해서 보여주는 MPA와 달리 SPA는 비동기 방식으로 페이지의 일부를 불러와 갱신
- 요즘에는 사용자의 집중도를 고려하여 SPA 방식의 웹사이트 구성을 선호하나, 규모가 큰 사이트의 경우 MPA를 선호하는 경향이 보임.



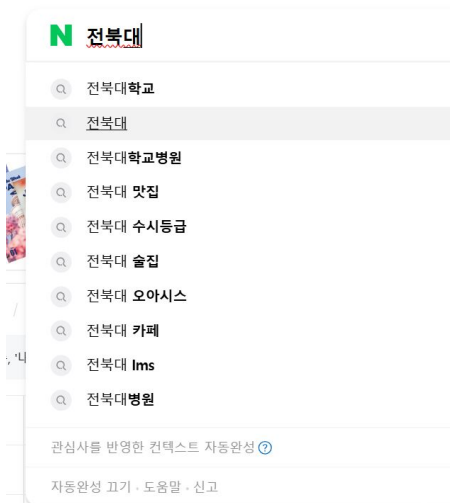
Ajax

- Asynchronous Javascript And Xml로 SPA의 기반이되는 기능

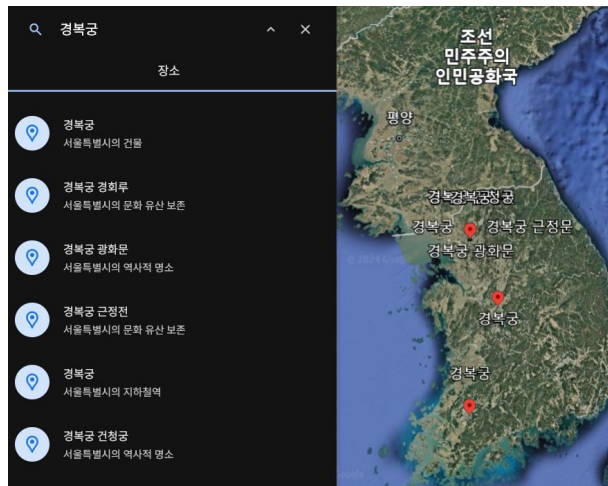
(Ajax 자체는 프레임워크 같은 것이 아닌 구현 방식을 의미한다.)

- 보통 JSON이나 XML과 같은 형태로 필요한 데이터만을 받아 페이지를 갱신한다.
- 전북대 웹사이트에서 주로 사용되는 jQuery 또한 Ajax를 이용하기 위한 도구이다.

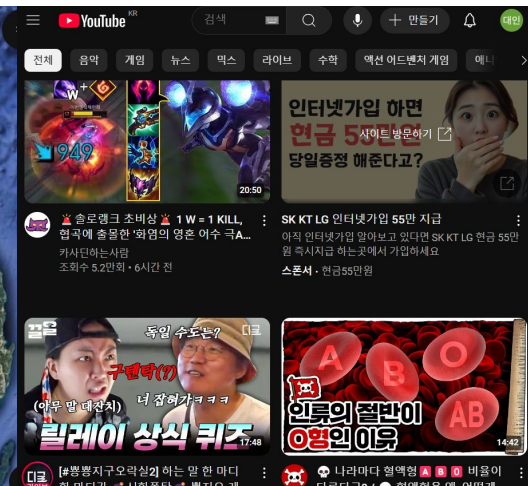
Ajax 적용 사례



네이버 검색어 자동완성



구글 어스



유튜브 무한스크롤 등

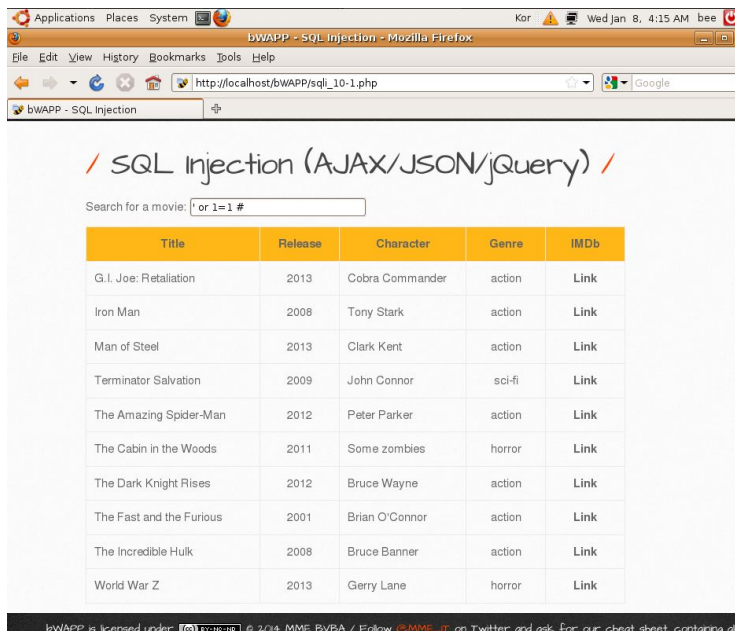
Bee box Ajax 실습 1- SQL Injection

/ SQL Injection (AJAX/JSON/jquery) /

Search for a movie:

Title	Release	Character	Genre	IMDb
Iron Man	2008	Tony Stark	action	Link

Bee box Ajax 실습 1- SQL Injection



SQL Injection (AJAX/JSON/jQuery) /

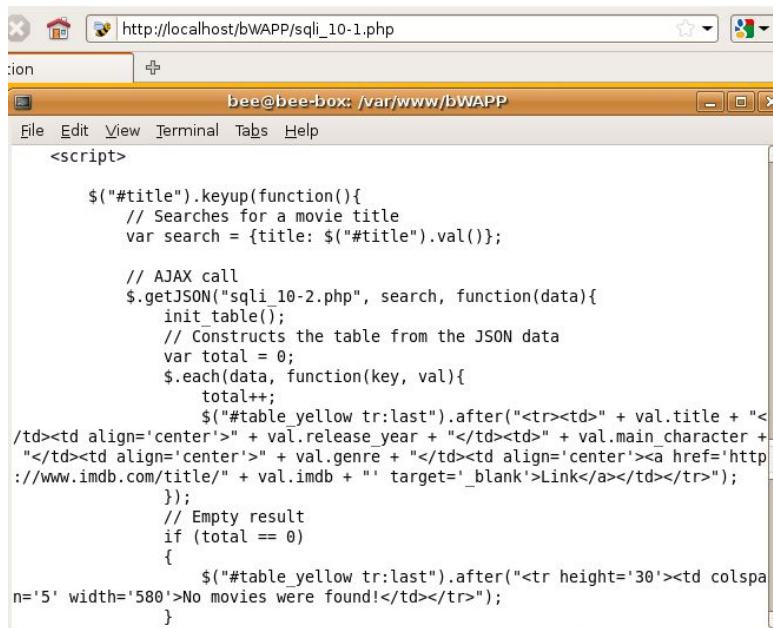
Search for a movie:

Title	Release	Character	Genre	IMDb
G.I. Joe: Retaliation	2013	Cobra Commander	action	Link
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
Terminator Salvation	2009	John Connor	sci-fi	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
The Cabin in the Woods	2011	Some zombies	horror	Link
The Dark Knight Rises	2012	Bruce Wayne	action	Link
The Fast and the Furious	2001	Brian O'Connor	action	Link
The Incredible Hulk	2008	Bruce Banner	action	Link
World War Z	2013	Gerry Lane	horror	Link

bWAPP is licensed under [CC BY-NC-SA](#) © 2014 MME BVBA / Follow [@MME_U](#) on Twitter and ask for our cheat sheet containing all

실행 결과: 일반적으로 수행할 수 있는 SQL injection 공격은 모두 수행 가능.

Bee box Ajax 실습 1- SQL Injection



The image shows a web browser window at the top with the address bar displaying `http://localhost/bWAPP/sqli_10-1.php`. Below the browser is a terminal window titled `bee@bee-box: /var/www/bWAPP`. The terminal contains the following JavaScript code:

```
<script>

$("#title").keyup(function(){
    // Searches for a movie title
    var search = {title: $("#title").val()};

    // AJAX call
    $.getJSON("sqli_10-2.php", search, function(data){
        init_table();
        // Constructs the table from the JSON data
        var total = 0;
        $.each(data, function(key, val){
            total++;
            $("#table yellow tr:last").after("<tr><td>" + val.title + "<
            /td><td align='center'>" + val.release_year + "</td><td>" + val.main_character +
            "</td><td align='center'>" + val.genre + "</td><td align='center'><a href='http
            ://www.imdb.com/title/" + val.imdb + "' target='_blank'>Link</a></td></tr>");
        });
        // Empty result
        if (total == 0)
        {
            $("#table yellow tr:last").after("<tr height='30'><td colspa
            n='5' width='580'>No movies were found!</td></tr>");
        }
    });
});
```



Bee box Ajax 실습 1- SQL Injection

```
switch ($_COOKIE["security_level"])
{
    case "0" :
        $data = no_check($data);
        break;

    case "1" :
        $data = sqli_check_1($data);
        break;

    case "2" :
        $data = sqli_check_2($data);
        break;

    default :
        $data = no_check($data);
        break;
}
```



Bee box Ajax 실습 1- SQL Injection

```
function sqli_check_1($data)
{
    return addslashes($data);
}

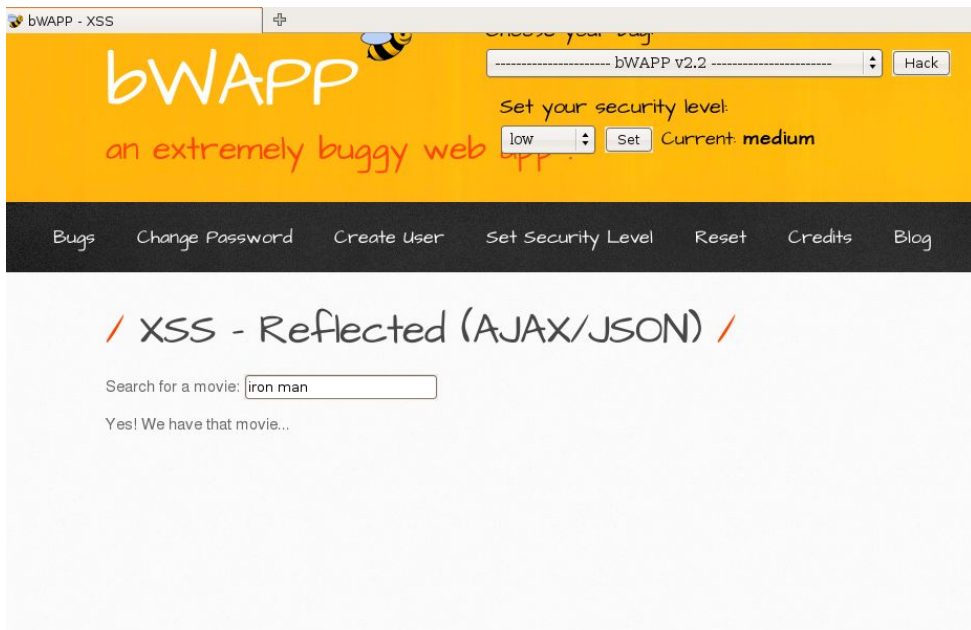
function sqli_check_2($data)
{
    return mysql_real_escape_string($data);
}
```



정말 Ajax에서 취약한가?

- SQL 인젝션의 경우 고전적인 웹에서도 쿼리 로직에 따라 발생할 수 있는 취약점임.
- json, 혹은 xml을 이용하여 데이터를 전달하는 방식에서 개발자의 실수(입력값 검증 누락)로 인해 발생하는 문제 말고는 Ajax 자체의 취약점이라고 보기는 어렵다.

Bee box Ajax 실습 2- XSS-Reflected



Bee box Ajax 실습 2- XSS-Reflected

/ XSS - Reflected (AJAX/JSON) /

Search for a movie:

??? Sorry, we don't have that movie :(



Bee box Ajax 실습 2- XSS-Reflected

```
function process()
{
    // Proceeds only if the xmlhttp object isn't busy
    if(xmlHttp.readyState == 4 || xmlhttp.readyState == 0)
    {
        // Retrieves the movie title typed by the user on the form
        // title = document.getElementById("title").value;
        title = encodeURIComponent(document.getElementById("title").value);

        // Executes the 'xss_ajax_1-2.php' page from the server
        xmlhttp.open("GET", "xss_ajax_2-2.php?title=" + title, true);
        // Defines the method to handle server responses
        xmlhttp.onreadystatechange = handleServerResponse;
        // Makes the server request
        xmlhttp.send(null);
    }
    else
        // If the connection is busy, try again after one second
        setTimeout("process()", 1000);
}
```

Bee box Ajax 실습 2- XSS-Reflected

XMLHttpRequest



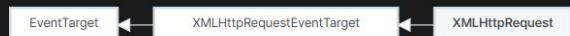
Baseline Widely available



Invalid slug for templ/sidebar: XMLHttpRequest

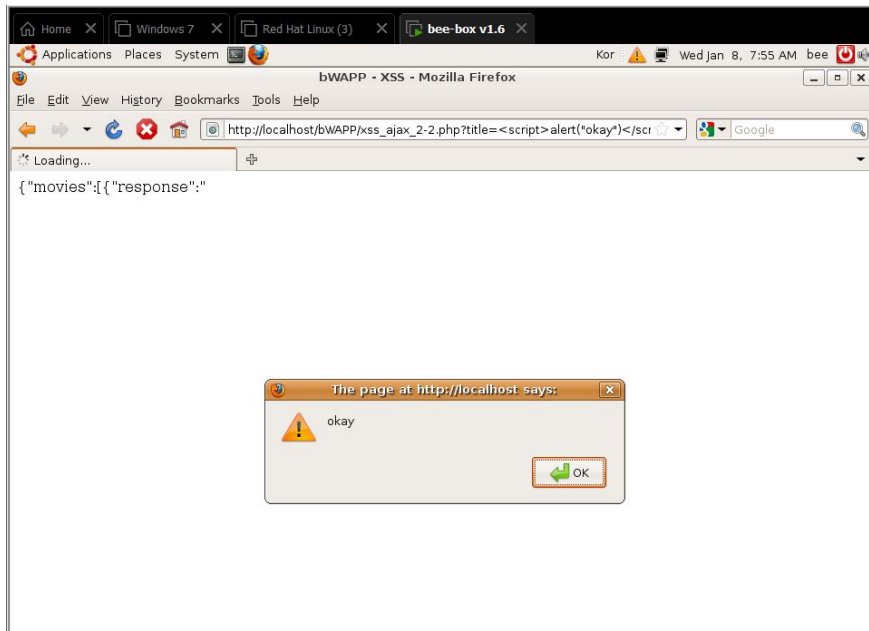
`XMLHttpRequest` (XHR) 객체는 서버와 상호작용할 때 사용합니다. XHR을 사용하면 페이지의 새로고침 없이도 URL에서 데이터를 가져올 수 있습니다. 이를 활용하면 사용자의 작업을 방해하지 않고 페이지의 일부를 업데이트할 수 있습니다.

`XMLHttpRequest` 는 [AJAX](#) 프로그래밍에 많이 사용됩니다.



이름에 `XML` 이 들어가긴 하지만, `XMLHttpRequest` 은 XML 뿐만 아니라 모든 종류의 데이터를 가져올 수 있습니다.

Bee box Ajax 실습 2- XSS-Reflected





Ajax에서 XSS 예방

- XSS 취약점 또한 입력값 검증 문제로 일어난다.
- `htmlspecialchars()` 함수로 입력값 검증이 이루어지면 문제를 해결할 수 있다.



결론

- Ajax에서 발생가능한 취약점은 입력값 검증으로 쉽게 예방 가능하다.
- Ajax를 이용한 비동기 방식의 통신은 클라이언트 상에서 요청하더라도 이 값에 대한 검증은 서버 측에서 해야한다.