DJANGO SECURITY

张龙

议题

1 安全那些事

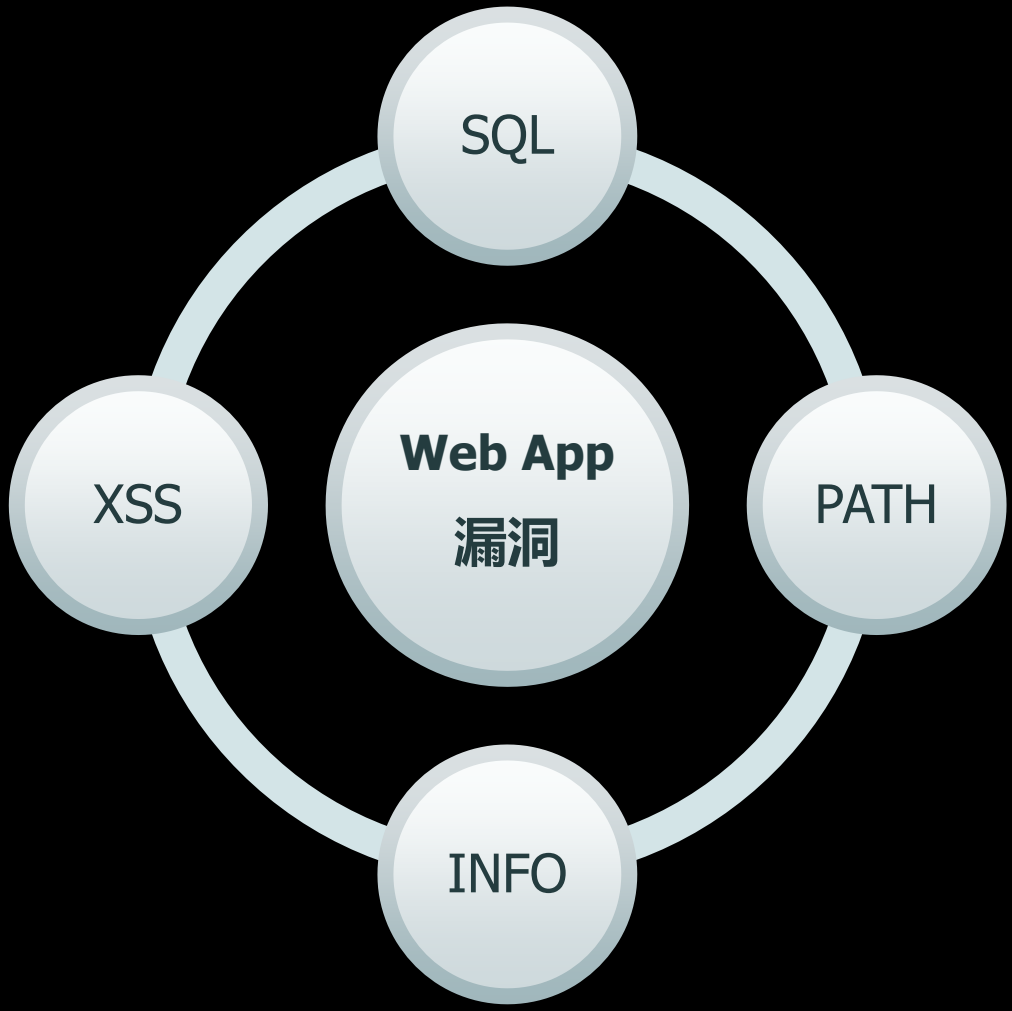
2 兵来将挡，水来土掩

3 道高一尺，魔高一丈

安全那些事

- Web开发框架就相当于web应用程序的操作系统，他决定了一个应用程序的模型结构和编程风格。
- 通过统一的入口点，可以做一些统一的安全防护、逻辑控制。
- 框架漏洞都不只是一种偶然，而是一种必然。

安全那些事



安全那些事

- 案例1

漏洞概要

缺陷编号： **WooYun-2011-03451**
漏洞标题： 百度django框架信息泄露漏洞(包括Mysql用户和密码)
相关厂商： **百度**
漏洞作者： **cnbird**
提交时间： 2011-11-28
公开时间： 2011-12-28
漏洞类型： 默认配置不当
危害等级： 中
自评Rank： 10
漏洞状态： 厂商已经确认
漏洞来源： **http://www.wooyun.org**
Tags标签： **第三方框架** **数据库密码泄漏** **django配置不当**

漏洞证明：

百度django框架配置不当信息泄露漏洞,包括Mysql用户和密码还有内网IP和域名。

```
&lt;type'array.array'&gt;: &lt;functior
'db':'cooder_dogfood',
'host':'tc-iit-icafedev01.vm.baidu.com',
'passwd':'iitmysql_cooder',
'port': 3306,
'use_unicode': True,
'user':'cooder_w'}</pre></td>
</tr>

</tbody>
```

www.wooyun.org

- 案例2

django/trunk/django/bin/compile-messages.py

r3590	r3592	
20	20	sys.stderr.write('processing file %s in %s\n' % (f, dirpath))
21	21	pf = os.path.splitext(os.path.join(dirpath, f))[0]
22		cmd = 'msgfmt -o "%s.mo" "%s.po"' % (pf, pf)
	22	# Store the names of the .mo and .po files in an environment
	23	# variable, rather than doing a string replacement into the
	24	# command, so that we can take advantage of shell quoting, to
	25	# quote any malicious characters/escaping.
	26	# See http://cyberelk.net/tim/articles/cmdline/ar01s02.html
	27	os.environ['djangocompilemo'] = pf + '.mo'
	28	os.environ['djangocompilepo'] = pf + '.po'
	29	cmd = 'msgfmt -o "$djangocompilemo" "$djangocompilepo"'
23	30	os.system(cmd)
24	31	

安全那些事

- 更多案例


English 中文 日本語
NSFOCUS
绿盟科技官方微博
首页
产品与解决方案
专业服务
客户支持
安全研究
工作机会
关于我们
安全漏洞
业界动态
紧急通告
研究成果
研究机构
安全漏洞
所有系统
所有类型
django
分页 (1) 共 4 条记录
2011-09-12 Django多个安全漏洞
2009-10-13 Django表单库正则表达式拒绝服务漏洞
2009-08-07 Django URL请求信息泄露漏洞
2007-10-30 Django i18n远程拒绝服务漏洞
分页 (1) 共 4 条记录
TOP
法律声明 | 联系我们 | 在线客服
© 2012绿盟科技 京公网安备110108002872号 京ICP证110355号

议题

1 安全那些事

2 兵来将挡，水来土掩

3 道高一尺，魔高一丈

兵来将挡，水来土掩

1. SQL注入

```
sql = "SELECT * FROM user_contacts WHERE username = '%s';" % username
```

```
SELECT * FROM user_contacts WHERE username = '' OR 'a' = 'a';
```

•解决方案

优先使用ORM模型，其次用占位符，
再次绝不信任用户提交的数据，在传递给SQL语句时，总是转义它。

```
foo.get_list(bar__exact="' OR 1=1")
```

```
SELECT * FROM foos WHERE bar = '\' OR 1=1'
```

1. SQL注入

例外

1. 传给 **extra()** 方法的 **where** 参数。

✘ Entry.objects.extra(where=["headline='%s'" % name])

✓ Entry.objects.extra(where=['headline=%s'], params=[name])

2. 使用底层数据库API的查询。

3. 绑定参数不能够作为标识符（表或列名等）。

提示

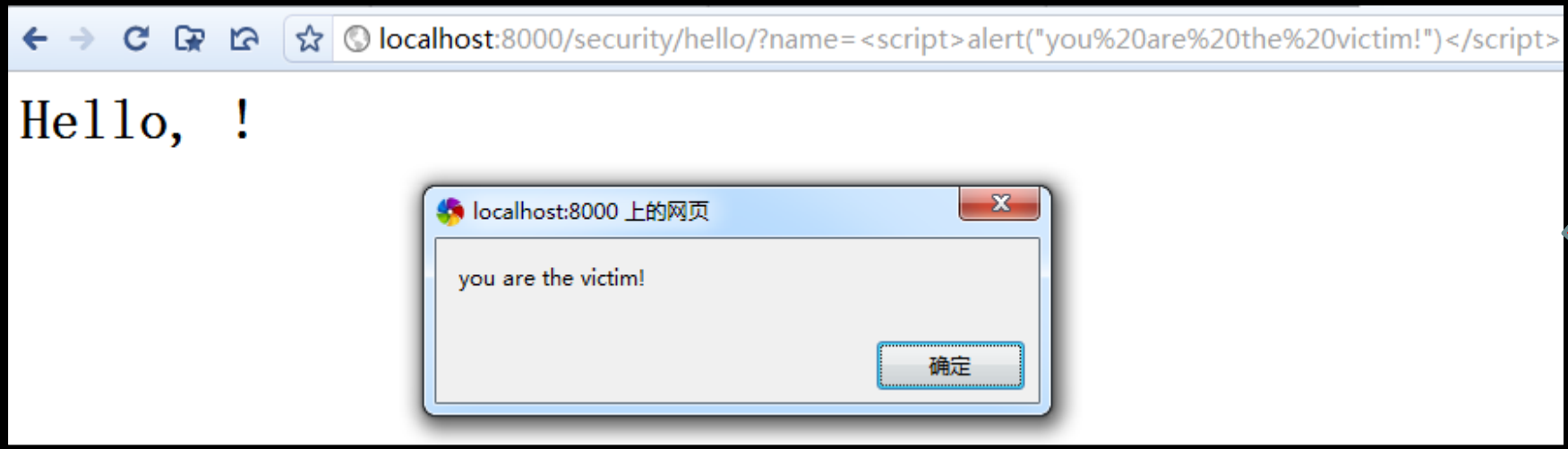
1. 千万不要拼接查询字符串

2. 跨站点脚本 (XSS)

```
from django.http import HttpResponse

def say_hello(request):
    name = request.GET.get('name', 'world')
    return HttpResponse('<h1>Hello, %s!</h1>' % name)
```

http://localhost:8000/security/hello/?name=**<script>alert(1)</script>**



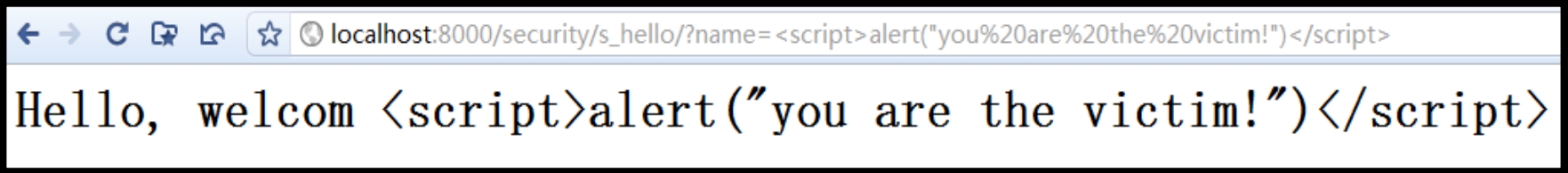
2. 跨站点脚本 (XSS)

•解决方案

总是转义可能来自某个用户的任何内容。

```
# views.py
from django.shortcuts import render_to_response
def s_say_hello(request):
    name = request.GET.get('name', 'world')
    return render_to_response('hello.html', {'name': name})

# hello.html
<h1>Hello, {{ name }}!</h1>
```

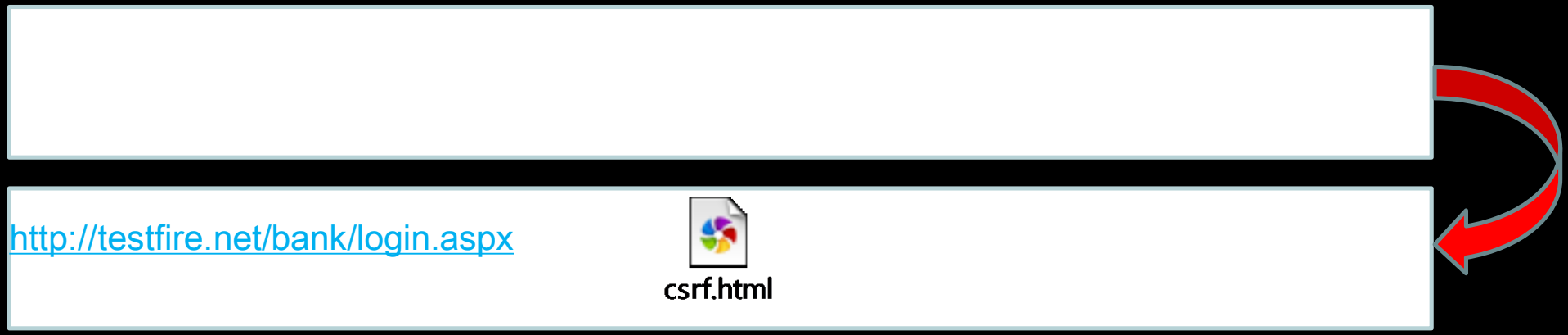


2. 跨站点脚本 (XSS)

提示

1. 用render_to_response吧，何必再造轮子

3. 伪造跨站点请求



•解决方案

开启CSRF保护

1. 添加‘django.middleware.csrf.CsrfViewMiddleware’中间件或者在特定的Views上添加@csrf_protect注解
2. 在POST表单内添加{% csrf_token %}tag
3. 确保使用了'django.core.context_processors.csrf'上下文处理器
   a. 使用RequestContext
   b. 手动导入并使用处理器生成CSRF令牌并添加到模板上下文中

3. 伪造跨站点请求

•原理

1. 向所有当前处理的请求的POST表单增添一个隐藏的表单字段csrfmiddlewaretoken，值为session id加上一个密钥的散列值。
2. 对POST请求检查是否存在 csrfmiddlewaretoken 及其是否正确。 如果否，返回403终止请求

提示

- CsrfMiddleware只针对 HTTP POST 请求
- 未使用会话cookie的POST请求无法受到保护
- 只有text/html或application/xml+xhtml的页面才会被修改
- 需要Django的会话框架。使用自定义会话或者身份验证框架手动管理会话cookies的无效

4. 会话伪造/劫持

•解决方案

请遵守基本准则：

1. 勿在URL中包含任何session信息（Django的session框架不容许session包含在URL中）
2. 勿直接在cookie中保存数据。存储一个在后台映射到session数据存储的session ID。
3. 模板中显示session数据，务必要对其进行转义。
4. 任何可能的地方都要防止攻击者进行session欺骗。
5. Django内置了保护措施来抵御暴力会话攻击

4. 会话伪造/劫持

局限

◆ 对中间人攻击无能为力

提示

✓ 通过HTTPS来提供网站服务。
✓ 若使用SSL，设置SESSION_COOKIE_SECURE=True

5. 邮件头部注入

"hello\ncc:spamvictim@example.com"

To: hardcoded@example.com
Subject: hello
cc: spamvictim@example.com

•解决方案

总是校验或者转义用户提交的内容

\ncc:zhanglong@intra.nsfocus.com

5. 邮件头部注入

提示

若不使用Django内建邮件功能来发送邮件，需要确保包含在邮件头部的换行符能够引发错误或者被去掉。

6. 目录遍历

filename

若filename=../../../../../etc/passwd?

•解决方案

永远 不要在编写**可以读取任何位置上的文件**的代码！

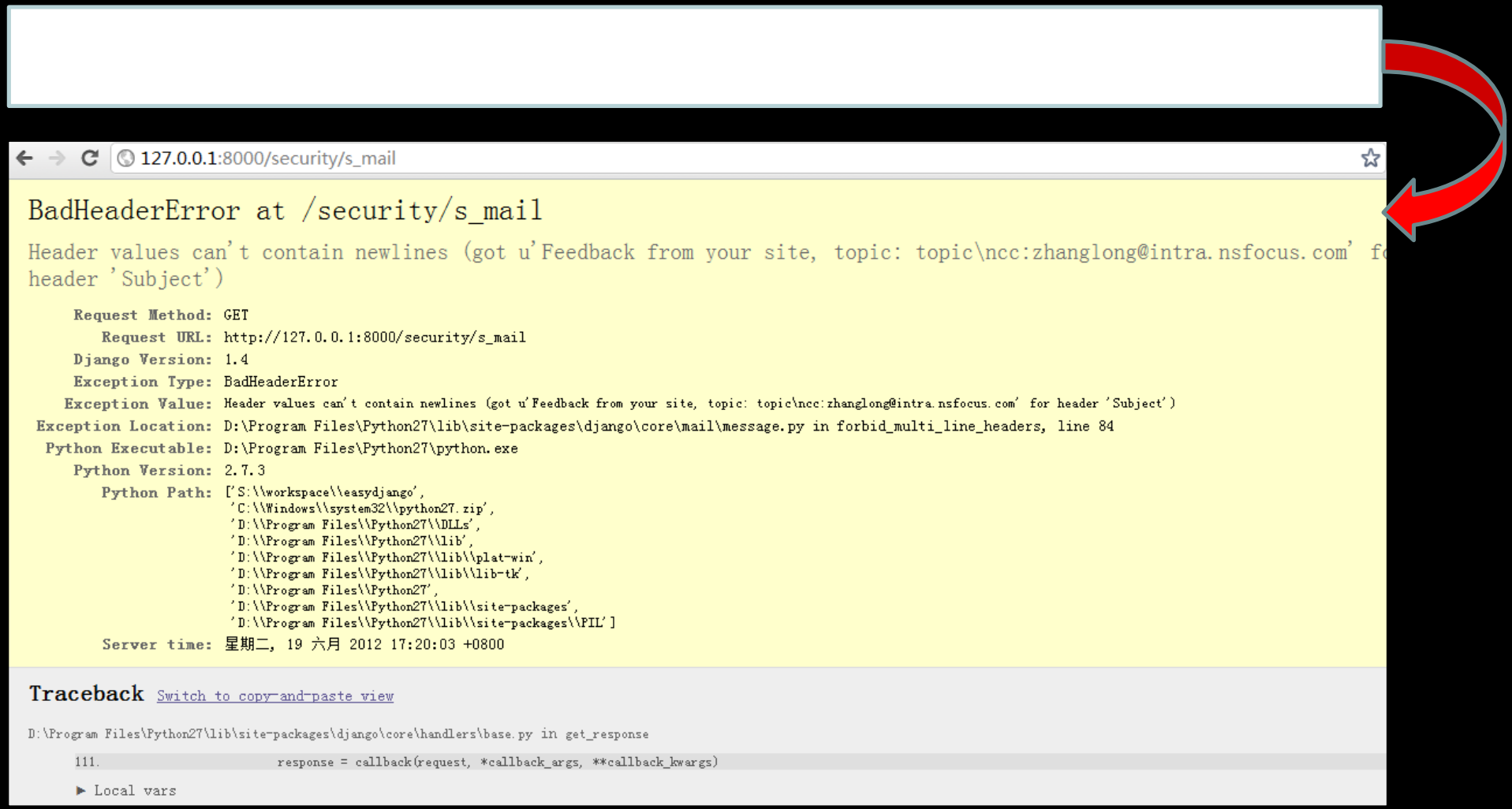
```
url(r'^security/(?P<path>.*)$', 'django.views.static.serve',
{'document_root' : 'S:\workspace\easydjango\easydjango\distinct\static'})
```

6. 目录遍历

提示

- 用static.serve读取文件要安全得多
- URLconf抽象层的使用，意味着不经过你明确的指定，Django决不会装载代码

7. 暴露错误消息

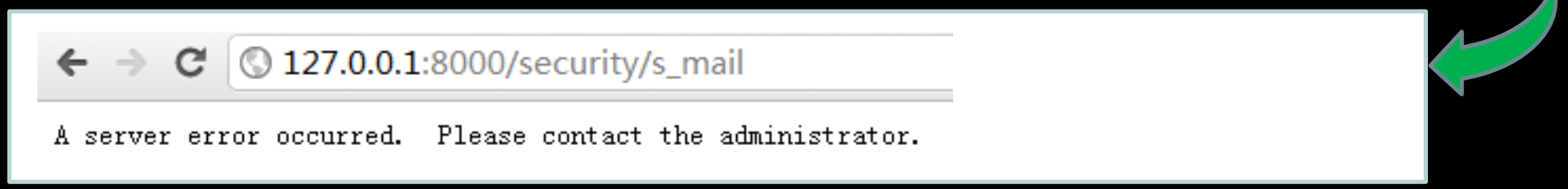

127.0.0.1:8000/security/s_mail
BadHeaderError at /security/s_mail
Header values can't contain newlines (got u'Feedback from your site, topic: topic\ncc:zhanglong@intra.nsfocus.com' fo
header 'Subject')
Request Method: GET
Request URL: http://127.0.0.1:8000/security/s_mail
Django Version: 1.4
Exception Type: BadHeaderError
Exception Value: Header values can't contain newlines (got u'Feedback from your site, topic: topic\ncc:zhanglong@intra.nsfocus.com' for header 'Subject')
Exception Location: D:\Program Files\Python27\lib\site-packages\django\core\mail\message.py in forbid_multi_line_headers, line 84
Python Executable: D:\Program Files\Python27\python.exe
Python Version: 2.7.3
Python Path: ['S:\\workspace\\easydjango',
'C:\\Windows\\system32\\python27.zip',
'D:\\Program Files\\Python27\\DLLs',
'D:\\Program Files\\Python27\\lib',
'D:\\Program Files\\Python27\\lib\\plat-win',
'D:\\Program Files\\Python27\\lib\\lib-tk',
'D:\\Program Files\\Python27',
'D:\\Program Files\\Python27\\lib\\site-packages',
'D:\\Program Files\\Python27\\lib\\site-packages\\PIL']
Server time: 星期二, 19 六月 2012 17:20:03 +0800
Traceback
Switch to copy-and-paste view
D:\Program Files\Python27\lib\site-packages\django\core\handlers\base.py in get_response
111. response = callback(request, *callback_args, **callback_kwargs)
▶ Local vars

7. 暴露错误消息

•解决方案

站点的访问者永远不应该看到与应用相关的出错消息

Debug=False



7. 暴露错误消息

提示

Apache和mod_python下，要保证Apache的配置文件关闭PythonDebug Off

漏网之鱼

点击劫持

•解决方案

```
# Uncomment the next line for simple clickjacking protection:
'django.middleware.clickjacking.XFrameOptionsMiddleware',
```

警钟长鸣

- 小心你的源代码
- 谨慎处理用户上传的文件
- 用django插件或者web服务器模块处理暴力破解攻击
- 该保密的要保密，入SECRET_KEY
- 透过防火墙访问数据库和缓存系统

议题

1 安全那些事

2 兵来将挡，水来土掩

3 道高一尺，魔高一丈

这些已经足够了？

回头看看开始的案例
木有银弹！！
……

课后作业

作业：

谢谢！