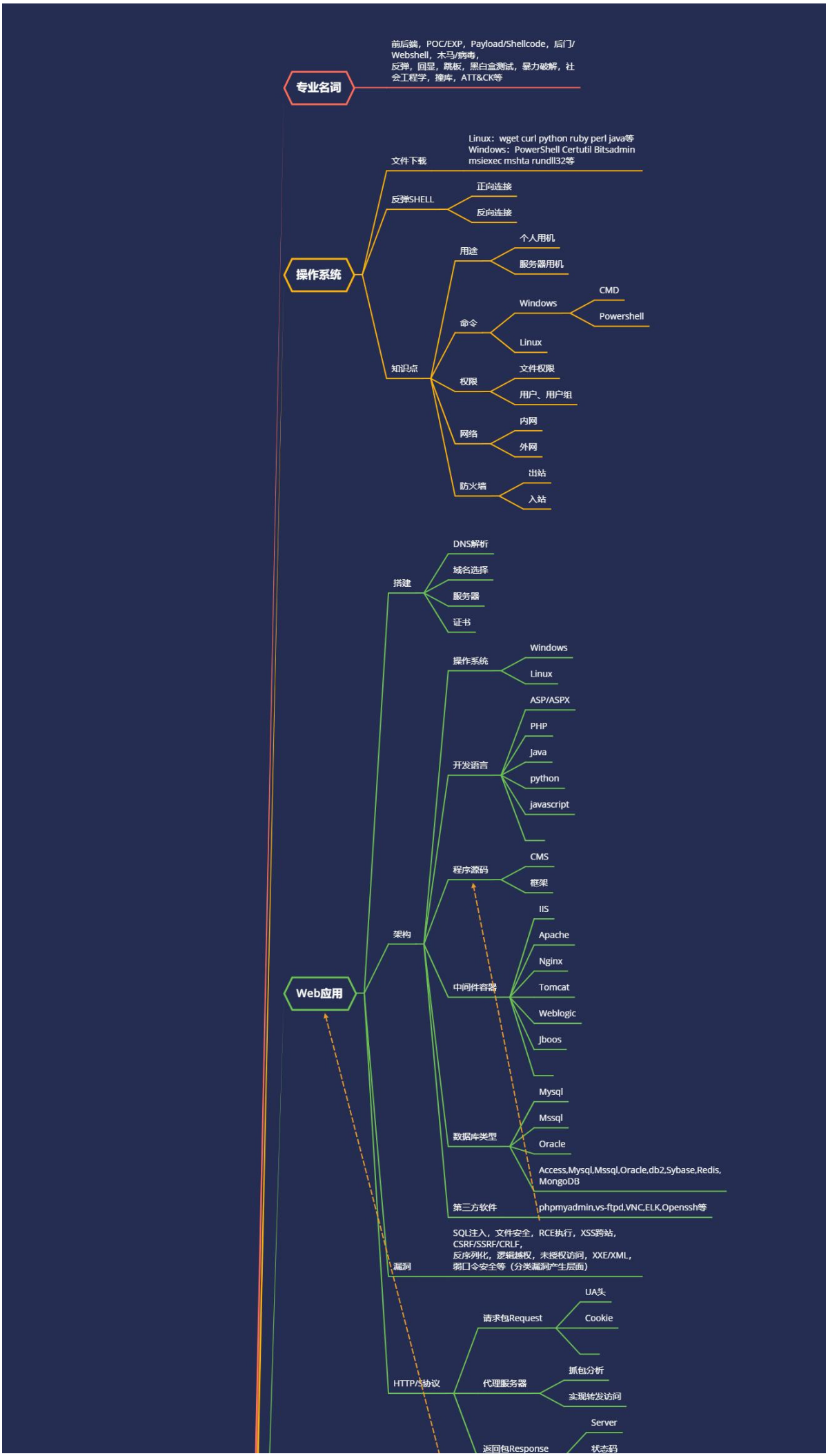
Day13　PHP 开发-个人博客项目&文件操作类&编辑器&上传下载删除读写


专业名词
前后端，POC/EXP，Payload/Shellcode，后门/Webshell，木马/病毒，
反弹，回显，跳板，黑白盒测试，暴力破解，社会工程学，撞库，ATT&CK等
操作系统
文件下载
Linux：wget curl python ruby perl java等
Windows：PowerShell Certutil Bitsadmin mshta rundll32等
反弹SHELL
正向连接
反向连接
知识点
用途
个人用机
服务器用机
命令
Windows
CMD
Powershell
Linux
权限
文件权限
用户、用户组
网络
内网
外网
防火墙
出站
入站
Web应用
搭建
DNS解析
域名选择
服务器
证书
架构
操作系统
Windows
Linux
开发语言
ASP/ASPX
PHP
Java
python
javascript
程序源码
CMS
框架
中间件容器
IIS
Apache
Nginx
Tomcat
Weblogic
Jboos
数据库类型
Mysql
Mssql
Oracle
Access,Mysql,Mssql,Oracle,db2,Sybase,Redis,MongoDB
第三方软件
phpmyadmin,vs-ftpd,VNC,ELK,Openssh等
漏洞
SQL注入，文件安全，RCE执行，XSS跨站，
CSRF/SSRF/CRLF，
反序列化，逻辑越权，未授权访问，XXE/XML，
弱口令安全等（分类漏洞产生层面）
HTTP/S协议
请求包Request
UA头
Cookie
代理服务器
抓包分析
实现转发访问
返回包Response
Server
状态码

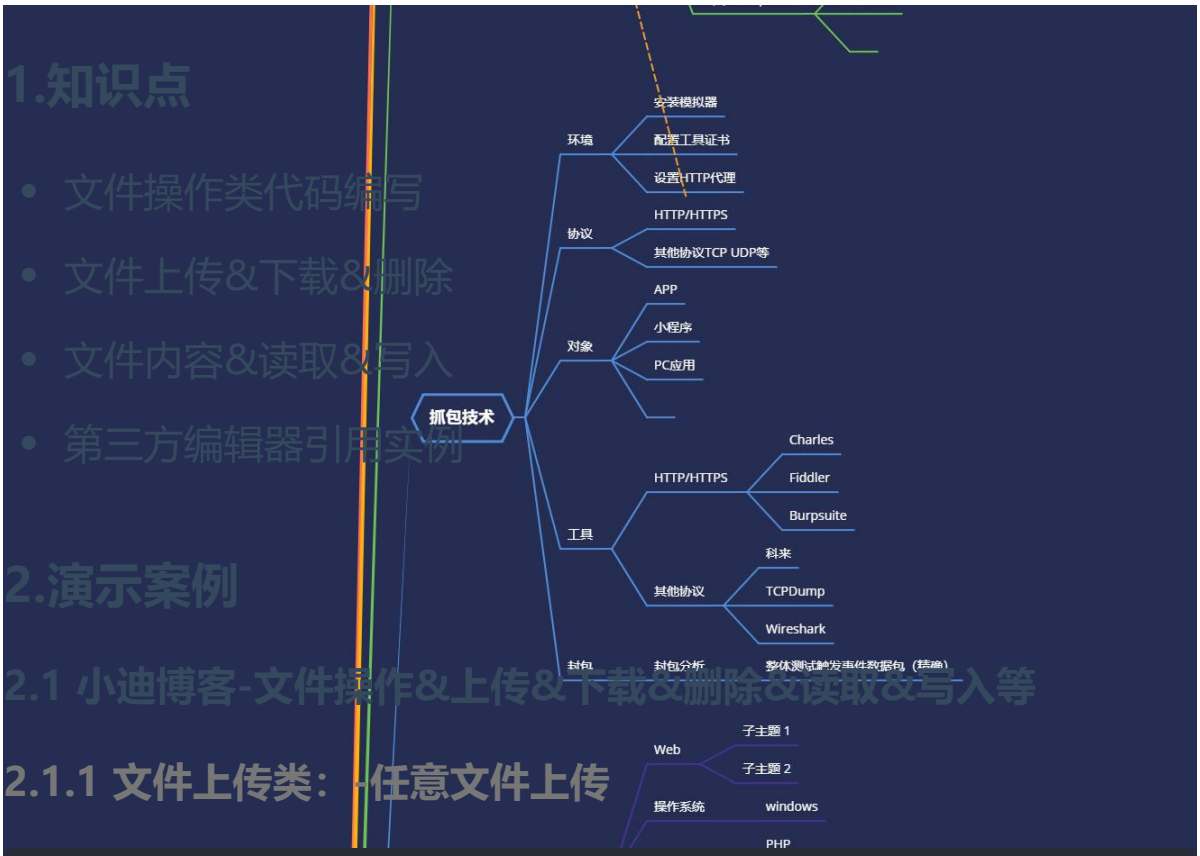
1.知识点

- 文件操作类代码编写
- 文件上传&下载&删除
- 文件内容&读取&写入
- 第三方编辑器引用实例

2.演示案例

2.1 小迪博客-文件操作&上传&下载&删除&读取&写入等

2.1.1 文件上传类：-任意文件上传




```
1  代码自主写
2  编辑器引用
```

代码：


```
1  <html lang="en">
2  <head>
3      <meta charset="UTF-8">
4      <title>文件上传</title>
5  </head>
6  <body>
7
8  <script type="text/javascript" charset="utf-8"
   src="../blog/ueditor/ueditor.config.js">
   </script>
9  <script type="text/javascript" charset="utf-8"
   src="../blog/ueditor/ueditor.all.js"></script>
10 <h1>编辑器上传</h1>
```

```
11  <div>
12      <script id="editor" type="text/palin"
    name="bianji"></script>
13  </div>
14
15  <script type="text/javascript">
16      var ue=UE.getEditor('editor');
17  </script>
18  <h1>文件上传</h1>
19  <form action="upload.php" method="post"
    enctype="multipart/form-data">
20      <p><input type="file" name="upload"></p>
21      <p><input type="submit" value="上传"></p>
22  </form>
23  </body>
24  </html>
```

获取方式3 黑源码

```
1   <?php
2   //获取上传文件名
3   @$file_name=$_FILES['upload']['name'];
4   //获取上传文件类型
5   @$file_type=$_FILES['upload']['type'];
6   //获取上传文件大小
7   @$file_size=$_FILES['upload']['size'];
8   //获取上传文件临时文件名
9   @$file_tmpname=$_FILES['upload']['tmp_name'];
10  //获取上传文件是否错误
11  @$file_error=$_FILES['upload']['error'];
12
13  echo $file_name."<hr>";
14  echo $file_type."<hr>";
15  echo $file_size."<hr>";
```

```php
echo $file_tmpname."<hr>";
echo $file_error."<hr>";

if (@$file_error>0){
    echo '上传出错! ';
}
else{
    move_uploaded_file(@$_FILES["upload"]["tmp_name"], "upload/" . @$_FILES["upload"]["name"]);
    echo "文件存储在: " . "upload/" . @$_FILES["upload"]["name"];
}
?>
```

可能产生文件上传漏洞:

- 原理: 由于程序员在对用户文件上传功能实现代码没有严格限制用户上传文件后缀,以及文件类型或者处理缺陷, 而导致用户可以越过本身权限向服务器上传木马去控制服务器.
- 危害 :操作木马文件提权 获取网站权限.

2.1.2 文件下载类: -任意文件下载

```
1  直连 URL 访问
2  传参头部修改
```

代码:

```html
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>文件下载</title>
```

```
</head>
<body>

<script type="text/javascript" charset="utf-8" src="../blog/ueditor/ueditor.config.js">
</script>
<script type="text/javascript" charset="utf-8" src="../blog/ueditor/ueditor.all.js"></script>
<h1>文件列表</h1>
<?php getfilename(); ?>

<h1>直连下载</h1>
<form action="" method="post">
    <input type="text" name="filename">
    <input type="submit" value="下载">
</form>
<?php
@$name=$_POST['filename'];filenameurl($name);?>
<h1>传参下载</h1>
<form action="" method="post">
    <input type="text" name="downname">
    <input type="submit" value="下载">
</form>
<?php
@$name=$_POST['downname'];filenameget($name);?>

</body>
</html>
```

```
<?php
//自定义文件文件夹读取函数
function getfilename(){
```

```php
    $dir=getcwd();
    $file=scandir($dir.'/soft');
    foreach ($file as $value){
        if($value != '.' && $value != '..') {
            $arr[] = $value;
            echo $value.'<br>';
        }
    }
}
//自定义文件直连下载
function filenameurl($name){

 $url='http://'.$_SERVER['HTTP_HOST'].'/blog/soft/'.$name;
    #header("location:$url");
}
//自动以文件传参下载
function filenameget($name){
    $filename = $name;
    $download_path = "soft/";
    if(eregi("\.\.", $filename)) die("抱歉，你不能下载该文件！");
    $file = str_replace("..", "", $filename);
    if(eregi("\.ht.+", $filename)) die("抱歉，你不能下载该文件！");

// •创建文件下载路径
    $file = "$download_path$file";

// •判断文件是否存在
    if(!file_exists($file)) die("抱歉，文件不存在！");
```

```php

// 文件类型，作为头部发送给浏览器
    $type = filetype($file);

// 获取时间和日期
    $today = date("F j, Y, g:i a");
    $time = time();

// 发送文件头部
    /*
    header("Content-type: $type");
    header("Content-Disposition: attachment;filename=$filename");
    header("Content-Transfer-Encoding: binary");
    header('Pragma: no-cache');
    header('Expires: 0');
    */
// 发送文件内容
    set_time_limit(0);
    readfile($file);
}
?>
```

可能产生文件下载漏洞：

- 原理：简单来说，就是一些网站由于业务需求，往往需要提供文件查看或下载功能，但若对用户查看或下载的文件不受限制，则恶意用户就能能够查看或下载任意敏感文件，这就是文件查看与下载漏洞。
- 危害：下载服务器的任意文件；下载web业务的代码，服务器和系统的具体配置信息；下载数据库的配置信息；对内网的信息探测；下载数据库文件；下载使用脚本代码。

2.1.3 文件删除类-任意文件删除

```
文件删除
文件夹删除
```

代码:

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>文件删除</title>
</head>
<body>

<script type="text/javascript" charset="utf-8" src="../blog/ueditor/ueditor.config.js">
</script>
<script type="text/javascript" charset="utf-8" src="../blog/ueditor/ueditor.all.js"></script>
<h1>文件删除</h1>
<?php getfilename(); ?>
<form action="" method="post">
    <input type="text" name="filename">
    <input type="submit" value="删除">
</form>
<?php @$name=$_POST['filename'];filedel($name);?>
<h1>文件夹删除</h1>
<?php getfilename()?>
<form action="" method="post">
    <input type="text" name="filedir">
    <input type="submit" value="删除">
```

```
</form>
<?php @$dir=$_POST['filedir'];filedeldir($dir);?>
</body>
</html>
```

```php
<?php
//自定义文件文件夹读取函数
function getfilename(){
    $dir=getcwd();
    $file=scandir($dir);
    foreach ($file as $value){
        if($value != '.' && $value != '..') {
            $arr[] = $value;
            echo $value.'<br>';
        }
    }
}
//自定义文件删除函数
function filedel($name){
    @unlink($name);
}
//自定义文件夹删除函数
function filedeldir($dir){
    @rmdir($dir);
}
?>
```

可能产生文件删除漏洞：

- 原理：同样是被删除文件的变量用户可控，且没有进行严格的校检，所以导致任意文件删除，再配合目录遍历，删除硬盘上的其他文件。

- 危害：这个漏洞的危害还是很大的，别人可以删除你电脑上的私密文件等。可能哪天重启服务器发现服务器崩溃了，都有可能是这个漏洞造成的。

2.1.4 文件内容操作类-任意文件读取&写入

```
文件读取
文件写入
```

代码：

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>文件内容操作</title>
</head>
<body>

<script type="text/javascript" charset="utf-8" src="../blog/ueditor/ueditor.config.js">
</script>
<script type="text/javascript" charset="utf-8" src="../blog/ueditor/ueditor.all.js"></script>
<h1>文件列表</h1>
<?php getfilename(); ?>

<h1>读取操作</h1>
<form action="" method="post">
    <input type="text" name="r">
    <input type="submit" value="读取">
</form>
<?php @$name=$_POST['r'];fileread($name);?>
```

```
<h1>写入操作</h1>
<form action="" method="post">
    文件：<input type="text" name="w">
    内容：<input type="text" name="txt">
    <input type="submit" value="写入">
</form>
<?php
@$name=$_POST['w'];@$txt=$_POST['txt'];filewrite
($name,$txt);?>

</body>
</html>
```

```
<?php
//自定义文件文件夹读取函数
function getfilename(){
    $dir=getcwd();
    $file=scandir($dir);
    foreach ($file as $value){
        if($value != '.' && $value != '..') {
            $arr[] = $value;
            echo $value.'<br>';
        }
    }
}
//自定义文件读取函数
function fileread($name){
    $f=fopen($name,"r");
    $code=fread($f,filesize($name));
    echo $code;
    fclose($f);
}
```

```php
//自定义文件写入函数
function filewrite($name,$txt){
    $f=fopen($name,"a+");
    fwrite($f,$txt);
    fclose($f);
}
?>
```

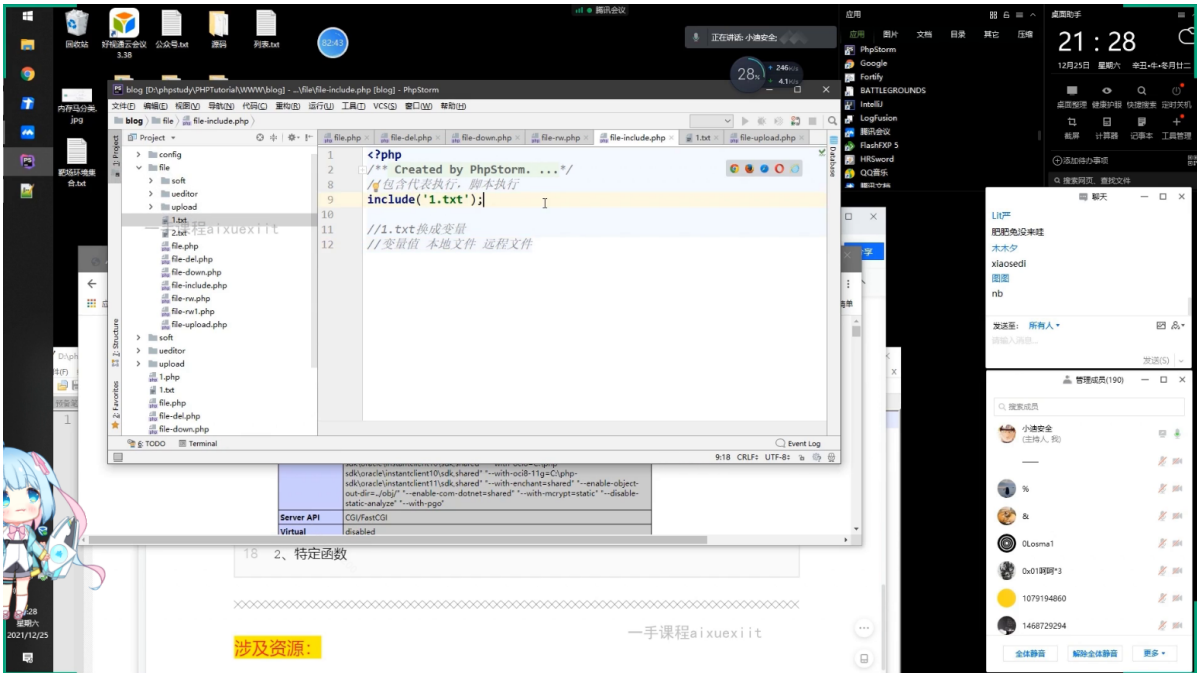
可能产生任意文件读取写入漏洞：

- 原理：任意文件读取写入漏洞*的*原理其实就是由于程序对客户端传入的参数未作合法性的检验造成的。
- 危害：可以*读取*或者下载服务器的配置文件，脚本文件；读取或者下载数据库的配置文件；读取网站源码文件，进行代码审计；对内网的信息进行探测等

2.1.5 文件包含-任意文件包含

```
本地文件包含
远程文件包含
```

代码：



- 原理：文件包含漏洞属于代码注入漏洞，为了减少重复代码的编写，引入了文件包含函数，通过文件包含函数将文件包含进来，直接使用包含文件的代码。
- 危害：读取敏感文件；远程包含shell；图片上传并包含图片shenll；使用伪协议；包含日志文件GetShell；截断包含

3.Web漏洞核心

```
1  可控变量
2  特定函数-函数的多样化
```