



워드프레스 FTP 인증 오류 해결 방법

DZBLOG 2015-01-30 | #wordpress

워드프레스를 사용하는 이유는 무수히 많은 테마와 플러그인을 사용하기 위함일 것이다. 테마와 플러그인을 설치하는데 오류가 나서 사용할 수 없다면, 더 이상 워드프레스를 사용할 이유는 없다. 대장이 경험한 몇 가지 오류들을 살펴보고 이것을 해결하는 방법을 적어보도록 한다.

FTP 인증 생략

만약 사용하고 있는 워드프레스가 가입형이 아니라면, 당신은 호스팅 서버에 접근 할 수 있는 FTP 혹은 SFTP 정보를 알고 있을 것이다. 호스트네임(IP or domain), 접속 계정명과 비밀번호 이렇게 세 가지 정보가 필요하다. 그런데 정확하게 입력했음에도 접근 할 수 없다는 메시지가 뜨는 경우가 있다. 이 경우 서버의 설정으로 해결 할 수 있으나 이래저래 봐야할게 많으므로 그냥 FTP 인증을 생략해버리자.

`wp-config.php` 파일을 열어 아래 코드를 추가한다.

```
define('FS_METHOD', 'direct');
```

이러면 더 이상 FTP 인증을 거치지 않아도 된다. 만약 워드프레스 폴더에 적절할 권한이 부여되어 있다면, 이로써 플러그인과 테마를 설치하고 업데이트하는데는 문제가 없다. 매번 FTP 정보를 입력하지 않아도 되어 편리하다. 하지만 만약 파일을 읽거나 쓸 수 있는 권한이 없다면 설치도 업데이트도 되지 않는다. 그렇다면 권한을 줘야 한다.

필요한 폴더에 권한 부여

업데이트나 설치는 모두 파일을 쓰는 것이다. 폴더를 읽을 수 있는 권한도 당연히 필요하겠지. 권한이 필요한 폴더는 다음과 같다.

1. wordpress 폴더
2. wp-content/
3. wp-content/plugins/
4. wp-content/upgrade/ (워드프레스 코어 업그레이드 필요)

권한을 변경하는 명령어는 다음과 같다.

혹시 잘 모르는 사람들을 위해 간략히 위 명령어를 설명하자면 `755`는 순서대로 **OWNER, GROUP, OTHER** 순으로 권한을 뜻한다. 읽기권한은 **4**, 쓰기권한은 **2**, 실행권한은 **1**이라는 값을 가진다. 위에서 적은 **7**은 **4,2,1**를 더한 값으로 읽기/쓰기/실행권한을 모두 부여한다는 것을 뜻한다.

그렇다면 여기서 문제, **5**는 어떤 권한을 가지게 될까?

그렇다. **4+1** 즉 읽기와 실행권한을 부여하게 된다. 보안을 생각한다면 쓰기권한은 함부로 부여하지 않는게 좋다. 쓸 수 있다는 것은 지울 수도 있다는 것을 의미하며, 해커가 나쁜 마음을 먹고 심각한 에러를 발생시킬 수 있는 파일을 작성할 수도 있기 때문이다.

소유자만 쓰기 권한이 있다면 소유자의 계정에 접근하지 않는 이상 쓸 수 없다. 하지만 **OTHER** 에도 쓰기 권한을 부여한다면, 다른 서비스의 취약점으로 시스템에 있는 어떤 계정든 접근하게 되었을 때 이 폴더의 데이터는 삭제될 수도 수정될 수도 있다는 말이 된다. 그만큼 권한은 보안에 있어 상당히 중요하다.

잠깐 설명히 길어졌는데 본론으로 돌아와서, 위 명령어를 입력하여 권한을 변경했음에도 접근이 되지 않을 때가 있다. 왜 그럴까?

소유자 확인

우리가 `chmod` 명령어를 통해 변경한 것은 다시 말해서 소유자만 쓸 수 있게끔 하삼! 이라는건데, 그 전에 이 폴더의 소유자가 누구인지 확인할 필요가 있다. 만약 당신이 **root** 계정으로 워드프레스를 설치했다면 폴더에 소유자가 **root** 일 가능성이 크다. 즉 폴더에 무언가를 쓰기 위해서라면 **root** 여야 한다는 것이다. 하지만 워드프레스는 **root**가 아니다. 따라서 역시 접근 오류를 뱉어내며 설치나 업데이트를 하지 못하는 것이다.

그런데 여기서 잠깐 워드프레스가 **root**가 아니라는 것은 워드프레스도 하나의 계정이라는 의미일까?



그렇다. 정확히는 워드프레스를 구동하는 아파치 혹은 엔진엑스의 서비스 계정이다. 워드프레스에서 서버에 파일을 건들기 위해서는 워드프레스가 사용하는 서비스 계정에게 건들고자 하는 폴더/파일에 대한 접근권한이 반드시 필요하다. 보통은 `www-data` 일 것이다. 따라서 소유자를 `www-data`로 변경해줘야만 워드프레스는 비로소 자신이 필요한 폴더에 무언가를 쓰거나 수정할 수 있는 것이다.

보통은 `www-data`라고 했다. 그럼 이 서비스 계정은 어떻게 확인 할 수 있을까?

웹서버 서비스 계정 확인

먼저 자신의 리눅스 버전을 확인해보자.

```
$ cat /etc/issue
```

대장의 리눅스 버전인 우분투 14.04.1 을 기준으로 서비스 계정명을 확인해보도록 하겠다. 이전에 쓰던 우분투나 CentOS(센토스)랑은 그 방법이 조금 달라졌다.  

`httpd.conf` 파일이 없어졌더라? 여튼 계정명 확인을 위해 달려보자.

```
$ vim /etc/apache2/apache2.conf
```

쪽 내리면서 110 번째 줄 정도를 보면 아래와 같은 부분이 있다.

```
# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
```

예전에는 여기서 바로 확인했던 것 같은데... 여튼 주석에 적힌대로 저 파일을 열어보자.

```
$ cat /etc/apache2/envvars
```

```
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
```

드디어 우리가 원하던 정보를 찾았다. 역시 평범하게 `www-data` 라는 계정과 그룹으로 서비스하고 있었다. 리눅스 배포판마다 그 서비스 계정이 상이할 수 있으니 이와 같은 방법으로 확인하도록 하자. 나아가 자신이 아파치의 서비스 계정을 바꾸고 싶을 때 어느 부분을 바꿔야 하는지도 알게되었다.

자 이로써 워드프레스가 접근하고자 하는 폴더의 소유자를 누구로 해야할지 확신이 설 것이다. 이제 소유자를 바꿔보자.

소유자 변경

소유자를 변경하는 명령어는 다음과 같다.

```
sudo chown [계정명]:[그룹명] [파일명]
```

예를들어 `wp-content/` 의 소유자를 변경한다면 아래와 같이 입력한다.

```
sudo chown www-data:www-data wp-content
```



그러면 그 폴더의 소유자만 썩 바뀐다. 그런데... 이거 상당히 귀찮다. 그냥 워드프레스 폴더와 그 내부 모든 파일의 소유자를 `www-data`로 바꿀 수는 없을까? 있다. 바로 `-R` 옵션을 사용하는 것이다.

```
sudo chown -R www-data:www-data wordpress
```

그러면 워드프레스 이하 모든 폴더와 파일의 소유자가 바뀐다. 이제 755 접근권한을 가진 폴더나 파일에 워드프레스가 접근 할 수 있다. 정확히는 아파치를 사용하여 동작하는 모든 서비스가 접근 할 수 있는 것이지만 거기까지는 생각하지 말자. 자 이제 워드프레스에 접속해서 테마와 플러그인을 설치든 업데이트든 마음껏 해보자. 제대로 따라오기만 했다면 인증없이 그리고 접근 오류 따위도 없이 완료될 것이다.

작은 문제점

완벽한 것 같지만 이렇게하면 작은 문제점이 발생한다. 바로 내 계정으로 FTP에 접속하여 파일을 업로드하거나 다운로드 할 수 없다는 것이다... SSH로 접속하면 `sudo` 명령어를 통해 루트 권한을 잠깐 부여받아 뭐든 수행할 수 있지만 FTP는 다르다. 그럼 이 문제는 어떻게 해결해야 할까?

두 가지 방법이 있다.

`[yourId]:[www-data] 775`

하나의 폴더의 소유자는 자신의 계정으로 하고, 워드프레스가 파일쓰기가 필요한 폴더의 접근권한을 775로 바꾸는 것이다. 즉 `www-data` 그룹의 계정은 쓰기가 가능하도록 하는 것이다.

`www-data` 그룹에 사용자 계정 추가

이건 내가 사용하고 있는 방법인데 일단 전체적으로는 소유자와 그룹 모두 `www-data`로 해놓고, `www-data` 그룹에 나의 계정을 추가하고 내가 FTP로 접속하여 파일을 업/다운로드 할 폴더만 775 권한을 주는 것이다.

그룹에 사용자를 추가하는 명령어는 다음과 같다. 기존 그룹을 변경하는 것이 아니라 보조 그룹에 넣는 방법이다.

```
$ sudo usermod -G [userId]:[group]
```

하나에 대한 소유권과 접근권한을 사용자가 강력하게 통제할 수 있는 장점 덕분에 대부분의 서버가 유닉스/리눅스 기반이지 않을까? 생각한다.



Copyright © 2017 Dezang

[Home](#) | [About](#) | [Writing](#) | [Projects](#)