

Web Penetration test

윤상현

2018-06-03

## 목차

1. 개요
2. SQL Injection
3. XSS
4. LFI
5. CSRF

## 개요

IT가 나날이 발전하고 있다. 컴퓨터가 냉장고, 자동차 등 사물을 가리지 않고 내장되어 있고, AI가 스스로 전화를 걸고 예약을 하는가 하면, 드론이 직접 물건을 배달한다. 중국에서는 13억명의 얼굴을 3초안에 인식하는 기술이 상용화 되어있고, 각종 휴대기기에서는 생체인증을 통해 개인을 구별하는 시대이다.

기술이 발전할수록 보안 문제는 더욱 중요한 문제로 대두되고 있다. 최근 절대 해킹되지 않는다는 가상화폐가 해킹되어 피해 사례가 생겼고, 멕시코에서는 은행 다섯 곳이 해킹당해 거액의 현금이 인출되는 피해가 생겼다. 이 같은 침해 사례가 늘어감에 따라 보안 관련 시장 또한 성장세를 보이고 있다. 기업과 공공기관의 보안의식이 점점 강화되고 있다는 증거이며 이는 매우 고무적인 현상이다.

기업 차원의 보안은 생각보다 간단하다. 보안이 중요하다는 의식이 있으면 전문 인력을 고용하여 컨설팅하고 취약점에 대한 대응을 하면 된다. 하지만 피해자가 개인이라면? 일단 개인에 대한 해킹이 발생하면 대처하기가 무척 곤란하다. 하지만 기업에 대한 해킹과 달리 개인에 대한 해킹은 그 대비가 쉬운 편이다. 즉 침해대응은 어렵지만 개인이 조심한다면 보안사고 대비는 충분히 가능하다는 것이다.

본 테스트에서는 개인에게 취약한 해킹기법에 대해 소개하고, 실제 환경에서 해킹 테스트를 진행하며, 취약점을 보완하는 방법에 대해 설명한다.

## 테스트 환경

Penetration Tester area : **Kali Linux**

Server Area : **Mysql 5.7, Word Press 3.8.1**

## SQL injection(SQL 주입)

보호나라에서는 SQL Injection을 아래와 같이 정의하고 있다.

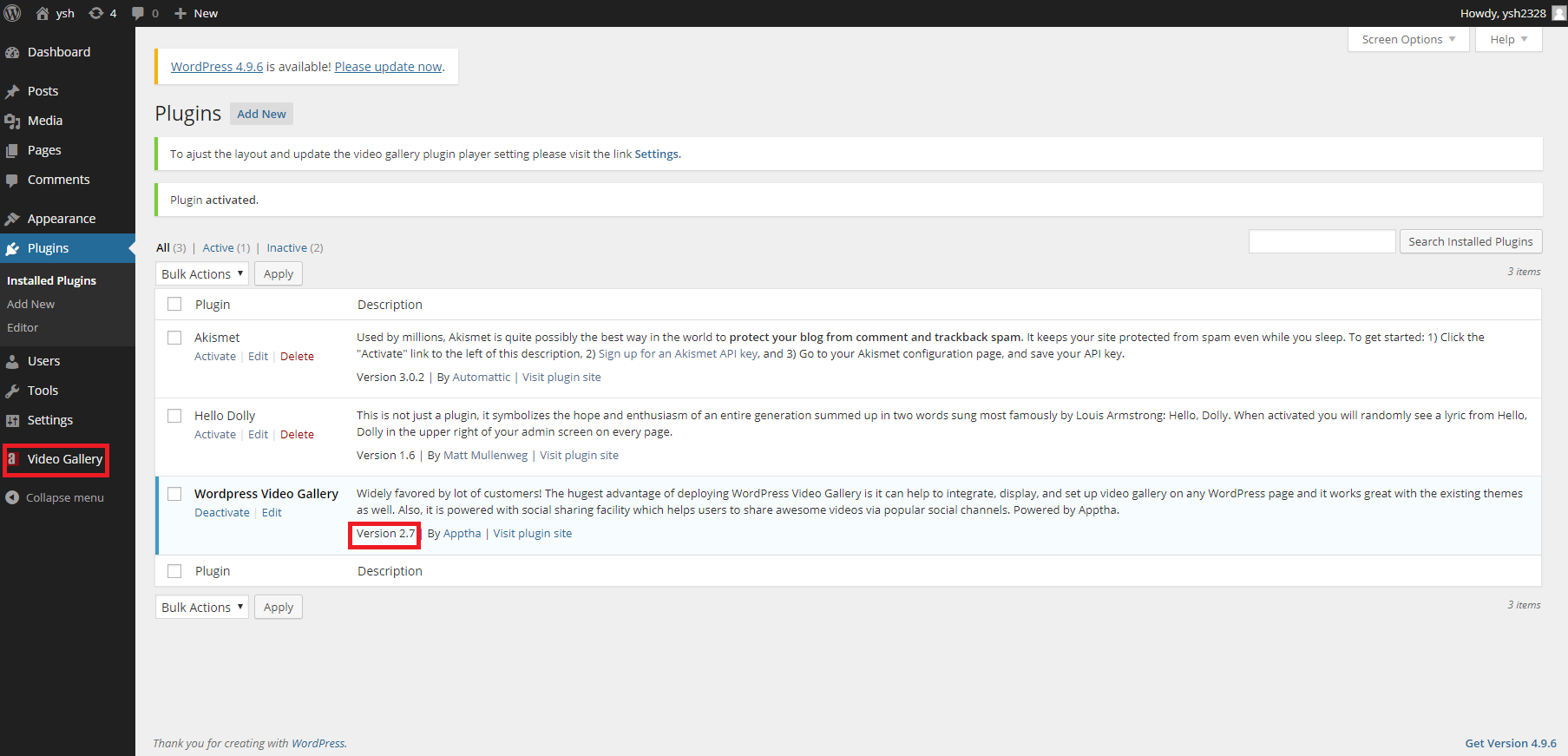
현재 대부분의 웹사이트들은 사용자로부터 입력받은 값을 이용해 데이터 베이스 접근을 위한 SQL Query를 만들고 있다.

사용자 로그인 과정을 예로 들면, 사용자가 유효한 계정과 패스워드를 입력했는지 확인하기 위해 사용자 계정과 패스워드에 관한 SQL Query를 만듭니다. 이때 **SQL Injection 기법을 통해서 정상적인 SQL Query를 변조할 수 있도록 조작된 사용자 이름과 패스워드를 보내 정상적인 동작을 방해할 수 있습니다. 이러한 비정상적인 SQL Query를 이용해 다음과 같은 공격이 가능**합니다.

* *사용자 인증을 비정상적으로 통과할 수 있다.*
* *데이터베이스에 저장된 데이터를 임의로 열람할 수 있다.*
* *데이터베이스의 시스템 명령을 이용하여 시스템 조작이 가능하다.*

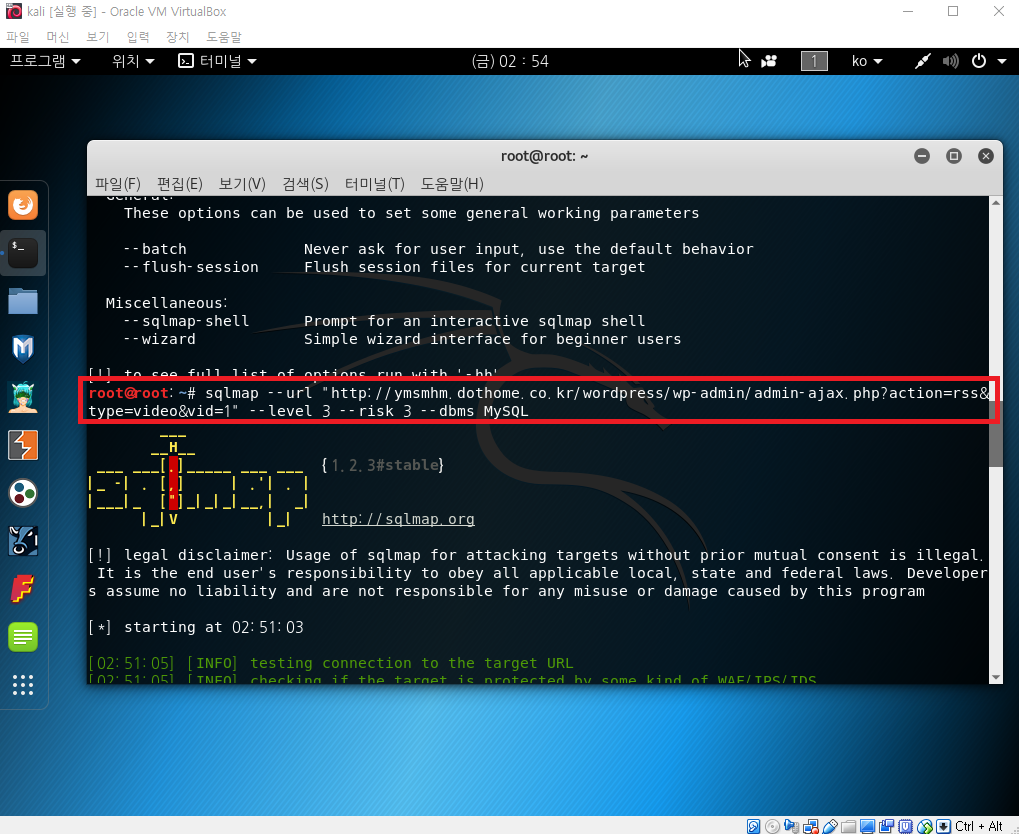
이러한 취약점을 SQL Injection 취약점이라고 하며, 사용자가 데이터 입력이 가능한 수많은 웹페이지 상에 이러한 취약점이 존재할 수 있습니다.

시연

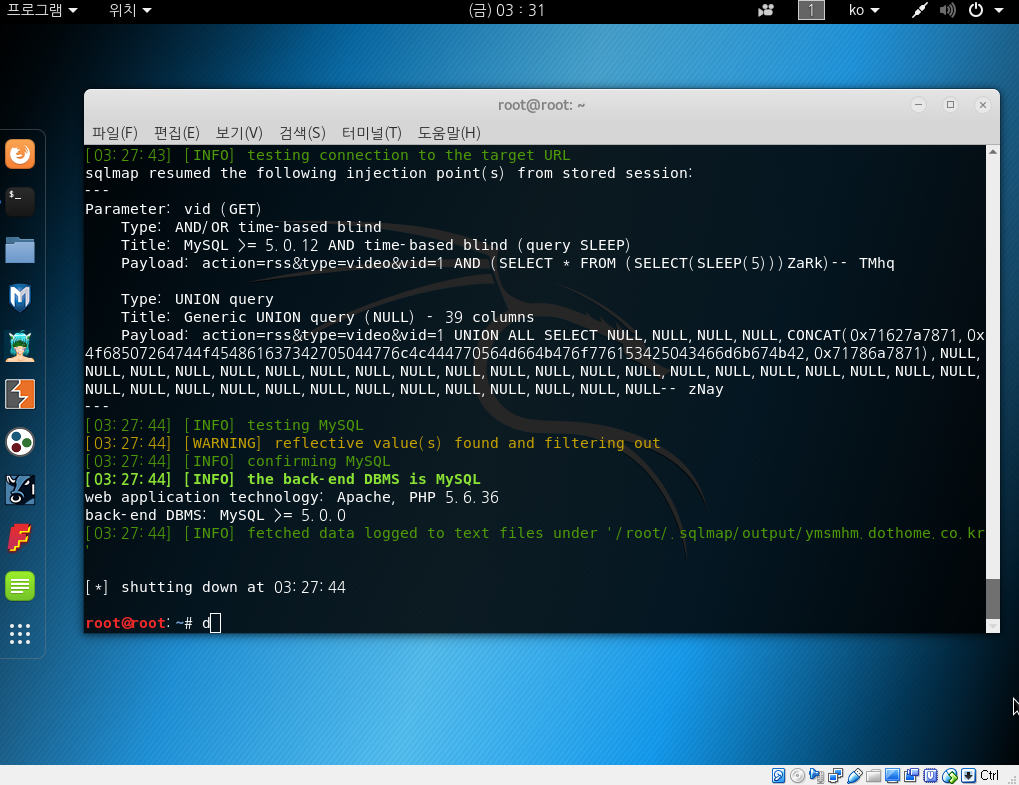


SQL injection의 대상이 되는 워드프레스 플러그인(**contus-video-gallery 2.7**)

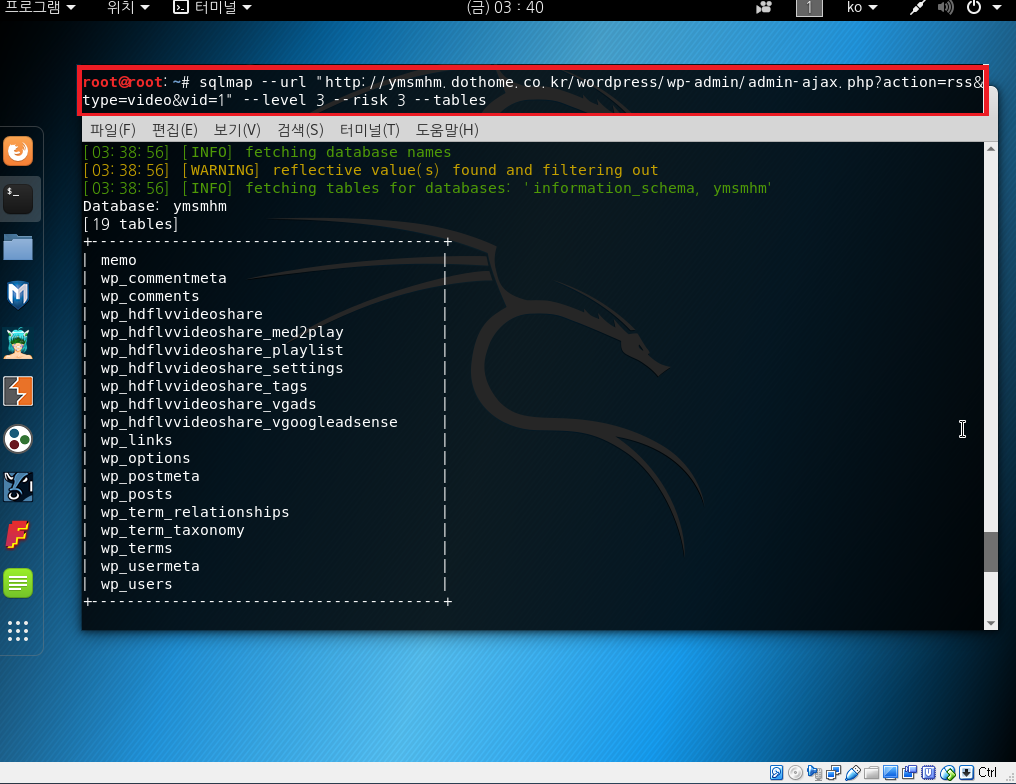
해당 플러그인은 SQL Injection 취약점이 있는 플러그인으로 현재 취약점 패치가 된 새로운 버전이 릴리즈 되었다. 취약점 분석을 위해 취약점이 있는 버전으로 진행한다.



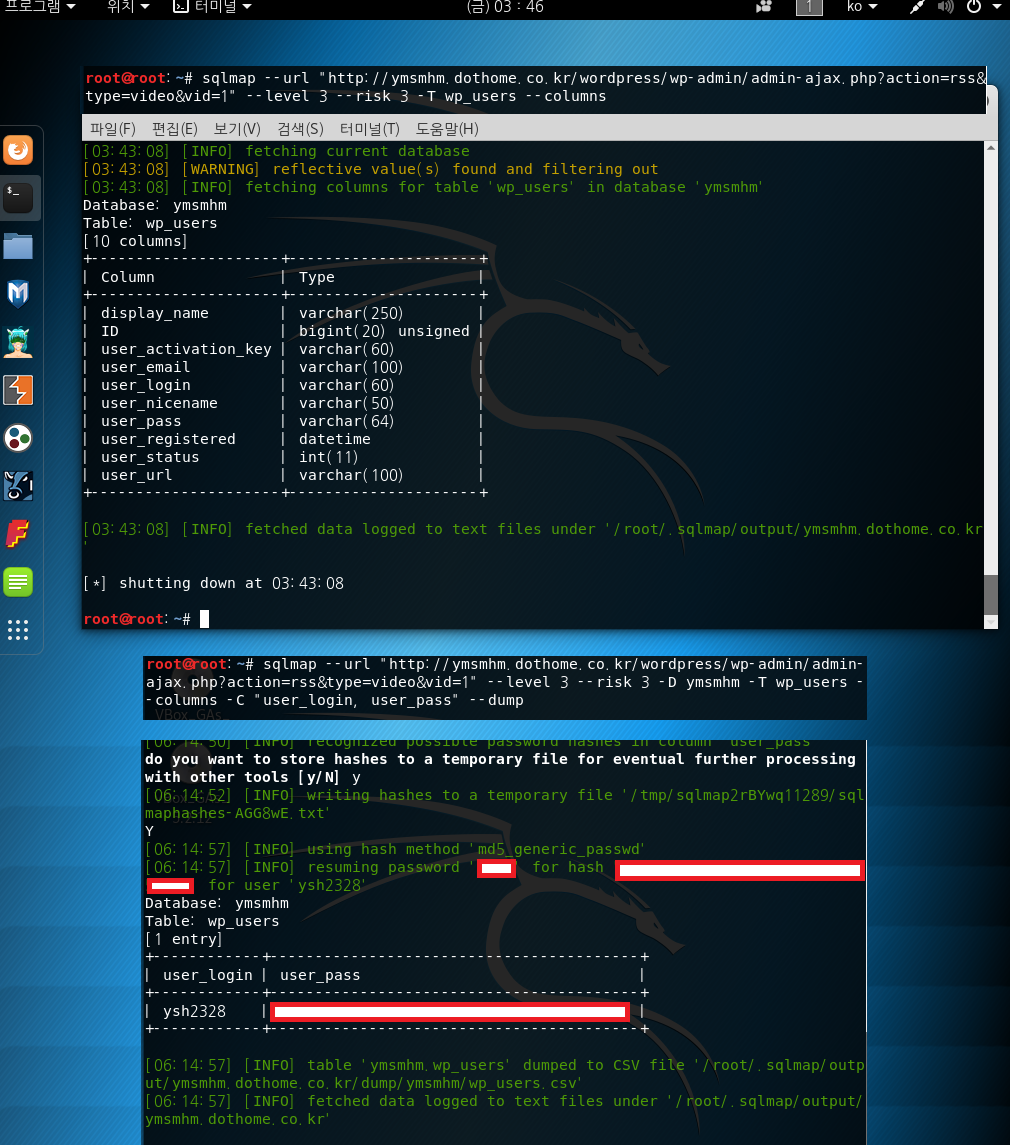
SQL Injection을 위해 sqlmap이라는 툴을 사용한다. 칼리 리눅스에는 내장되어 있으며 Window환경에서도 사용 가능하다. 해당 코드를 입력하면 워드프레스와 취약한 플러그인이 설치된 도메인에 대해 취약점 분석을 시작한다. 30분에서 1시간 가량 시간이 소모된다.



결과. SQL Map은 해당 도메인에 대해 어떤 공격이 가능한지 보여준다. vid변수에 취약점을 발견했다.



SQL내에 어떤 테이블이 있는지 확인하는 명령어와 그 결과



wp\_user 칼럼을 반환하는 명령어와 user\_login(ID), user\_pass(password) 칼럼에 대한 sql인젝션 공격.

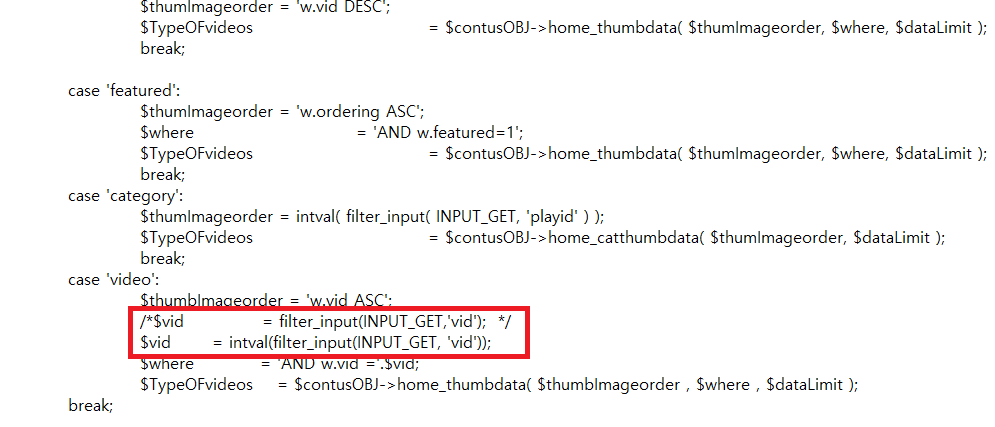
ysh2328이라는 관리자 아이디와 비밀번호가 크랙된 모습이다. 해당 시스템은 비밀번호를 md5로 암호화하여 정보를 저장했지만 sql map에서는 이를 복호화하여 보여준다. 비밀번호가 충분히 길고 복잡하다면 시간이 매우 오래 걸릴 것으로 사료된다.

취약점 보완방법

1. 해당 플러그인은 변수 vid부분에 취약점이 발견되었다. vid를 통해 전달되는 특수문자

/\*, –, ‘, “, ?, #, (, ), ;, @, =, \*, +, union, select, drop, update, from, where, join, substr, user\_tables, user\_table\_columns, information\_schema, sysobject, table\_schema, declare, dual,… 등

들을 엄격하게 필터링하면 sql injection 공격을 방지할 수 있다. 사용자의 입력이 DB Query에 동적으로 영향을 주는 경우 입력된 값이 개발자가 의도한 값 인지 검증한다.



intval함수는 매개변수를 정수형으로 변환하기 때문에 위에서 언급한 특수문자를 넣을 수 없다. 따라서 해당 변수를 intval로 감싼다면 sql injection을 효과적으로 방지할 수 있다.

1. 해당 취약점은 이미 contus사에 보고가 되었고 취약점을 보완한 새로운 버전을 내놓았다.(새로운 버전에도 취약점이 발견되었고 또 다시 새로운 버전을 릴리즈했다) 보안의식을 충분히 가지고 있다면 본인이 운영하는 사이트에 대해 관심을 가지고 항상 최신버전으로 업데이트를 하는 것이 중요하다. 괜히 새로운 버전을 내놓는게 아니다. 사용자들은 업데이트를 주기적으로 해 주는 것 만으로 침해 대응이 충분히 가능하다.

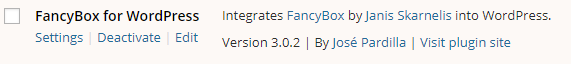
## XSS – Cross Site Scripting.

KISA와 OWASP에서는 XSS를 다음과 같이 정의하고 있다.

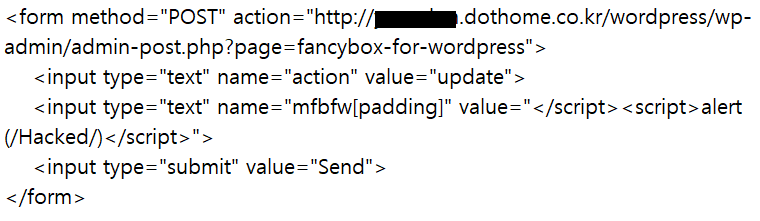
XSS는 애플리케이션에서 브라우저로 전송하는 페이지에서 사용자가 입력하는 데이터를 검증하지 않거나, 출력 시 위험 데이터를 무효화 시키지 않을 때 발생한다. 즉 공격자가 의도적으로 브라우저에서 실행될 수 있는 악성 스크립트를 웹서버에 입력시킬 때, 또는 출력시킬 때 위험한 문자를 중성화 하지 않고 처리하는 애플리케이션에서 발생한다. 일반적으로 자바스크립트에서 발생하지만 VB Script, ActiveX등 클라이언트에서 실행되어 동적 데이터를 생성하는 모든 언어에서 발생이 가능하다.

XSS는 전 세계 시스템 공격 방법 통계에서 3위인 SQL인젝션 다음으로 4위를 차지할 정도로 여전히 많이 사용되는 기법이다. 실제로 XSS취약점은 비교적 쉽게 공격할 수 있으며 웹 앱 개발 시 제거되지 않아 매우 광범위하게 분포 되고 있다.

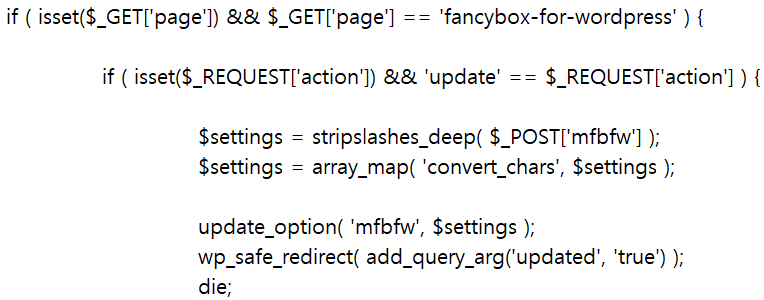
시연



XSS 공격 대상이 되는 플러그인 FancyBox 3.0.2, 이미지나 여러 멀티미디어를 예쁘고 쉽게 표현해준다.

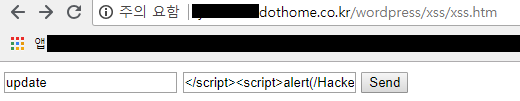


HTML폼을 작성하여 wordpress에 업로드하였다. 위 html코드는 fancybox폴더의fancybox.php 에게 action값과 mfbfw[padding] 값을 넘긴다. mfbfw[padding]은 배열로, alert스크립트를 넘겨준다.

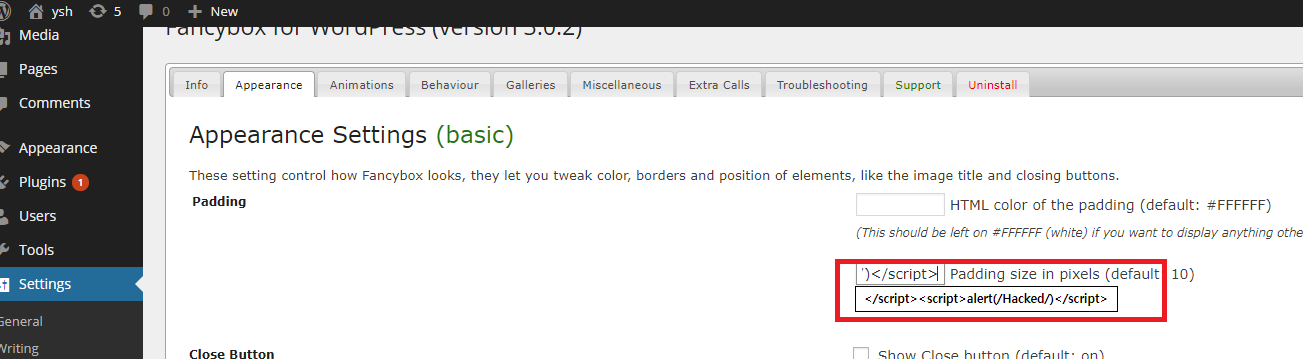


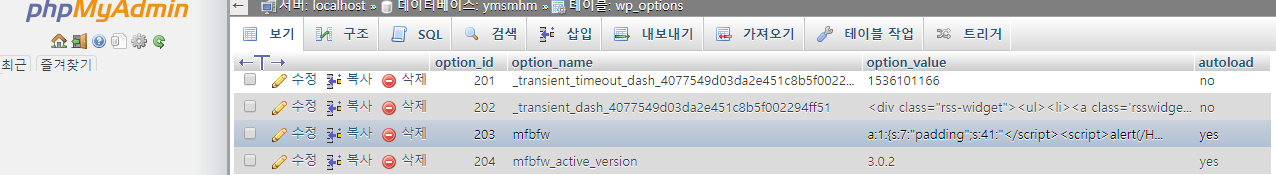
Fancybox.php 코드 일부분이다. page변수로 fancybox-for-wordpress가 있으면 action이 update인지 확인한다.

맞으면 mfbfw[padding] 값을 배열로 받아 DB에 저장한다.

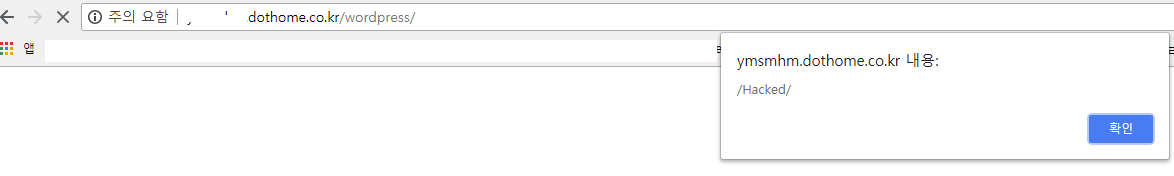


xss.htm 폼에 접속한 모습이다. Send를 누르면 값을 전달 완료한 것이다.

]



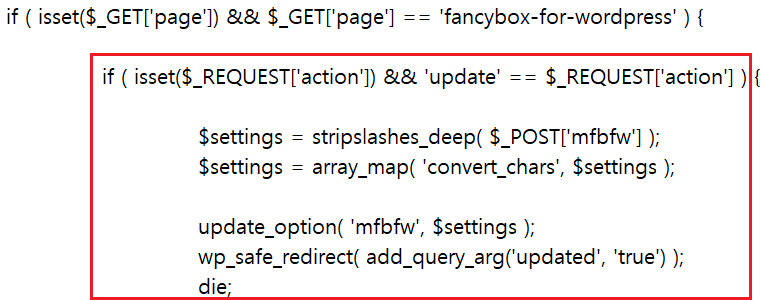
성공적으로 저장된 모습.



워드프레스에 다시 접속한 결과이다. alert스크립트가 실행되어 /hacked/ 라는 메시지를 출력하고 있다.

취약점 보완방법

1. 코드수정



빨간줄 부분을 통째로 없애버린다. 매개변수로 값을 받는 것을 아예 없애버려서 취약점을 보완하였다.

+당연히 업데이트를 통해 취약점을 보완하는 방법도 있다.

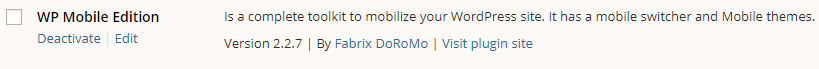
## LFI(Local File INclusion).

OWASP에서는 LFI를 다음과 같이 정의하고 있다.

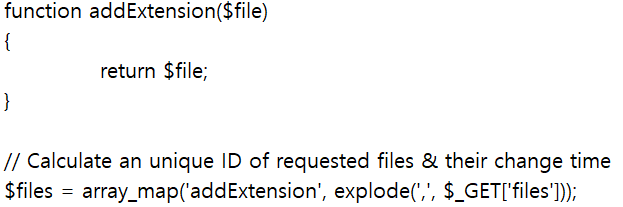
“LFI는 응용 프로그램에 구현 된 취약한 포함 절차를 악용하여 서버에 이미 로컬로 존재하는 파일을 포함시키는 기술이다. 이 취약점은 페이지가 입력되어야 할 파일에 대한 경로를 입력으로 받았을 때 이 입력이 적절하게 삭제되지 않아 디렉토리 탐색 문자 (예 : 점선, 슬래시)를 삽입 할 수 있는 경우에 발생한다.

LFI를 응용하여 RFI(Remote File Inclusion)공격, 즉 원격파일을 업로드하거나 다운받게 할 수 있다.

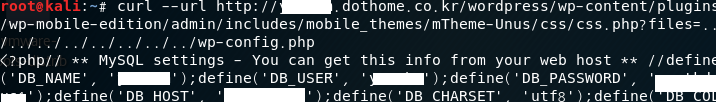
시연



공격대상이 되는 플러그인이다.

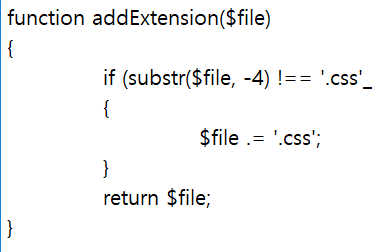
다음은 wordpress/wp-content/plugins/wp-mobile-edition/admin/includes/mobile\_themes/mTheme-Unus/css/css.php 파일 일부이다.

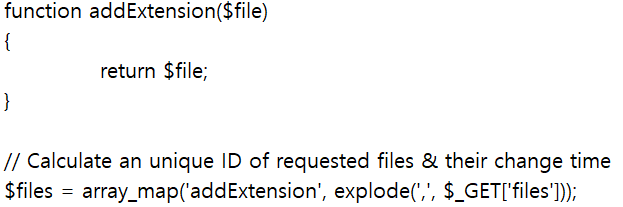
return $file 부분은 file 매개변수를 통해 주소창에서 파일을 받는다. 받은 파일은 addExtension함수를 거쳐 그대로 반환된다.



curl을 이용하여 css.php파일에 접근하고 매개변수에 워드프레스 파일을 넘겨주었다. 워드프레스는 오픈소스 기반이기 때문에 wp-config.php라는 파일의 경로를 누구나 알 수 있어 쉽게 열람할 수 있다. 오픈소스 기반은 LFI 공격에 더 취약하다는 것이다.

마찬가지로 디렉터리 경로를 조금만 수정한다면 etc/passwd파일을 열람할 수 있을 것이다. 하지만 내가 산 것은 닷홈전체가 아니라 도메인 하나이고 잡혀가고 싶지 않기 때문에 못 할거같다.

취약점 보완방법



문제가 되었던 코드 부분과 수정된 모습. 필터링을 통해 파일 가장 뒤 네글자가 .css가 아니라면 .css를 추가해준다. css파일이 아니면 열람할 수 없게 만든 것이다.

만약 파일명만 받고 뒤에 .css를 추가만 해주는 코드였다면 %00이나 %zz같은 시스템이 종료문자로 인식하는 언어를 추가해주어 또 다시 우회할 수 있을 것이다.

두 번째 방법으로는 역시 업데이트가 있다. Web mobile edition 최신버전에서는 css.php 파일을 아예 없애버렸다.

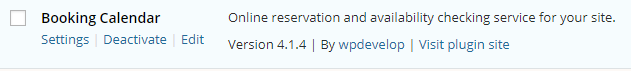
## CSRF(Cross Site Request Forgery)

XSS공격과 함께 거론되는, 빼놓을 수 없는 공격 방식이다. OWASP에서 정보를 찾아보자.

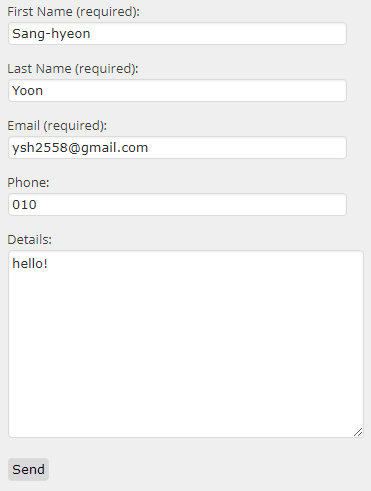
“교차 사이트 요청 위조 (CSRF)는 최종 사용자가 현재 인증 된 웹 응용 프로그램에서 원치 않는 작업을 강제로 수행하는 공격입니다. CSRF 공격은 공격자가 위조 된 요청에 대한 응답을 볼 방법이 없기 때문에 데이터 도난이 아니라 상태 변경 요청을 대상으로 합니다. 소셜 엔지니어링 (이메일 또는 채팅을 통한 링크 보내기와 같은)의 도움으로 공격자는 웹 응용 프로그램의 사용자를 속여 공격자가 선택한 작업을 실행하게 할 수 있습니다. 피해자가 일반 사용자 인 경우 CSRF 공격이 성공하면 자금 이동, 전자 메일 주소 변경 등과 같은 상태 변경 요청을 강제할 수 있습니다. 피해자가 관리 계정 인 경우 CSRF는 전체 웹 응용 프로그램을 손상시킬 수 있습니다.

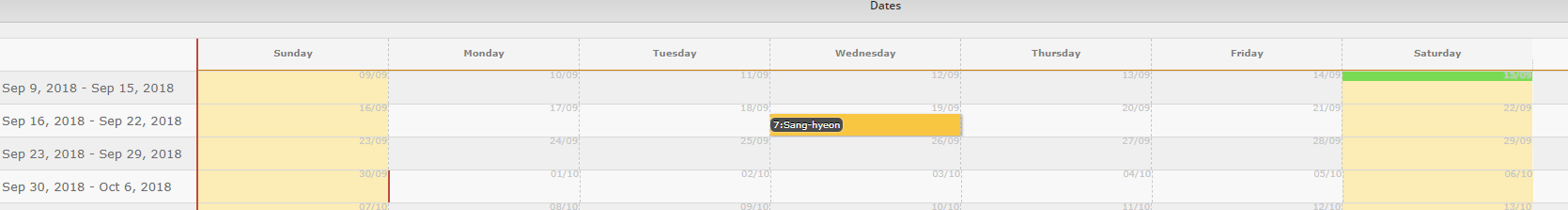
즉 사용자가 본인이 이용하는 사이트와 데이터를 주고 받을 때, 사이트 취약점을 이용하여 사용자의 데이터를 임의로 조작하는 방식이다.

시연

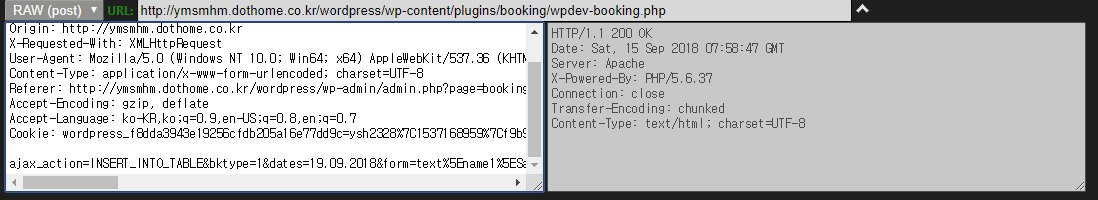


Booking calender라는 플러그인은 사용자에게는 쉬운 예약을, 관리자에게는 편리한 예약 관리를 제공하는 플러그인이다.

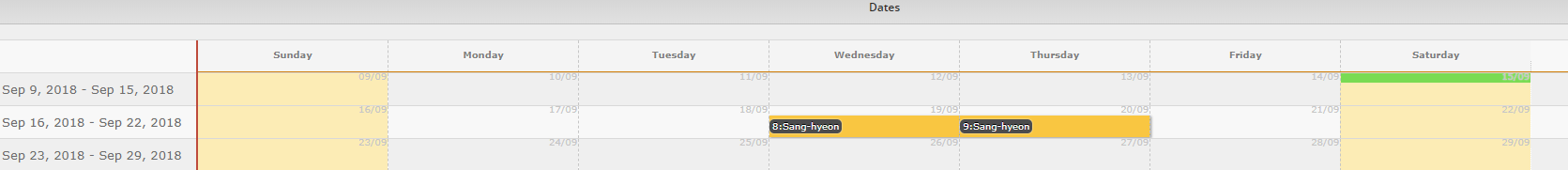




일반 사용자가 이렇게 폼을 채워 전송하면, 예약 요청이 완료된다.



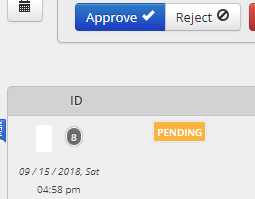
전송을 할 때 크롬 확장 프로그램으로 해당 헤더를 캡쳐링한 모습이다. insert부분이 보이는데 조작하여 재전송 하게 되면,

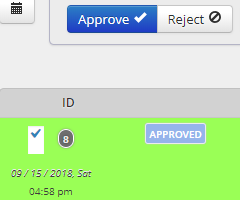


예약이 하나 늘어난 것을 볼 수 있다. 데이터를 추가한 것이다.

이번에는 데이터 수정과 삭제를 해보겠다.

예약 요청을 하게 되면 아래와같이 승인 요청을 기다리게 되는데



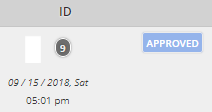


승인 버튼을 누르면 위와 같이 변한다. 이때 헤더를 캡쳐해보면 다음과 같은 코드를 전송하는데,



마찬가지로 id만 조작하면 쉽게 승인을 낼 수 있다.

우리가 임의로 추가한 예약은 id가 9번이므로 id를 9로 바꿔주고 요청을 재 전송하면,



id9번 예약이 마음대로 승인이 된 것을 볼 수 있다.

예를 들어 쇼핑몰에서 입금이 확인되고 승인을 내리는데, 절차를 무시하고 admin이 아닌 사용자가 승인을 내려버리면 불법으로 물건을 받을 수 있게 되는 것이다.

삭제도 같은 원리로 진행한다.



id만 바꿔 재 전송하면 해당 예약이 사라진 것을 볼 수 있다.

취약점 보완방법

1. 해당 플러그인은 적절한 권한 검사를 하지 못하였다. 아래와 같은 간단한 코드를 이용하여 권한 검사를 할 수 있다.

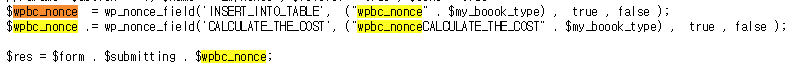
if( current\_user\_can( 'administrator' ) ){

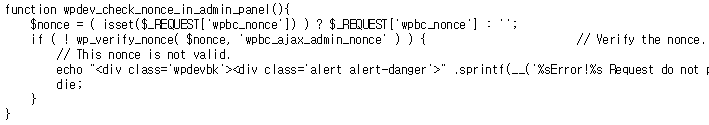
}

워드프레스에서 지원하는 함수이다. wpdev-booking-ajax.php에서는 별 다른 권한체크 없이 데이터의 삭제, 수정, 삭제를 할 수 있게 되어있는데 수정, 삭제의 경우 관리자(administrator)만, 추가의 경우에는 로그인 한 사용자에게만 허용한다면 외부에서 데이터를 수정하는 일은 없을 것이다.

1. 마찬가지로, 간편한 업데이트를 통해 해결한다.







업데이트 버전에서는 ajax를 할 때 권한 검사를 하고 wpdev\_check\_nonce\_in\_admin\_panel()을 추가, 수정, 삭제 등 많은 곳에 수시로 사용하고 있다.