PHP代码审计

审计套路

- 通读全文法（麻烦，但是最全面）
- 敏感函数参数回溯法（最高效，最常用）
- 定向功能分析法（根据程序的业务逻辑来审计）
    - 初始安装
    - 信息泄露
    - 文件上传
    - 文件管理
    - 登录认证
    - 数据库备份恢复
    - 找回密码
    - 验证码
    - 越权
    - 注入
    - 第三方组件
    - CSRF,SSRF,XSS.....

审计方法

- 1.获取源码
- 2.本地搭建调试
  可先使用扫描器识别常见传统漏洞，验证扫描器结果，手动正则
- 3.把握大局
  对网站结构，入口文件(查看包含了哪些文件)，配置文件(看数据库编码)，路由，伪全局变量和全局 filter，资源加载顺序，了解数据库处理模式，考察 filter 是否绕过，了解 XSS 过滤机制，考察 filter 是否可绕过，错误信息输出控制，对每个模块的功能进行了解，配合文件数据库监控，从安装到后台功能使用和前台功能使用走一波，仔细观察每步的变化，找不到问题再开始认真审计

常见漏洞

安装问题

- **1.自动删除这个安装文件**
  通过生成一个lock文件来判断程序是否安装过
- **2.根本无验证**
  安装完成后不会自动删除文件，又不会生成lock判断是否安装过
  参考漏洞：PHPSHE B2C 重装 wooyun-2014-062047.html
- **3.安装 file**
  直接用 GET 提交 step 绕过，直接进入下一步
- **4.变量覆盖导致重装**
  可以 GET,POST,COOKIE 任意提交一个变量名 $insLockfile，给其赋空值，覆盖掉 $insLockfile，从而让 file_exists 为 false 就不会退出
  参考漏洞：frcms 重装系统 wooyun-2014-073244.html
- **5.判断 lock 后，无 exit**
  判断是否存在 lock 文件，如果存在 lock 文件，就会 header 到 index.php，但是 header 后并没有 exit，所以 并不会退出，类似的还有 javascript 弹个框
  参考漏洞：开源轻论坛 StartBBS 前台 getshell wooyun-2013-045143.html
- **6.解析漏洞**
  在安装完成后会将 install.php 重命名为 index.php.bak，但是由于 Apache 的解析漏洞：如果无法识别到最后一个后缀的话，就会向上解析，那么就又变成了 php 了，然后结合安装时的变量覆盖又成重装了。
- **7.满足一些条件不会退出的**
  参考漏洞：建站之星 Sitestar 前台 Getshell 一枚 wooyun-2014-054387.html

包含漏洞

- **1.本地文件包含**
  很多都限制了包含的后缀结尾必须为.php，例如include($a.'.php')，需要截断后面的 .php
  - 00 截断

    ```
    gpc off && php < 5.3.4
    ```

  - 长文件名截断
  - 转换字符集造成的截断

> <iconv()截断>

- 伪协议
  - 截取字符判断是不是 .php
  - 用 zip (或者 phar)协议绕过

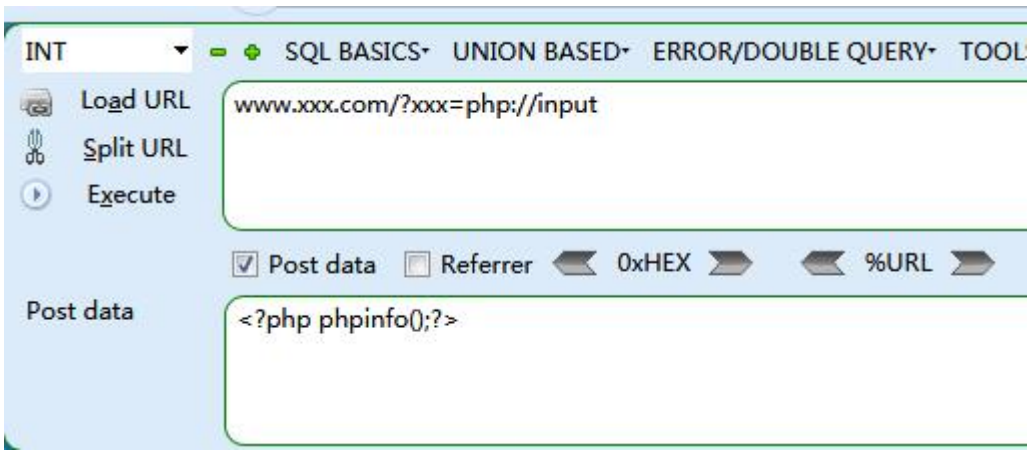
    首先新建一个 1.php，里面 phpinfo，然后压缩成 .zip，然后把 zip 的名字改成 yu.jpg，然后把这个 .jpg 上传上去，然后包含

    Example:

    > http://localhost/php/include.php?include_file=zip://C:\wamp\www.php\1.jpg%231.php`
- 包含日志，环境变量

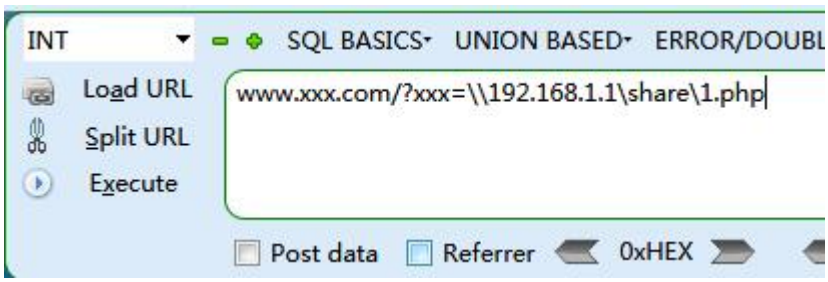
- **2.见 <PHP技巧之截断>**
- **3.远程文件包含**
  - 包含远程文件，或者伪协议 php://input data 等
  - 条件：allow_url_include on ，默认是 off
  - allow_url_include off 条件下 RFI
    - allow_url_include 为 on ,allow_url_fopen 为 off

      伪协议：

    - allow_url_include && allow_url_fopen 为 off

      包含共享文件：

      互联网上 445 端口基本上被过滤

- allow_url_fopen 默认是 on

找回密码

- **1.验证token**

  在找回密码的时候生成一个 token，然后存储到数据库中，然后把找回密码的地址发到邮箱中，url 中就含有 token，由用户点开后就能修改密码
- **2.延伸**

  一些 cms 的密码加密方式很难破掉，有时候拿到了管理的密码破不开，利用方法：一般找回密码是用的邮箱，首先把管理的邮箱注入出来，然后再去找回密码，再把数据库的 token 注入出来，构造一下地址，就能重置密码
- **3.rand函数生成token**
  - `$resetpwd=md5((rand());`

    对 rand() 函数生成出来的数字进行 MD5

    某些平台下(例如 windows) RAND_MAX 只有 32768，如果需要的范围大于 32768，那么指定 min 和 max 参数就可以生成大于 RAND_MAX 的数了，或者考虑用 mt_rand() 来替代它

    参考漏洞：Thinksaas 找回密码处设计错误利用账户可找回密码 wooyun-2014-050304.html
  - `$encryptstring=md5($this->time.$verification.$auth);`

    ```
    $timetemp=date("Y-m-d H:i:s",$this-
    >time);$auth=util::strcode($timetemp,'ENCODE');`
    ```

    算法的 KEY 并没有初始化，如果知道了这个时间，就可以生成加密的字符串

    参考漏洞：Hdwiki 设计缺陷知邮箱可改密码（包括管理员） wooyun-2014-067410.html

上传漏洞

- **1.未验证上传后缀**
- **2.验证上传后缀被bypass**
- **3.上传的文件验证了上传后缀，但是文件名不重命名**

  截断 yu.php%00.jpg
- **4.上传路径可控**
- **5.解析漏洞**

    - Nginx

      yu.jpg/1.php
    - Apache

      yu.php.xxx
- 6.验证方法
    - MIME，客户端的 JS 验证，白名单，黑名单
- 7.绕过
    - 大小写
    - 文件名没 trim

      在文件名后面加空格，windows 下的 x.php%81-%99 decode 后仍为 x.php ，windows 下的特性 .php::$data

文件操作

任意文件删除，任意文件复制，任意文件重命名，任意文件移动，任意文件下载
首先尝试拿到配置文件中的数据库连接账号和密码，然后外链
拿到配置文件，拿到加密解密函数的 key，生成加密字符串，结合具体的代码利用

- 1.文件删除
    - 由于全局的过滤而不能注入时，可以用任意文件删除删掉这个文件
    - 删除安装文件生成的 lock 文件，重装
    - 参考漏洞：phpcms 2008 sp4 爆路径及任意文件删除漏洞 wooyun-2010-0412.html
- 2.文件复制
    - 要复制的文件 要复制的路径

      当两个都完全可控，可以直接把自己上传的图片复制成一个 .php 马
    - 复制的文件可控 要复制的路径不可控

      ```
      copy(ROOT_PATH."$webdb[updir]/$value",ROOT_PATH."$webdb[updir]/{$value}.jpg")
      ```

      可以把 $value 控制为保存了 qibocms 的加密函数的 key 的配置文件，复制后成一个 .jpg 直接打开就可以看到 key
- 3.文件下载

  下载配置文件,拿到 key

  参考漏洞：qibocmsV7 整站系统任意文件下载导致无限制注入多处 wooyun-2014-066459.html

- 4.文件写入
- 5.文件包含

加密函数

拿到加密函数的 key，加密一些特殊字符然后拿到加密的字符串

- 1.加密可逆
  弱算法导致了知道明文，知道密文，可逆，拿到加密函数的 key，从而自己生成一个想要的加密字符串
  参考漏洞：DedeCMS-V5.7-SP1(2014-07-25)sql 注入+新绕过思路 wooyun-2014-071655.html
  参考漏洞：phpcms 最新版绕过全局防御暴力注入 wooyun-2014-066138.html
- 2.加密可控
  要加密的内容是可控的，密文会输出，这个可控的点能引入特殊字符，那么把一些特殊字符带入到这里面，拿到密文，再找到一处 decode 后会进行特殊操作的点，然后进行各种操作。
  参考漏洞：程氏舞曲 CMS 某泄露，导致 sql 注入 wooyun-2014-080370.html
  参考漏洞：PHPCMS最新版 (V9) SQL 注入一枚 wooyun-2013-024984.html
- 3.key泄露
  参考漏洞：一个 PHPWIND 可拿 shell 的高危漏洞 wooyun-2014-072727.html

XSS

- 1.输入输出
- 2.foreach()的key值
- 3.removeXSS函数
- 多个函数处理，插入辣鸡数据绕过第一个函数后，第二个函数过滤了辣鸡数据

CSRF

- 1.后台敏感操作
- 2.修改权限、导出数据等高危功能
- 3.Login Form CSRF

SSRF

- 1.绕过本地 IP 过滤(畸形 IP,本地网段覆盖不完全)
- 2.协议白名单
- 3.跳转到本地 IP
- 4.DNS 解析到本地 IP
- 5.DNS rebinding
- 6.Content-Disposition:attachment;filename="evil_file.exe;.txt" 分号截断可绕过跳板机的 filter

撸库

- 1.失败后没有清空 session 中的验证码
- 2.ip 第一次出现，验证码为默认值
- 3.验证码 md5 显示在 cookie 中
- 4.session 保存到文件

命令注入

- 常见命令注入函数
  - sysytem()
  - exec()
  - passthru()
  - shell_exec()
  - call_user_func()
  - array_map()
  - array_filter()
  - usort()
  - pcntl_exec()
  - popen()
  - proc_open()
  - mail()   ---> PHPmail RCE
  - $_GET['a']($_GET['b'])

登录认证

- 1.session
- 2.算法

XXE

- simplexml_load_string()
  默认情况下会解析外部实体,造成安全威胁,导致任意文件读取、命令执行漏洞。

越权

- 1.通过 ID 操作
- 2.通过 cookie 操作

注入

把用户可控的一些变量，带入到了数据库的各种操作中，并且没有做好过滤，例如：在注册用户的时候检测用户名是否存在，SQL 语句是拼接 SQL

- 1.select注入
  一般使用 union select 联合查询
- 2.update注入
  - update set 的位置
    看这个表的哪个 column 会被展示出来，就把查询出来的内容显示到这里
  - where 后
    通过盲注的方式列出数据
- 3.insert注入
  把要输出的数据插入到这个 column 里面去
- 4.delete注入
  通过盲注的方式列出数据
- 5.数字型注入
  变量并没有用单引号括住，不需要用单引号区分数据与 SQL 命令，这样就会让一般的GPC等机制无用，因为不包括特殊字符强制类型转换 intval
- 6.字符型、搜索型
  - 有单引号括住，需要闭合单引号
  - 全局没有做 addslashes，在查询的时候再对一些用户可控的变量进行 addslashes，遗漏了某些变量没 addslashes
  - 全局做 addslashes,在全局文件中对 GET POST COOKIE 做 addslashes
    首先 get magic quotes gpc 判断 GPC 是否开启，如果没开启就调用 addslashes 来转义，如果开启就不调用 addslashes

- **7.Magic_quotes_gpc**
  - Magic_quotes_gpc 在稍微高点的版本默认都是 on，5.4 已经废除，',",\,NULL 会在前面添加上一个转义符
- **8.宽字节注入**
  - 数据库字符集 GBK 的宽字节注入

    数据库的连接方式不同，数据库与 PHP 的编码不一致，转换过程中可能存在

    错误方法：set names gbk
  - 转换字符集造成的宽字节注入

    从 gbk 转到 utf8

    参考漏洞：74cms 最新版 注入 8-9 wooyun-2014-063225.html

    从 utf8 转到 gbk，尽从 UTF8 转成 GBK 之后成了 %e5%5c，对 GET POST COOKIE 做了 addslashes，' 转义后为 \'->%5c %e5%5c5c' 两个 \ ，则引号出来

    参考漏洞：qibocms 下载系统 SQL 注入一枚 wooyun-2014-055842.html
- **9.解码导致**
  - 先提交 encode 的，那么就能不被转义，decode 后再带入查询，造成了注入，无视 GPC
  - urlencode
  - base64_decode
  - XML
  - Json_decode

    参考漏洞：qibocms B2b 注入一枚 wooyun-2014-053187.html

    参考链接：phpdisk V7 sql 注入 2 wooyun-2014-056822.html
- **10.变量覆盖**

  变量覆盖有 extract、parse_str、$$
  - extract

    extract($_POST) 直接从 POST 数组中取出变量，覆盖掉之前的一些变量，覆盖的话，一般是覆盖掉表前缀之类的。

    ```
    selet * from $pre_admin where xxx
    ```

    像这种就覆盖掉 $pre,然后直接补全语句注入

参考漏洞：qibocms 分类注入一枚可提升自己为管理 wooyun-2014-053189.html

参考漏洞：phpdisk V7 sql 注入 2 wooyun-2014-051734.html

- $$

参考漏洞：MetInfo 最新版 (5.2.4) 一处 SQL 盲注漏洞 wooyun-2014-055338.html

- **11.Replace**
  - 把 ' 替换成空，但是通过又全局有转义 ?< 单引号 ' 转义为 \' ,然后替换 ' 为空格，留下 \ ,注释掉 '，破坏原本的 SQL，用户提交一个 ' 全局转义成 \'，然后这过滤函数又会把 ' 替换成空，那么就留下 \ 导致可以吃掉一个单引号。

    需要 double query ,两处可控输入

    ```
    select * from c_admin where username=' admin\' and email=' inject#'
    ```

  - 有时会把 '" 都替换成空，然后提交之后去掉了 '，不把 ' 替换成空，但是 " 也会被转义，那么提交一个 " 就又剩下了一个转义符了。

    参考漏洞：PHPCMS 全版本通杀 SQL 注入漏洞 wooyun-2014-050636.html

  - 一些 replace 是用户可控的，就是说用户可以控制替换为空的内容

    ```
    $order_sn=str_replace($_GET['subject'],'',$_GET['out_trade_no']);
    ```

    这里因为会被转义，如果提交 ' 就变成 \'，并且这里替换为空的内容 get 来的，那就想办法把 \ 替换掉 addslashes 会对 ', ", \, NULL 转义，' 变成 \'，" 变成 \"，\ 变成 \\，NULL 变成 \0，提交 %00' 会被转义生成 \0\'，这时候再提交把 0 替换成空，那么就剩下 \\'，\\ 表示 \ 的转义，' 单引号也就成功出来了。

    参考漏洞：cmseasy 绕过补丁 SQL 注入一枚 wooyun-2014-053198.html

- **12.server 注入**

  -只对 GET POST COOKIE 进行 addslashes，没有对 SERVER 进行转义，一些 server 的变量，用户可控并写入数据库

```
QUERY_STRING , X_FORWARDED_FOR , CLIENT_IP , HTTP_HOST ,
ACCEPT_LANGUAGE
```

最常见的当然也就是 X_FORWARDED_FOR，一般是在 IP 函数中用户，没有验证 ip 是否合法，直接 return。

参考漏洞：Phpyun 注入漏洞二 wooyun-2014-068853.html

- 正则验证错误

  参考漏洞：CmsEasy 最新版本无限制 SQL 注射 wooyun-2014-062957.html

- **13.file 注入**
  - 全局只对 GET POST COOKIE 转义，遗漏了 files，，且不受 GPC，files 注入一般是因为上传，会把上传的名字带到 insert 当中入库

    参考漏洞：qibocms 黄页系统 SQL 注入一枚 wooyun-2014-065837.html
  - 在入库的时候对文件的名字进行转义，在获取后缀后再入库时对文件名转义了却没有对后缀转义也导致了注入

    参考漏洞：Supesite 前台注入 #2 (Insert) wooyun-2014-079041.html
- **14.未初始化造成的注入**

  php < 4.20 时，register_globals 默认都是 on，逐渐 register_globals 默认都是 off

  伪全局机制，遗漏了初始化

  参考漏洞：qibocms 地方门户系统注入一个问题 wooyun-2014-080867.html

  参考漏洞：qibocms 地方门户系统注入 wooyun-2014-080870.html

  参考漏洞：齐博地方门户系统 SQL 注入漏洞 wooyun-2014-079938.html

  参考漏洞：齐博整站/地方门户 SQL 注入漏洞 wooyun-2014-080259.html
- **15.数组中的key**

  判断 GPC 是否开启，如果 off 就对数组中的 value 进行 addslashes，没有对数组中的 key 进行转义，key 带入 sql，听说低版本的 php 对二维数组中的 key 就算 GPC ON 也不会转义

  参考漏洞：qibocms V7 整站系统最新版 SQL 注入一枚 & 另外一处能引入转义符的地方。 wooyun-2014-069746.html

  参考漏洞：qibocms 多个系统绕过补丁继续注入 wooyun-2014-070072.html

  参考漏洞：qibocms全部开源系统 Getshell wooyun-2014-070366.html

参考漏洞：Discuz 5.x 6.x 7.x 前台 SQL 注入漏洞一枚 wooyun-2014-071516.html

- **16.offset**

  $_GET[a] 提交的是一个数组，且含有一个 key 为 0，那么 $a 就是对应的这个 key 的 value，但是这里并没有强制要求为数组。

  提交一个字符串就为了 \，吃掉一个单引号，然后就在 $b 处写入 inject 可以注入

  参考漏洞：qibocms 地方门户系统 wooyun-2014-080875.html

- **17.第三方插件**

  常见的 uc_cencert / alipay / tenpay / chinabank

  - 默认 UC 里面都会 striplashes
  - uckey 默认的
  - uckey 这个常量没有初始化
  - uckey 可控

    参考漏洞：phpmps 注入 (可修改其他用户密码,官网成功)--UC wooyun-2014-060159.html

    参考漏洞：PHPEMS (在线考试系统) 设计缺陷 Getshell一枚(官网已 shell)--UC wooyun-2014-061135.html

    参考漏洞：最土团购注入一枚可直接提升自己为管理 & 无限刷钱。--CHINABANK wooyun-2014-058479.html

    参考漏洞：Destoon Sql 注入漏洞 2（有条件）--TENPAY wooyun-2014-055026.html

    参考漏洞：Destoon Sql 注入漏洞一枚（有条件）--TENPAY wooyun-2014-054947.html

    参考漏洞：CSDJCMS 程式舞曲最新版 Sql 一枚--TENPAY wooyun-2014-052363.html

- **18.数字型注入**

  - 一般数字型的都不会加单引号，$id 没被单引号且没有被强制类型转换

    参考漏洞：qibocms 地方门户系统 注入#3 wooyun-2014-080873.html

  - 不是一些数字型，忘记加单引号

```
$query=$_SGLOBAL['db']->query('select * from
'.tname('spacetags').' where itemid=\''.$itemid.'\' and
status = \''.status.'\'');
```

$itemid 首先带入查询中，被单引号，如果查询有接过才会带入到 delete 中，如果无接过就不执行 delete。在数据库中 itemid 中存储的是 int 类型，所以这里本意是只能提交数字型才能查询出结果，如果不是提交数字的话，那么就查询不出来结果，就不去执行下面的 delete 语句了。但是由于 mysql 的类型转换，因为这里存储的是 int 类型，所以 4xxxx 跟 4 是一样的。

```
$_SGLOBAL['db']->query('delete from
'.tname('spacetags').' where itemid='.$itemid.' and
tagid in ('.simplode($deletetagidarr).') and
status=\''.$status.'\'')
```

参考漏洞：Supesite 前台注入 #3 (Delete) wooyun-2014-079045.html

- PHP 弱类型语言

  参考漏洞：phpyun v3.2 (20141226) 两处注入。wooyun-2014-088872.html

- **19.二次注入**
  - 涉及到的是入库和出库

    在入库时经过全局转义，`insert into table(username) values (a\'');`

    入库后转义符就会消失，那么就是 a'，把这个查询出来，那么出库的就是 a'，如果再带入到了查询，那么就成功的引入了单引号导致了注入，很多时候数据库中存储的长度是有限制的。

参考漏洞：phpyun v3.2 (20141226) 两处注入。 wooyun-2014-088872.html

参考漏洞：qibocms 地方门户系统 二次注入#5 wooyun-2014-080877.html

参考漏洞：74cms (20140709) 二枚二次注入 wooyun-2014-068362.html

参考链接：Hdwiki 最新版二次注入一枚 wooyun-2014-067424.html

- **20.查询当中 key 可控**

  把 $_POST 带入到了查询函数，然后 foreach key ，foreach 出来的 key 做了查询中的 column。

  防止方法一般是把数据库中的 column 查询出来，然后 in_array 判断一下 $_POST 出来的 key 是否在数据库中的 column 中。

  参考漏洞：云人才系统 SQL 注入，绕过 WAF wooyun-2014-060166.html

  参考漏洞：Cmseasy SQL 注射漏洞之三 wooyun-2014-066221.html

- **21.striplashes**

  在全局 addslashes 后，在后面的文件中又 stripslashes 去掉了转义符，然后可以闭合单引号

  ```
  $_SESSION['flow_consignee'] =
  stripslashes_deep($consignee);
  ```

  参考漏洞：ecshop 全版本注入分析

  https://www.2cto.com/article/201301/182509.html

- **22.截取字符**
  - 会限制用户输入的长度，只截取一部分
  - cutstr($asd,32);，只允许输入 32 个字符，没有在截取字符的后面加其他字符

    提交一个 11111111111'，被转义后成 11111111111\'，绕后截取 32 个字符就是 11111111111\

    double query 的话，吃掉一个单引号，然后下一个连着的可控变量可以注入

    参考漏洞：Hdwiki (20141205) 存在 7 处 SQL 注入漏洞（含之前处理不当安全的漏洞）wooyun-2014-088004.html

- **23.注册 GLOBALS 变量**

把 GET POST COOKIE 循环出来，然后注册一个变量，这里不允许创建 GLOBALS 变量，如果设置了 REQUEST 的 GLOBALS，就直接退出 低版本 request order 是 GPC ,在 php5.3 以后 request order 默认成了 GP ，也就是 request 成了 get 和 post ，不包含 cookie，所以 $_REQUEST 里面就不包含 COOKIE 提交来的，而这里也把 COOKIE 循环出来，注册变量，所以这里在 COOKIE 里面提交 GLOBALS 就不会被检测出来，而且也成功注册了 GLOBALS 变量，所以再结合后面的一些些代码就造成了代码执行。
参考漏洞：Discuz! 某两个版本前台产品命令执行 wooyun-2014-080723.html

- **24.PDO 注入**
  - 查看 prepare() 硬编码的 string 是否可控
  - PDO 无法安全处理 order by 需求 bool-blind order by if([expr],id,name)

敏感逻辑

- **1.认证与会话**
  - 重置、找回密码
  - 人机验证绕过
  - session 固定
  - 密码存储、加密算法是否合理，rand()/mt_rand() 注意种子生成逻辑
  - sso / oauth / openid 使用合规
  - 注意 Cookie 中包含的可读数据
- **2.交易**
  - 条件竞争 select <> for update
  - 服务器端数据校验逻辑
- **3.投票、统计**
  - 未使用 REMOTE_ADDR 获取 ip 地址

PHP黑魔法

- 1.弱类型
  - string 转 int
    0e+ 纯数字优先转换
    字符截断 (int) / intval() wp content injection
    json_decode() 引入数字类型

- bypass
  - 使左右结果为弱类型的 0，构造 0e 科学计数法

    "1e3" bypass

    Magic hash
  - 传入空类型让函数报错返回 null

    如果结果来自数据库，让其取不到数据
  - 传入数组类型让函数报错返回 null

    string/array/null 类型可以从 GPC 传入
  - === 使左右结果为 true/false/null
  - is_numeric 传入 hex 编码的 str 返回 true
- 函数结果可控
  - in_array/intval/md5/strcmp<5.3/switch

- 2.正则匹配
  - preg_replace

    \0 $0

    php<7 /e,php<5.4用00截断构造/e,(regex) /e %00

    thinkphp url rce
  - preg_match

    php<=5.3 传入数组报错

总结

本文中提到的漏洞都来自于 wooyun 的历史数据，只提供了漏洞编号而没有提供地址，这个需要大家自行去寻找提供镜像备份的网站，或者下载备份文件自己本地查看，给大家带来的不变请大家见谅。

本文作者：**\xeb\xfe@信安之路**