"حق"

۱-باگ های پیمایش دایرکتوری،پیوست فایل داخلی:

Local File Inclusion(LFI)

توابع اسیب پذیر:

Include()
Include_once()
Require()
Require_one()
Fopen()
File()
Fil_get_contents()
….
..

خوب توابع بالا توابعی هستند که استفاده نادرست از اونها میتونه موجب بروز باگ های LFI بشن
که حالا سعی میکنیم برای هر کدم یک مثال ارائه کنیم

خوب همونطور که میدونید در PHP ثابت های GET/POST باعث میشن که یه متغییر مستقیما به وسیله Http header ها مقدار دهی بشن که به اصطلاح از این ثابت ها به عنوان متد GET یا POST یاد میشه..به عنوان مثل گفته میشه فلان متغییر با متد GET مقدار دهی میشه.
خوب حالا یه کد اسیب پذیر رو مورد برسی قرار میدیم:

```
if (isset($_GET['page'])) { $p = $_GET['page'];}
..
..
Include($p);
```

خوب در کد بالا یه متغییر به نام page با متد GET مقدار دهی شده تا به عنوان مثال باعث فراخوانی صفحات دیگه بشه به عنوان مثال :

Index.php?page=home.php

باعث فراخوانی صفحه home.php میشه حالا فرض کنید ما به جای home.php یه فایل دیگه رو به صورت زیر فراخوانی کنیم

Index.php?page=../../../../../etc/passwd

در مثال بالا /.. ها برای بالا رفت از دایرکتوری هاست.خوب فراخونی به صورت بالا باعث دیدن محتویات فایل passwd میشه که به راحتی با عث ایجاد باگ شد

تو کد اسیب پذیر بالا میشه به جای include() از توابع مشابه مثل include_once و یا Require(),require_once هم استفاده بشه در مثال بعدی از تابع require() به همراه کمی تغییرات در کد اسیب پذیر استفاده میکنیم

```
if (isset($_GET['page'])) { $p = $_GET['page'];}
..
..
Include($p.php);
```

خوب اگه دقت کنید برنامه نویس در کد بالا فقط اجازه میده فایل هایی با پسوند php. فراخوانی بشن پس

```
Index.php?page=../../../../../etc/passwd
```

مثل قبل اینجا برای ما کار ساز نیست چون حتما باید پسوند فایل فراخونی شد php. باشه!
را ه حل ما اینجا استفاده از null byte (%00) هست قبل از توضیح بیشتر یه تعریف از نال بایت میکنیم:

خوب نال بایت در حافظه سیستم به معنی انتهای رشته هست و به محض اینکه نال بایت در رشته ظاهر شه اون رشته پایان یافته در نظر گرفته میشه و هرچی بعد از نال بایت ظاهر شه دیگه در نظر گرفته نمیشه
به عنوان مثال رشته زیر رو در نظر بگیرید

```
$st="zhzh";
```

این رشته ۴ حرف هست ولی در حافظه ۵ خونه براش در نظر گرفته میشه به اینصورت:

z	h	z	h	\0

خوب میبینید که یه خونه از حافظه با استفاده از نال بایت انتهای رشته رو مشخص کرده.شما به هر زبانی که برنامه نویسی کنید نحوه ذخیره رشته به صورت بالا خواهد بود .

خوب ما اینجا از نال بایت سو استفاده میکنیم و مفسر php رو خر میکنیم !!!! ☺

```
Index.php?page=../../../../../etc/passwd%00
```

به همین راحتی نال بایت باعث میشه کد ما به صورت زیر ازریابی بشه

```
Include('../../../../../etc/passwd%00.php');
```

و با عث میشه انتهای رشته که php. بود در نظر گرفته نشه و فایل ما به راحتی اینکلود بشه!

مثال بعدی رو با fopen() بررسی میکنیم(اسیب پذیری مربوط به پرتال Power Editor در فایل Edit.php هست)

```
$te=$HTTP_GET_VARS['te'];
$dir=$HTTP_GET_VARS['dir'];
$filename="$dir/$te";
$fd=fopen($filename,"r");
..
..
```

خوب همونطر که میبینید ۲ متغییر $te و$dir توسط متد GET مقدار دهی میشن و متغییر $filename از ترکیب دو متغغیر قبلی ایجاد میشه ودر خط بعدی $filename توسط تابع fopen استفاده میشه که باعث ایجاد باگ میشه..تابع fopen قصد باز کردن فایل $filename رو داره و به نوعی در حال فراخوانی این فایل هست حال ما میتونیم به راحتی فایل دلخواه خودمون رو به صورت زیر فراخوانی کنیم:

editor.php? te=/etc/passwd&dir=../../../../../

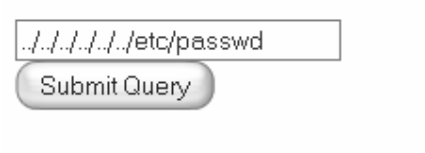
مثال بعدی رو با تابع file_get_contents() بررسی میکنیم..یهکد خیلی ساده فقط برای فهمیدن مشکل مثال میرنم:

```
File_get_contents($_POST['file']);
```

متغییر file با متد POST مقدار دهی میشه و بعد توسط تابع باید محتوای اون گرفته بشه که به نوعی فراخوانی فایل هست که به راحتی میتونیم باگ رو مشاهده کنیم کافیه متغییر file رو ب امقدار دلخواه خودمون مقدار دهی کنیم..خوب چون اینجا از متد POST استفاده شده ما هم باید با استفاده از این متد مقدار دهی رو انجام بدیم ساده ترین راه نوشتم یه فرم html هست که مقادیر رو برای ما ارسال کنه به صورت زیر:

```
<form method="post" action="editor.php"><br>
<input type="text" name="file" size="25"><br>
<input type="submit"><br>
</form>
```

این یه عکس از فرم ایجاد شده:



تابع file() نیز در صورت استفاده نادرست باعث ایجاد باگ خواهد شد. وظیفه این تابع خواندن محتویات یک فایل و قرار دادن ان در یک ارایه است پس به نوعی با عث فراخوانی فایل خواهد شد
در زیر مثال از یک کد اسیب پذیر داریم(کد اسیب پذیر مربوط به Wikepage 2007.2 و در فایل index.php مشاهده شد)

```
$templatefile=$_GET['template'];
if($templatefile="")
$templatefile="index.html";
$template=implode( "", file('theme/'.$pagevars["theme"].'/'.$templatefile));
```

خوب در کد بالا یه متغییر با متد GET مقدار دهی شده و بودن هیچ کنترل امنیتی به وسیله تابع file() به کار گرفته شده که باعث ایجاد باگ خواهد شد

Index.php? template=../../../../../etc/passwd

خوب در تمام مثال های بالا متغییر ها به وسیله متد GET/POST مقدار دهی شدند ولی گاهی اوقات برای مقدار دهی از کوکی ها هم استفاده میشه و برای این کار از تابع $_COOKIE استفاده میشه در بحث اسیب پذیری مربوط به کوکی ها توضیح کاملی در مورد توابع کوکی خواهم داد ☺

خوب یه مثال ساده از کد اسیب پذیری که در اون مقادیر کوکی باعث ایجاد باگ میشه رو مثال میزنم

```
If(isset($_COOKIE['ck']))
$lng=$_COOKE['ck'];
$page="../lang/.$lng.php";
Include_once($page);
```

خوب متغییر $lng توسط مقادیر کوکی مقدار دهی شده و بعد از اون در متغییر $page بکار برده شده و در خط اخر هم فراخوانی شده و به راحتی باعث ایجاد یک باگ شده.
خوب اگر ما کوکی رو ادیت کنیم و با توجه به $lng=$_COOKE['ck']; مقدار ck رو در کوکی به مقدار ../../../../../etc/passwd%00 تغییر بدیم و صفحه اسیب پذیر رو مشاهده کنیم
به راحتی یک باگ LFI خواهیم داشت.برای ویرایش کوکی دو راه دارید یکی ویرایش مستقیم کوکی با استفاده از یک ادیتور و دیگری استفاده از کد جاوا اسکرپیت زیر برای ست کردن مقدار کوکی:

```
javascript:document.cookie = "ck=../../../etc/passwd%00; path=/;";
```

خوب تا اینجا اکثر حالت هایی که باعث این نوع باگ میشد توضیح داده شدند و در کل هر تابعی که به نوعی باعث یک فراخوانی بشه هم میتونه باعث ایجاد این نوع باگ ها بشه!

به عنوان پایان این مبحث دو نوع روش برای پچ کردن اینگونه باگ ها رو یاداوری میکنم:
۱- استفاده از یک ارایه برای کنترل مقادیری که فراخوانی خواهند شد:
در صورتی که مقدار فراخوانی شده در ارایه موجود نبود فراخوانی انجام نخواهد شد.

```
$secur = array("home", "news", "blog", "admin");
$page=$_GET['p'];
if (in_array("$page", $secur)){
include("/home/".$page.".php");}
else
{
die "zaye shodi!!"
}
```

خوب در ابتدا یه ارایه تعریف کردیم و مقادیری که باید فراخوانی بشن رو در اون قرار دادیم در خط بعدی متغییر $page با متد GET مقدار دهی شده و در خط بعدی با استفاده از تابع in_array() چک میکنیم که مقدار داده شده به $page در ارایه موجود باشه که اگر موجود بود که اینکلود میشه و در غیر اینصورت پیغام !! zaye shodi چاپ میشه ☺

روش دوم استفاده از تابع stripos() برای فیلتر کردن مقادیری مثل ../ هست که منطقی تر و راحت تر هم هست

در پایان این نکته باید گفته بشه که:اینطور در نظر گرفته شده که شما یه اشنایی نسبی با php دارید و زیاد رو ی نحوه عملکرد توابع و وظیفه اونها توضیح داده نشده و صرفا به تشریح قسمت های اسیب پذیر پرداخته شده در صورتی که در مورد نحوه عملکرد هر کدام از توابع ذکر شده نیازمند اطلاعات بیشتر بودید از سایت php.net به صورت زیر کمک بگیرید:

http://php.net/func_name

این مقاله فقط برای شما نوشته شده صرفا بخاطر اینکه مطالب گفته شده رو بهتر درک کنید و دلیل دوم اینکه شما یکی از شاگرد های با استعداد من هستید که ازتون انتظار پیشرفت دارم ! فعلا قصد منتشر کردن مقاله رو ندارم..به امید اینکه شاهد پیشرفت روز افزون شما باشم.

www.virangar.net

hadihadi