



Web 应用程序报告

该报告包含有关 **web** 应用程序的重要安全信息。

安全报告

该报告由 IBM Security AppScan Standard 创建 9.0.2.1, 规则: 1932
扫描开始时间: 2016/4/26 20:30:16

目录

介绍

- 一般信息
- 登陆设置

管理综合报告

- 问题类型
- 有漏洞的 URL
- 修订建议
- 安全风险
- 原因
- WASC 威胁分类

按问题类型分类的问题

- 跨站点脚本编制 ②
- 链接注入（便于跨站请求伪造） ①
- 通过框架钓鱼 ①
- Missing "Content-Security-Policy" header ③
- Missing "X-Content-Type-Options" header ③
- Missing "X-XSS-Protection" header ③
- 发现内部 IP 泄露模式 ①

修订建议

- 查看危险字符注入的可能解决方案
- Config your server to use the "Content-Security-Policy" header
- Config your server to use the "X-Content-Type-Options" header
- Config your server to use the "X-XSS-Protection" header
- 除去 Web 站点中的内部 IP 地址

咨询

- 跨站点脚本编制
- 链接注入（便于跨站请求伪造）
- 通过框架钓鱼
- Missing "Content-Security-Policy" header
- Missing "X-Content-Type-Options" header
- Missing "X-XSS-Protection" header
- 发现内部 IP 泄露模式

应用程序数据

- cookie
- JavaScript
- 参数
- 注释
- 已访问的 URL
- 失败的请求
- 已过滤的 URL

介绍

该报告包含由 IBM Security AppScan Standard 执行的 Web 应用程序安全性扫描的结果。

高严重性问题:	2
中等严重性问题:	2
低严重性问题:	9
参考严重性问题:	1
报告中包含的严重性问题总数:	14
扫描中发现的严重性问题总数:	14

一般信息

扫描文件名称:	mfs0426002
扫描开始时间:	2016/4/26 20:30:16
测试策略:	Default
主机	10.62.233.181
操作系统:	Unknown
Web 服务器:	Unknown
应用程序服务器:	Any








登陆设置

登陆方法:	记录的登录
并发登陆:	已启用
JavaScript 执行文件:	已禁用
会话中检测:	已启用
会话中模式:	
跟踪或会话标识 cookie:	
跟踪或会话标识参数:	
登陆序列:	

管理综合报告




问题类型 7

TOC

问题类型		问题的数量
高	跨站点脚本编制	2 
中	链接注入（便于跨站请求伪造）	1 
中	通过框架钓鱼	1 
低	Missing "Content-Security-Policy" header	3 
低	Missing "X-Content-Type-Options" header	3 
低	Missing "X-XSS-Protection" header	3 
参	发现内部 IP 泄露模式	1 






有漏洞的 URL 3

TOC

URL		问题的数量
高	http://10.62.233.181:9425/mfs.cgi	8 
低	http://10.62.233.181:9425/acidtab.js	3 
低	http://10.62.233.181:9425/index.html	3 





修订建议 5

TOC

修复任务		问题的数量
高	查看危险字符注入的可能解决方案	4 
低	Config your server to use the "Content-Security-Policy" header	3 
低	Config your server to use the "X-Content-Type-Options" header	3 
低	Config your server to use the "X-XSS-Protection" header	3 
低	除去 Web 站点中的内部 IP 地址	1 


安全风险 ④

TOC

风险	问题的数量
高 可能会窃取或操纵客户会话和 cookie，它们可能用于模仿合法用户，从而使黑客能够以该用户身份查看或变更用户记录以及执行事务	3 
中 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息	11 
中 可能会在 Web 服务器上上载、修改或删除 Web 页面、脚本和文件	1 
低 可能会收集有关 Web 应用程序的敏感信息，如用户名、密码、机器名和/或敏感文件位置	10 




原因 ②

TOC

原因	问题的数量
高 未对用户输入正确执行危险字符清理	4 
低 Web 应用程序编程或配置不安全	10 

WASC 威胁分类

TOC

威胁	问题的数量
跨站点脚本编制	2 
内容电子欺骗	2 
信息泄露	10 

按问题类型分类的问题

跨站点脚本编制	
严重性:	高
CVSS 分数:	7.5
URL:	http://10.62.233.181:9425/mfs.cgi
实体:	mfs.cgi (Page)
风险:	可能会窃取或操纵客户会话和 cookie，它们可能用于模仿合法用户，从而使黑客能够以该用户身份查看或变更用户记录以及执行事务
原因:	未对用户输入正确执行危险字符清理
固定值:	查看危险字符注入的可能解决方案

差异: 参数 从以下位置进行控制: mfsmaster 至:

`%3E%22%27%3E%3Cscript%3Ealert%28%29%3C%2Fscript%3E`

参数 从以下位置进行控制: 9421 至: `%3E%22%27%3E%3Cscript%3Ealert%28%29%3C%2Fscript%3E`

参数 从以下位置进行控制: 9419 至: `%3E%22%27%3E%3Cscript%3Ealert%28%29%3C%2Fscript%3E`

参数 从以下位置进行控制: MooseFS 至:

`%3E%22%27%3E%3Cscript%3Ealert%28%29%3C%2Fscript%3E`

推理: 测试结果似乎指示存在脆弱性，因为 Appscan 在响应中成功嵌入了脚本，在用户浏览器中装入页面时将执行该脚本。

测试请求和响应:

```
GET /mfs.cgi?
masterhost=%3E%22%27%3E%3Cscript%3Ealert%28%29%3C%2Fscript%3E&masterport=%3E%22%27...
HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/index.html
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR
2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)
```

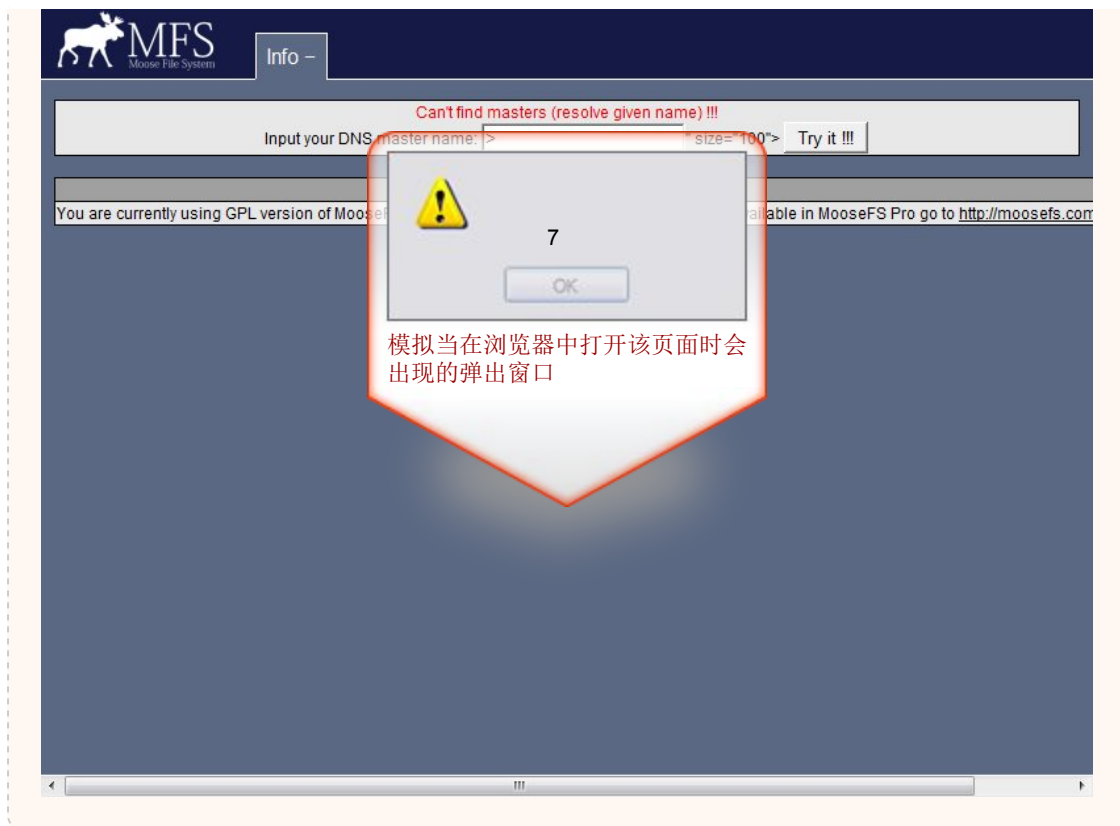
```

HTTP/1.1 200 Ok
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>MFS Info (&quot;&apos;&gt;&lt;&lt;script&gt;alert(7)&lt;/script&gt;</title>
<link rel="stylesheet" href="mfs.css" type="text/css" />
<script src="acidtab.js" type="text/javascript"></script>
</head>
<body>
<div id="header">
<table class="HDR" cellpadding="0" cellspacing="0" border="0">
<tr>
<td class="LOGO"><a href="http://www.moosefs.org"></a></td>
<td class="MENU"><table class="MENU" cellspacing="0">
<tr>
<td class="LUS">Info &#8722;</td>
</tr>
</table></td>
<td class="FILLER" style="white-space:nowrap;">
</td>
</tr>
</table>
</div>
<div id="container">
<table class="FR" cellspacing="0">
<tr>
<td align="center">
<span class="ERROR">Can't find masters (resolve given name) !!!</span><br>
/>
<form method="GET">
Input your DNS master name: <input type="text" name="masterhost"
value="&quot;&apos;&gt;&lt;&lt;script&gt;alert(7)</script>" size="100">
<input type="hidden" name="masterport"
value="&quot;&apos;&gt;&lt;&lt;script&gt;alert(7)&lt;/script&gt;">
<input type="hidden" name="mastercontrolport"
value="&quot;&apos;&gt;&lt;&lt;script&gt;alert(7)&lt;/script&gt;">
<input type="hidden" name="mastername"
value="&quot;&apos;&gt;&lt;&lt;script&gt;alert(7)&lt;/script&gt;">
<input type="submit" value="Try it !!!">
</form>
</td>
</tr>
</table>
<br/>
<table class="FR" cellspacing="0">
<tr><th>Notice</th></tr>
<tr><td>You are currently using GPL version of MooseFS. If you want to find out what great
features are available in MooseFS Pro go to <a
href="http://moosefs.com/products.html">http://moosefs.com/products.html</a></td></tr>
</table>
</div> <!-- end of container -->
</body>
</html>

```

测试响应



跨站点脚本编制

严重性: **高**

CVSS 分数: 7.5

URL: <http://10.62.233.181:9425/mfs.cgi>

实体: masterhost (Parameter)

风险: 可能会窃取或操纵客户会话和 cookie，它们可能用于模仿合法用户，从而使黑客能够以该用户身份查看或变更用户记录以及执行事务

原因: 未对用户输入正确执行危险字符清理

固定值: [查看危险字符注入的可能解决方案](#)

差异: 参数 从以下位置进行控制: `mfsmaster` 至: `mfsmaster"/><script>alert(132)</script>`

推理: 测试结果似乎指示存在脆弱性，因为 Appscan 在响应中成功嵌入了脚本，在用户浏览器中装入页面时将执行该脚本。

测试请求和响应:

```
GET /mfs.cgi?masterhost=mfsmaster"/><script>alert(132)
</script>&masterport=9421&mastercontro... HTTP/1.1
```

```

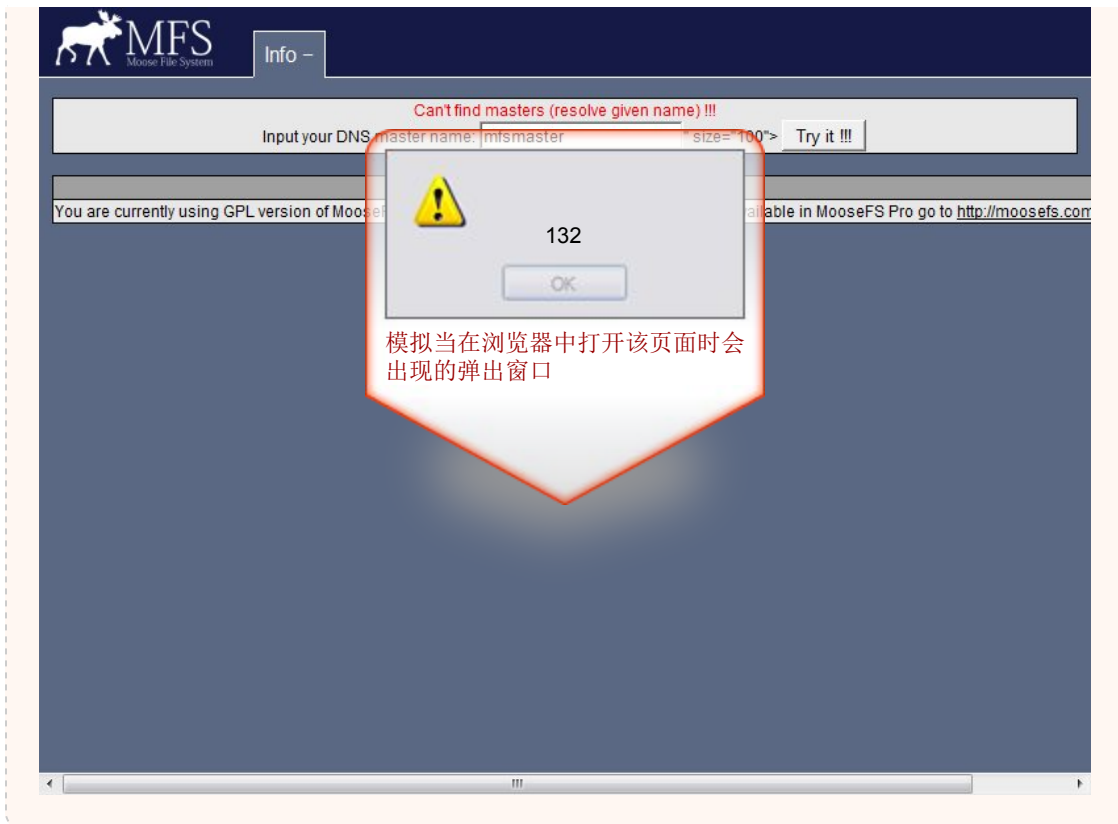
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/index.html
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>MFS Info (MooseFS)</title>
<link rel="stylesheet" href="mfs.css" type="text/css" />
<script src="acidtab.js" type="text/javascript"></script>
</head>
<body>
<div id="header">
<table class="HDR" cellpadding="0" cellspacing="0" border="0">
<tr>
<td class="LOGO"><a href="http://www.moosefs.org"></a></td>
<td class="MENU"><table class="MENU" cellspacing="0">
<tr>
<td class="LUS">Info &#8722;</td>
</tr>
</table></td>
<td class="FILLER" style="white-space:nowrap;">
</td>
</tr>
</table>
</div>
<div id="container">
<table class="FR" cellspacing="0">
<tr>
<td align="center">
<span class="ERROR">Can't find masters (resolve given name) !!!</span><br>
</td>
<td>
<form method="GET">
Input your DNS master name: <input type="text" name="masterhost" value="mfsmaster"/><script>alert(132)</script>" size="100">
<input type="hidden" name="masterport" value="9421">
<input type="hidden" name="mastercontrolport" value="9419">
<input type="hidden" name="mastername" value="MooseFS">
<input type="submit" value="Try it !!!">
</form>
</td>
</tr>
</table>
<br/>
<table class="FR" cellspacing="0">
<tr><th>Notice</th></tr>
<tr><td>You are currently using GPL version of MooseFS. If you want to find out what great features are available in MooseFS Pro go to <a href="http://moosefs.com/products.html">http://moosefs.com/products.html</a></td></tr>
</table>
</div> <!-- end of container -->
</body>
</html>

```

测试响应



问题 1 / 1

TOC

链接注入（便于跨站请求伪造）

严重性： 中

CVSS 分数： 6.4

URL： http://10.62.233.181:9425/mfs.cgi

实体： masterhost (Parameter)

风险： 可能会窃取或操纵客户会话和 **cookie**，它们可能用于模仿合法用户，从而使黑客能够以该用户身份查看或变更用户记录以及执行事务
可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会在 **Web** 服务器上上载、修改或删除 **Web** 页面、脚本和文件

原因： 未对用户输入正确执行危险字符清理**固定值：** [查看危险字符注入的可能解决方案](#)**差异：** 参数 从以下位置进行控制： `mfsmaster` 至：`%22%27%3E%3CIMG+SRC%3D%22%2FWF_XSRF129.html%22%3E`**推理：** 测试结果似乎指示存在脆弱性，因为测试响应包含文件“WF_XSRF.html”的链接。**测试请求和响应：**

```
GET /mfs.cgi?
masterhost=%22%27%3E%3CIMG+SRC%3D%22%2FWF_XSRF129.html%22%3E&masterport=9421&ma...
HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/index.html
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR
2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html; charset=UTF-8

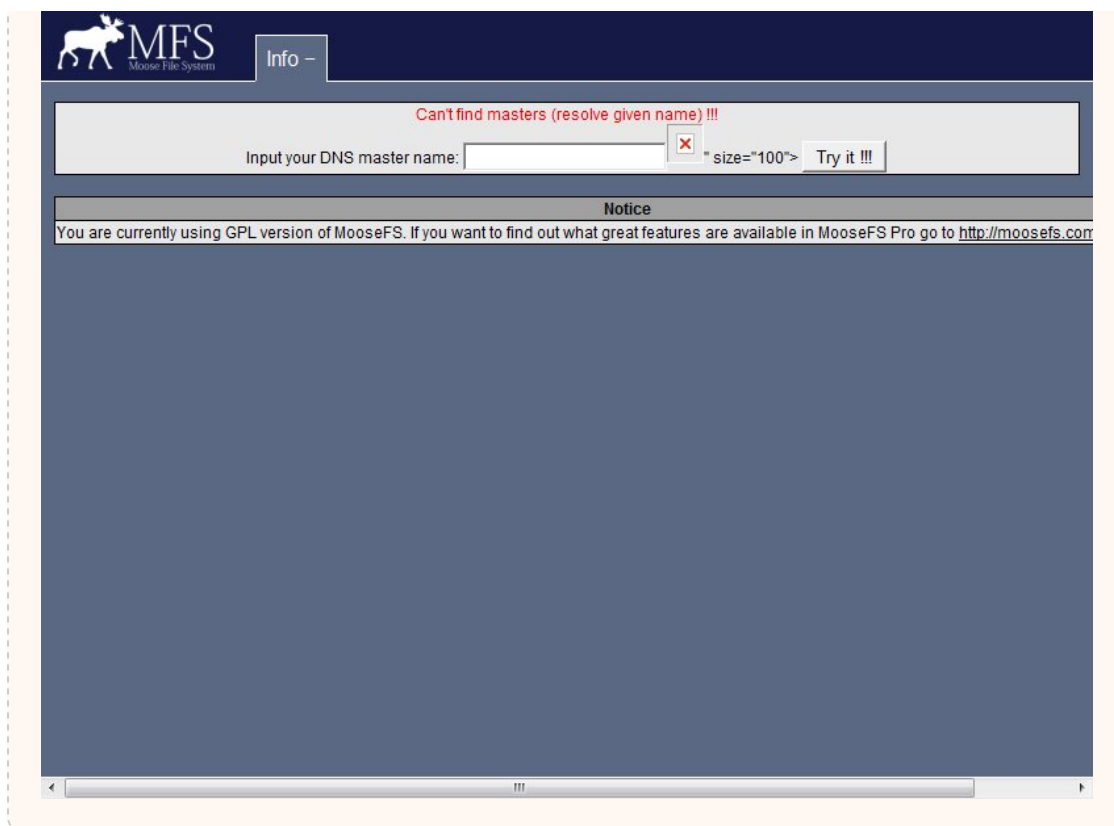
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>MFS Info (MooseFS)</title>
<link rel="stylesheet" href="mfs.css" type="text/css" />
<script src="acidtab.js" type="text/javascript"></script>
</head>
<body>
<div id="header">
<table class="HDR" cellpadding="0" cellspacing="0" border="0">
```

```

<tr>
<td class="LOGO"><a href="http://www.moosefs.org"></a></td>
<td class="MENU"><table class="MENU" cellspacing="0">
<tr>
<td class="LUS">Info &#8722;</td>
</tr>
</table></td>
<td class="FILLER" style="white-space:nowrap;">
</td>
</tr>
</table>
</div>
<div id="container">
<table class="FR" cellspacing="0">
<tr>
<td align="center">
<span class="ERROR">Can't find masters (resolve given name) !!!</span><br
/>
<form method="GET">
Input your DNS master name: <input type="text" name="masterhost"
value=""><IMG SRC="/WF_XSRF129.html">" size="100">
<input type="hidden" name="masterport" value="9421">
<input type="hidden" name="mastercontrolport" value="9419">
<input type="hidden" name="mastername" value="MooseFS">
<input type="submit" value="Try it !!!">
</form>
</td>
</tr>
</table>
<br/>
<table class="FR" cellspacing="0">
<tr><th>Notice</th></tr>
<tr><td>You are currently using GPL version of MooseFS. If you want to find out what great
features are available in MooseFS Pro go to <a
href="http://moosefs.com/products.html">http://moosefs.com/products.html</a></td></tr>
</table>
</div> <!-- end of container -->
</body>
</html>

```

测试响应



中 通过框架钓鱼 1

TOC

问题 1 / 1

TOC

通过框架钓鱼

严重性: 中

CVSS 分数: 6.4

URL: http://10.62.233.181:9425/mfs.cgi

实体: masterhost (Parameter)

风险: 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息

原因: 未对用户输入正确执行危险字符清理

固定值: 查看危险字符注入的可能解决方案

差异: 参数 从以下位置进行控制: mfsmaster 至:

mfsmaster%27%22%3E%3Ciframe+id%3D100+src%3Dhttp%3A%2F%2Fdemo.testfire.net%2Fphishing.html%3E

推理: 测试结果似乎指示存在脆弱性, 因为测试响应包含 URL "http://demo.testfire.net/phishing.html" 的 frame/iframe。

测试请求和响应:

```
GET /mfs.cgi?
masterhost=mfsmaster%27%22%3E%3Ciframe+id%3D100+src%3Dhttp%3A%2F%2Fdemo.testfire.net%2F...
HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/index.html
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR
2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>MFS Info (MooseFS)</title>
<link rel="stylesheet" href="mfs.css" type="text/css" />
<script src="acidtab.js" type="text/javascript"></script>
</head>
<body>
<div id="header">
<table class="HDR" cellpadding="0" cellspacing="0" border="0">
<tr>
<td class="LOGO"><a href="http://www.moosefs.org"></a></td>
<td class="MENU"><table class="MENU" cellspacing="0">
<tr>
<td class="LUS">Info &#8722;</td>
</tr>
</table></td>
<td class="FILLER" style="white-space:nowrap;">
</td>
</tr>
</table>
</div>
<div id="container">
<table class="FR" cellspacing="0">
<tr>
<td align="center">
<span class="ERROR">Can't find masters (resolve given name) !!!</span><br>
/>
<form method="GET">
Input your DNS master name: <input type="text" name="masterhost"
value="mfsmaster"><iframe id=100 src=http://demo.testfire.net/phishing.html>" size="100">
<input type="hidden" name="masterport" value="9421">
<input type="hidden" name="mastercontrolport" value="9419">
<input type="hidden" name="mastername" value="MooseFS">
<input type="submit" value="Try it !!!">
</form>
</td>
</tr>
</table>
<br/>
<table class="FR" cellspacing="0">
<tr><th>Notice</th></tr>
<tr><td>You are currently using GPL version of MooseFS. If you want to find out what great
features are available in MooseFS Pro go to <a
href="http://moosefs.com/products.html">http://moosefs.com/products.html</a></td></tr>
</table>
</div> <!-- end of container -->
</body>
</html>
```

测试响应



问题 1 / 3

TOC

Missing "Content-Security-Policy" header

严重性: 低

CVSS 分数: 5.0

URL: http://10.62.233.181:9425/index.html

实体: index.html (Page)

风险: 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息, 如用户名、密码、机器名和/或敏感文件位置

原因: Web 应用程序编程或配置不安全

固定值: [Config your server to use the "Content-Security-Policy" header](#)

差异:

推理: AppScan detected that the Content-Security-Policy response header is missing, which increases exposure to various cross-site injection attacks

测试请求和响应:

```
GET /index.html HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html
Content-Length: 537

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>MFS</title>
  <script type="text/javascript">
    document.location.href="mfs.cgi"
    // the above uses default parameter values; full invocation example is shown below
    //      document.location.href="mfs.cgi?
masterhost=mfsmaster&masterport=9421&mastercontrolp...
  </script>
</head>
<body>
</body>
</html>
```

Missing "Content-Security-Policy" header

严重性:

低

CVSS 分数: 5.0

URL: http://10.62.233.181:9425/mfs.cgi

实体: mfs.cgi (Page)

风险: 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息, 如用户名、密码、机器名和/或敏感文件位置

原因: Web 应用程序编程或配置不安全

固定值: Config your server to use the "Content-Security-Policy" header

差异:

推理: AppScan detected that the Content-Security-Policy response header is missing, which increases exposure to various cross-site injection attacks

测试请求和响应:

```
GET /mfs.cgi?sections=CS HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/mfs.cgi
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>MFS Info (MooseFS)</title>
<link rel="stylesheet" href="mfs.css" type="text/css" />
<script src="acidtab.js" type="text/javascript"></script>
</head>
<body>
<div id="header">
<table class="HDR" cellpadding="0" cellspacing="0" border="0">
<tr>
<td class="LOGO"><a href="http://www.moosefs.org"></a></td>
<td class="MENU"><table class="MENU" cellspacing="0">
<tr>
<td class="UU"><a href="mfs.cgi?sections=IN">Info</a> <a href="mfs.cgi?sections=CS%7CIN">+</a>
</td>
<td class="US">Servers &#8722;</td>
<td class="SU"><a href="mfs.cgi?sections=HD">Disks</a> <a href="mfs.cgi?sections=CS%7CHD">+</a>
</td>
<td class="UU"><a href="mfs.cgi?sections=EX">Exports</a> <a href="mfs.cgi?sections=CS%7CEX">+</a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MS">Mounts</a> <a href="mfs.cgi?sections=CS%7CMS">+</a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MO">Operations</a> <a href="mfs.cgi?sections=CS%7CMO">+</a>
</td>
</tr>
</table>
</td>
</tr>
</table>
</div>
</body>
</html>
```

```

</a></td>
<td class="UU"><a href="mfs.cgi?sections=QU">Quotas</a> <a href="mfs.cgi?sections=CS%7CQU">+</a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MC">Master Charts</a> <a href="mfs.cgi?
sections=CS%7CMC">+</a></td>
<td class="LUU"><a href="mfs.cgi?sections=CC">Server Charts</a> <a href="mfs.cgi?
sections=CS%7CCC">+</a></td>
</tr>
</table></td>
<td class="FILLER" style="white-space:nowrap;">
</td>
</tr>
</table>
</div>
<div id="container">
<table class="acid_tab acid_tab_zebra_C1_C2 acid_tab_storageid_mfscs" cellspacing="0">
<tr><th colspan="17">Chunk Servers</th></tr>
<tr>
<th rowspan="2" class="acid_tab_enumerate">#</th>
<th rowspan="2">host</th>
<th rowspan="2">ip</th>
<th rowspan="2">port</th>
<th rowspan="2">id</th>
<th rowspan="2">version</th>
<th rowspan="2">load</th>
<th rowspan="2">maintenance</th>
<th colspan="4">'regular' hdd space</th>
<th colspan="4">'marked for removal' hdd space</th>

</tr>
<tr>
<th>chunks</th>
<th>used</th>
<th>total</th>
<th class="PROGBAR">% used</th>
<th>chunks</th>
<th>used</th>
<th>total</th>
<th class="PROGBAR">% used</th>

</tr>
<tr>
<td align="right"></td>
<td align="left">(unresolved)</td>
<td align="center"><span class="sortkey">010_093_211_036 </span>10.93.211.36</td>
<td align="center">9422</td>
<td align="center">1</td>
<td align="center"><span class="sortkey">00002_000_088_1 </span><span
class="OKVERSION">2.0.88</span></td>
<td align="right">0</td>
<td align="center"><a href="mfs.cgi?
sections=CS&CSmaintenanceon=10.93.211.36%3A9422">switch on</a></td>
<td align="right">0</td><td align="right"><span class="sortkey">449912832 </span>
<a style="cursor:default" title="449 912 832 B">429&nbsp;MiB</a></td><td align="right"><span
class="sortkey">30101889024 </span><a style="cursor:default" title="30 101 889 024
B">28&nbsp;GiB</a></td>
<td align="center"><span class="sortkey">1.4946332160 </span><div class="PROGBOX"
style="width:200px;"><div class="PROGCOVER" style="width:98.51%;"></div><div class="PROGAVG"
style="width:1.49%"></div><div class="PROGVALUE"><a style="cursor:default" title="1.4946% =
(avg+0.0000%)">1.49</a></div></div></td>
<td align="right">0</td><td align="right"><span class="sortkey">0 </span><a
style="cursor:default" title="0 B">0&nbsp;B</a></td><td align="right"><span class="sortkey">0
</span><a style="cursor:default" title="0 B">0&nbsp;B</a></td>
<td align="center"><span class="sortkey">-1 </span><div class="PROGBOX"
style="width:200px;"><div class="PROGCOVER" style="width:100%;"></div><div class="PROGVALUE">-
</div></div></td>
</tr>
</table>
<br/>
<table class="acid_tab acid_tab_zebra_C1_C2 acid_tab_storageid_mfsmbl" cellspacing="0">
<tr><th colspan="4">Metadata Backup Loggers</th></tr>
<tr>
<th class="acid_tab_enumerate">#</th>
<th>host</th>
<th>ip</th>
<th>version</th>

</tr>
<tr>
<td align="right"></td><td align="left">(unresolved)</td><td align="center"><span
class="sortkey">010_093_216_120 </span>10.93.216.120</td><td align="center"><span

```

```

class="sortkey">00002_000_088_1 </span><span class="OKVERSION">2.0.88</span></td>
</tr>
</table>
</div> <!-- end of container -->
</body>
</html>

```

问题 3 / 3

TOC

Missing "Content-Security-Policy" header

严重性: 低

CVSS 分数: 5.0

URL: http://10.62.233.181:9425/acidtab.js

实体: acidtab.js (Page)

风险: 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息，如用户名、密码、机器名和/或敏感文件位置

原因: Web 应用程序编程或配置不安全

固定值: [Config your server to use the "Content-Security-Policy" header](#)

差异:

推理: AppScan detected that the Content-Security-Policy response header is missing, which increases exposure to various cross-site injection attacks

测试请求和响应:

```

GET /acidtab.js HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/mfs.cgi
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

```

```

HTTP/1.1 200 Ok
Content-Type: application/x-javascript
Content-Length: 13846

```

```

/*
 * Copyright (C) 2016 Jakub Kruszona-Zawadzki, Core Technology Sp. z o.o.
 *
 * This file is part of MooseFS.
 *
 * MooseFS is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, version 2 (only).
 *
 * MooseFS is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with MooseFS; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02111-1301, USA
 * or visit http://www.gnu.org/licenses/gpl-2.0.html
 */

```

```

if (typeof acid_ready == "undefined") {
    acid_ready = {
        ready : 0,
        functions : new Array(),
        docready : function() {
            if (acid_ready.ready==0) {
                acid_ready.ready = 1;
                var i;
                for (i=0 ; i<acid_ready.functions.length ; i++) {
                    acid_ready.functions[i]();
                }
            }
        },
        domloaded : function(e) {
            acid_ready.docready();
        },
        loaded : function(e) {
            acid_ready.docready();
        },
        readystate : function(e) {
            if (document.readyState=="complete" ||
document.readyState=="interactive") {
                acid_ready.docready();
            }
        },
        readytest : function() {
            if (acid_ready.ready==0) {
                if (typeof document.readyState!="undefined") {
                    if (document.readyState=="complete" ||
document.readyState=="interactive") {
                        acid_ready.docready();
                    } else {
                        setTimeout("acid_ready.readytest()",100);
                    }
                }
                return;
            }
            if (typeof document.documentElement!="undefined" && typeof
document.documentElement.doScroll!="undefined") {
                try {
                    document.documentElement.doScroll('left');
                } catch (e) {
                    setTimeout("acid_ready.readytest()",100);
                    return;
                }
            }
            acid_ready.docready();
        }
    },
    register : function(fn) {
        if (acid_ready.ready) {
            fn();
        } else {
            var l = acid_ready.functions.length;
            acid_ready.functions[l]=fn;
        }
    },
    init : function() {
        if (window.addEventListener) {
            document.addEventListener("DOMContentLoaded",acid_ready.domloaded,false);
            document.addEventListener("readystatechange",acid_ready.readystate,false);
            window.addEventListener("load",acid_ready.loaded,false);
        } else if (window.attachEvent) {
            document.attachEvent("onDOMContentLoaded",acid_ready.domloaded);
            document.attachEvent("onreadystatechange",acid_ready.readystate);
            window.attachEvent("onload",acid_ready.loaded);
        }
        setTimeout("acid_ready.readytest()",100);
    }
}
acid_ready.init();
}

if (typeof acid_tab == "undefined") {
    acid_tab = {
        init: function() {


```

```

var tabs,i;
if (!document.createElement || !document.getElementsByTagName) {
    return;
}
tabs = document.getElementsByTagName('table');
if (tabs) {
    for (i=0 ; i<tabs.length ; i++) {
        if (tabs[i].className.search(/\bacid_tab\b/) != -1) {
            acid_tab.preparetab(tabs[i]);
        }
    }
}

preparetab: function(table) {
    var i,j,k,x,h,m,s,p,c,z,r;
    // find thead using 'TH' node names
    if (table.getElementsByTagName('thead').length == 0) {
        var thead = document.createElement('thead');
        while (table.rows.length>0 &&
table.rows[0].cells[0].nodeName=="TH") {
            thead.appendChild(table.rows[0]);
        }
        table.insertBefore(thead,table.firstChild);
    }
    // create tHead if necessary
    if (typeof table.tHead == "undefined") {
        table.tHead = table.getElementsByTagName('thead')[0];
    }
    // backup and remove extra bodies
    table.acid_tab_tbodyesbackup = new Array();
    for (i=0 ; i<table.tBodies.length ; i++) {
        table.acid_tab_tbodyesbackup[i] = table.tBodies[i];
    }
    while (table.tBodies.length>1) {
        table.removeChild(table.tBodies[1]);
    }
    // no body? - exit
    if (table.tBodies.length==0) {
        return;
    }
    // check settings
    m = table.className.match(/\bacid_tab_zebra_([a-zA-Z0-9]+)_([a-zA-Z0-9]+)\b/);
    if (m) {
        table.acid_tab_zebra = new Array();
        table.acid_tab_zebra[0] = m[1];
        table.acid_tab_zebra[1] = m[2];
    }
    if (table.className.search(/\bacid_tab_noindicator\b/) != -1) {
        table.acid_tab_indicator = 0;
    } else {
        table.acid_tab_indicator = 1;
    }
    m = table.className.match(/\bacid_tab_storageid_([a-zA-Z0-9]+)\b/);
    if (m) {
        table.acid_tab_storageid = m[1];
    } else {
        table.acid_tab_storageid = ""
    }
    ...
    ...
    ...

```

Missing "X-Content-Type-Options" header**严重性:** **CVSS 分数:** 5.0**URL:** http://10.62.233.181:9425/index.html**实体:** index.html (Page)**风险:** 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息, 如用户名、密码、机器名和/或敏感文件位置**原因:** Web 应用程序编程或配置不安全**固定值:** [Config your server to use the "X-Content-Type-Options" header](#)**差异:****推理:** AppScan detected that the X-Content-Type-Options response header is missing, which increases exposure to drive-by download attacks**测试请求和响应:**

```
GET /index.html HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html
Content-Length: 537

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>MFS</title>
    <script type="text/javascript">
      document.location.href="mfs.cgi"
      // the above uses default parameter values; full invocation example is shown below
      //      document.location.href="mfs.cgi?
masterhost=mfsmaster&masterport=9421&mastercontrolp...
    </script>
  </head>
  <body>
  </body>
</html>
```

Missing "X-Content-Type-Options" header

严重性: 低

CVSS 分数: 5.0

URL: http://10.62.233.181:9425/mfs.cgi

实体: mfs.cgi (Page)

风险: 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息, 如用户名、密码、机器名和/或敏感文件位置

原因: Web 应用程序编程或配置不安全

固定值: [Config your server to use the "X-Content-Type-Options" header](#)

差异:

推理: AppScan detected that the X-Content-Type-Options response header is missing, which increases exposure to drive-by download attacks

测试请求和响应:

```
GET /mfs.cgi?sections=CS HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/mfs.cgi
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>MFS Info (MooseFS)</title>
<link rel="stylesheet" href="mfs.css" type="text/css" />
<script src="acidtab.js" type="text/javascript"></script>
</head>
<body>
<div id="header">
<table class="HDR" cellpadding="0" cellspacing="0" border="0">
<tr>
<td class="LOGO"><a href="http://www.moosefs.org"></a></td>
<td class="MENU"><table class="MENU" cellspacing="0">
<tr>
<td class="UU"><a href="mfs.cgi?sections=IN">Info</a> <a href="mfs.cgi?sections=CS%7CIN"></a>
</td>
<td class="US">Servers &#8722;</td>
<td class="SU"><a href="mfs.cgi?sections=HD">Disks</a> <a href="mfs.cgi?sections=CS%7CHD"></a>
</td>
<td class="UU"><a href="mfs.cgi?sections=EX">Exports</a> <a href="mfs.cgi?sections=CS%7CEX"></a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MS">Mounts</a> <a href="mfs.cgi?sections=CS%7CMS"></a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MO">Operations</a> <a href="mfs.cgi?sections=CS%7CMO"></a></td>
<td class="UU"><a href="mfs.cgi?sections=QU">Quotas</a> <a href="mfs.cgi?sections=CS%7CQU"></a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MC">Master Charts</a> <a href="mfs.cgi?sections=CS%7CMC"></a></td>
<td class="LUU"><a href="mfs.cgi?sections=CC">Server Charts</a> <a href="mfs.cgi?sections=CS%7CCC"></a></td>
</tr>
</table></td>
<td class="FILLER" style="white-space:nowrap;">
```



```

</td>
</tr>
</table>
</div>
<div id="container">
<table class="acid_tab acid_tab_zebra_C1_C2 acid_tab_storageid_mfscs" cellspacing="0">
<tr><th colspan="17">Chunk Servers</th></tr>
<tr>
<th rowspan="2" class="acid_tab_enumerate">#</th>
<th rowspan="2">host</th>
<th rowspan="2">ip</th>
<th rowspan="2">port</th>
<th rowspan="2">id</th>
<th rowspan="2">version</th>
<th rowspan="2">load</th>
<th rowspan="2">maintenance</th>
<th colspan="4">'regular' hdd space</th>
<th colspan="4">'marked for removal' hdd space</th>

</tr>
<tr>
<th>chunks</th>
<th>used</th>
<th>total</th>
<th class="PROGBAR">% used</th>
<th>chunks</th>
<th>used</th>
<th>total</th>
<th class="PROGBAR">% used</th>

</tr>
<tr>
<td align="right"></td>
<td align="left">(unresolved)</td>
<td align="center"><span class="sortkey">010_093_211_036 </span>10.93.211.36</td>
<td align="center">9422</td>
<td align="center">1</td>
<td align="center"><span class="sortkey">00002_000_088_1 </span><span
class="OKVERSION">2.0.88</span></td>
<td align="right">0</td>
<td align="center"><a href="mfs.cgi?
sections=CS&CSmaintenanceon=10.93.211.36%3A9422">switch on</a></td>
<td align="right">0</td><td align="right"><span class="sortkey">449912832 </span>
<a style="cursor:default" title="449 912 832 B">429&nbsp;MiB</a></td><td align="right"><span
class="sortkey">30101889024 </span><a style="cursor:default" title="30 101 889 024
B">28&nbsp;GiB</a></td>
<td align="center"><span class="sortkey">1.4946332160 </span><div class="PROGBOX"
style="width:200px;"><div class="PROGCOVER" style="width:98.51%;"></div><div class="PROGAVG"
style="width:1.49%"></div><div class="PROGVALUE"><a style="cursor:default" title="1.4946% =
(avg+0.0000%)">1.49</a></div></div></td>
<td align="right">0</td><td align="right"><span class="sortkey">0 </span><a
style="cursor:default" title="0 B">0&nbsp;B</a></td><td align="right"><span class="sortkey">0
</span><a style="cursor:default" title="0 B">0&nbsp;B</a></td>
<td align="center"><span class="sortkey">-1 </span><div class="PROGBOX"
style="width:200px;"><div class="PROGCOVER" style="width:100%;"></div><div class="PROGVALUE">-
</div></div></td>
</tr>
</table>
<br/>
<table class="acid_tab acid_tab_zebra_C1_C2 acid_tab_storageid_mfsmbl" cellspacing="0">
<tr><th colspan="4">Metadata Backup Loggers</th></tr>
<tr>
<th class="acid_tab_enumerate">#</th>
<th>host</th>
<th>ip</th>
<th>version</th>

</tr>
<tr>
<td align="right"></td><td align="left">(unresolved)</td><td align="center"><span
class="sortkey">010_093_216_120 </span>10.93.216.120</td><td align="center"><span
class="sortkey">00002_000_088_1 </span><span class="OKVERSION">2.0.88</span></td>

</tr>
</table>
</div> <!-- end of container -->
</body>
</html>

```

Missing "X-Content-Type-Options" header**严重性:** 低**CVSS 分数:** 5.0**URL:** http://10.62.233.181:9425/acidtab.js**实体:** acidtab.js (Page)**风险:** 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息，如用户名、密码、机器名和/或敏感文件位置**原因:** Web 应用程序编程或配置不安全**固定值:** [Config your server to use the "X-Content-Type-Options" header](#)**差异:****推理:** AppScan detected that the X-Content-Type-Options response header is missing, which increases exposure to drive-by download attacks**测试请求和响应:**

```
GET /acidtab.js HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/mfs.cgi
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: application/x-javascript
Content-Length: 13846

/*
 * Copyright (C) 2016 Jakub Kruszona-Zawadzki, Core Technology Sp. z o.o.
 *
 * This file is part of MooseFS.
 *
 * MooseFS is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, version 2 (only).
 *
 * MooseFS is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with MooseFS; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02111-1301, USA
 * or visit http://www.gnu.org/licenses/gpl-2.0.html
 */

if (typeof acid_ready == "undefined") {
  acid_ready = {
    ready : 0,
    functions : new Array(),
    docready : function() {
      if (acid_ready.ready==0) {
        acid_ready.ready = 1;
        var i;
        for (i=0 ; i<acid_ready.functions.length ; i++) {
          acid_ready.functions[i]();
        }
      }
    }
  };
}
```

```

    }
    },
    domloaded : function(e) {
        acid_ready.docready();
    },
    loaded : function(e) {
        acid_ready.docready();
    },
    readystate : function(e) {
        if (document.readyState=="complete" ||
document.readyState=="interactive") {
            acid_ready.docready();
        }
    },
    readytest : function() {
        if (acid_ready.ready==0) {
            if (typeof document.readyState!="undefined") {
                if (document.readyState=="complete" ||
document.readyState=="interactive") {
                    acid_ready.docready();
                } else {
                    setTimeout("acid_ready.readytest()",100);
                }
                return;
            }
            if (typeof document.documentElement!="undefined" && typeof
document.documentElement.doScroll!="undefined") {
                try {
                    document.documentElement.doScroll('left');
                } catch (e) {
                    setTimeout("acid_ready.readytest()",100);
                    return;
                }
            }
            acid_ready.docready();
        }
    },
    register : function(fn) {
        if (acid_ready.ready) {
            fn();
        } else {
            var l = acid_ready.functions.length;
            acid_ready.functions[l]=fn;
        }
    },
    init : function() {
        if (window.addEventListener) {
document.addEventListener("DOMContentLoaded",acid_ready.domloaded,false);
document.addEventListener("readystatechange",acid_ready.readystate,false);
            window.addEventListener("load",acid_ready.loaded,false);
        } else if (window.attachEvent) {
            document.attachEvent("onDOMContentLoaded",acid_ready.domloaded);
            document.attachEvent("onreadystatechange",acid_ready.readystate);
            window.attachEvent("onload",acid_ready.loaded);
        }
        setTimeout("acid_ready.readytest()",100);
    }
}
acid_ready.init();
}

if (typeof acid_tab == "undefined") {
    acid_tab = {
        init: function() {
            var tabs,i;
            if (!document.createElement || !document.getElementsByTagName) {
                return;
            }
            tabs = document.getElementsByTagName('table');
            if (tabs) {
                for (i=0 ; i<tabs.length ; i++) {
                    if (tabs[i].className.search(/\\bacid_tab\\b/) != -1) {
                        acid_tab.preparetab(tabs[i]);
                    }
                }
            }
        }
    }
}

```

```

    }
    ,

    preparetab: function(table) {
        var i,j,k,x,h,m,s,p,c,z,r;
        // find thead using 'TH' node names
        if (table.getElementsByTagName('thead').length == 0) {
            var thead = document.createElement('thead');
            while (table.rows.length>0 &&
table.rows[0].cells[0].nodeName=="TH") {
                thead.appendChild(table.rows[0]);
            }
            table.insertBefore(thead,table.firstChild);
        }
        // create tHead if necessary
        if (typeof table.tHead == "undefined") {
            table.tHead = table.getElementsByTagName('thead')[0];
        }
        // backup and remove extra bodies
        table.acid_tab_tbodyesbackup = new Array();
        for (i=0 ; i<table.tBodies.length ; i++) {
            table.acid_tab_tbodyesbackup[i] = table.tBodies[i];
        }
        while (table.tBodies.length>1) {
            table.removeChild(table.tBodies[1]);
        }
        // no body? - exit
        if (table.tBodies.length==0) {
            return;
        }
        // check settings
        m = table.className.match(/\bacid_tab_zebra_([a-zA-Z0-9]+)_([a-zA-Z0-9-9]+)\b/);
        if (m) {
            table.acid_tab_zebra = new Array();
            table.acid_tab_zebra[0] = m[1];
            table.acid_tab_zebra[1] = m[2];
        }
        if (table.className.search(/\bacid_tab_noindicator\b/)!==-1) {
            table.acid_tab_indicator = 0;
        } else {
            table.acid_tab_indicator = 1;
        }
        m = table.className.match(/\bacid_tab_storageid_([a-zA-Z0-9-9]+)\b/);
        if (m) {
            table.acid_tab_storageid = m[1];
        } else {
            table.acid_tab_storageid = ""
        }
        ...
        ...
        ...
    }

```

低

Missing "X-XSS-Protection" header 3

TOC

问题 1 / 3

TOC

Missing "X-XSS-Protection" header

严重性: 低

CVSS 分数: 5.0

URL: http://10.62.233.181:9425/index.html

实体: index.html (Page)

风险: 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息，如用户名、密码、机器名和/或敏感文件位置

原因: Web 应用程序编程或配置不安全

固定值: [Config your server to use the "X-XSS-Protection" header](#)

差异:

推理: AppScan detected that the X-XSS-Protection response header is missing, which may allow Cross-Site Scripting attacks

测试请求和响应:

```
GET /index.html HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html
Content-Length: 537

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>MFS</title>
    <script type="text/javascript">
      document.location.href="mfs.cgi"
      // the above uses default parameter values; full invocation example is shown below
      //          document.location.href="mfs.cgi?
masterhost=mfsmaster&masterport=9421&mastercontrolp...
    </script>
  </head>
  <body>
  </body>
</html>
```

Missing "X-XSS-Protection" header

严重性: 低

CVSS 分数: 5.0

URL: http://10.62.233.181:9425/mfs.cgi

实体: mfs.cgi (Page)

风险: 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息, 如用户名、密码、机器名和/或敏感文件位置

原因: Web 应用程序编程或配置不安全

固定值: [Config your server to use the "X-XSS-Protection" header](#)

差异:

推理: AppScan detected that the X-XSS-Protection response header is missing, which may allow Cross-Site Scripting attacks

测试请求和响应:

```
GET /mfs.cgi?sections=CS HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/mfs.cgi
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>MFS Info (MooseFS)</title>
<link rel="stylesheet" href="mfs.css" type="text/css" />
<script src="acidtab.js" type="text/javascript"></script>
</head>
<body>
<div id="header">
<table class="HDR" cellpadding="0" cellspacing="0" border="0">
<tr>
<td class="LOGO"><a href="http://www.moosefs.org"></a></td>
<td class="MENU"><table class="MENU" cellspacing="0">
<tr>
<td class="UU"><a href="mfs.cgi?sections=IN">Info</a> <a href="mfs.cgi?sections=CS%7CIN"></a>
</td>
<td class="US">Servers &#8722;</td>
<td class="SU"><a href="mfs.cgi?sections=HD">Disks</a> <a href="mfs.cgi?sections=CS%7CHD"></a>
</td>
<td class="UU"><a href="mfs.cgi?sections=EX">Exports</a> <a href="mfs.cgi?sections=CS%7CEX"></a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MS">Mounts</a> <a href="mfs.cgi?sections=CS%7CMS"></a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MO">Operations</a> <a href="mfs.cgi?sections=CS%7CMO"></a></td>
<td class="UU"><a href="mfs.cgi?sections=QU">Quotas</a> <a href="mfs.cgi?sections=CS%7CQU"></a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MC">Master Charts</a> <a href="mfs.cgi?sections=CS%7CMC"></a></td>
<td class="LUU"><a href="mfs.cgi?sections=CC">Server Charts</a> <a href="mfs.cgi?sections=CS%7CCC"></a></td>
</tr>
</table></td>
<td class="FILLER" style="white-space:nowrap;">
```

```

</td>
</tr>
</table>
</div>
<div id="container">
<table class="acid_tab acid_tab_zebra_C1_C2 acid_tab_storageid_mfscs" cellspacing="0">
  <tr><th colspan="17">Chunk Servers</th></tr>
  <tr>
    <th rowspan="2" class="acid_tab_enumerate">#</th>
    <th rowspan="2">host</th>
    <th rowspan="2">ip</th>
    <th rowspan="2">port</th>
    <th rowspan="2">id</th>
    <th rowspan="2">version</th>
    <th rowspan="2">load</th>
    <th rowspan="2">maintenance</th>
    <th colspan="4">'regular' hdd space</th>
    <th colspan="4">'marked for removal' hdd space</th>
  </tr>
  <tr>
    <th>chunks</th>
    <th>used</th>
    <th>total</th>
    <th class="PROGBAR">% used</th>
    <th>chunks</th>
    <th>used</th>
    <th>total</th>
    <th class="PROGBAR">% used</th>
  </tr>
  <tr>
    <td align="right"></td>
    <td align="left">(unresolved)</td>
    <td align="center"><span class="sortkey">010_093_211_036 </span>10.93.211.36</td>
    <td align="center">9422</td>
    <td align="center">1</td>
    <td align="center"><span class="sortkey">00002_000_088_1 </span><span
class="OKVERSION">2.0.88</span></td>
    <td align="right">0</td>
    <td align="center"><a href="mfs.cgi?
sections=CS&CSmaintenanceon=10.93.211.36%3A9422">switch on</a></td>
    <td align="right">0</td><td align="right"><span class="sortkey">449912832 </span>
<a style="cursor:default" title="449 912 832 B">429&nbsp;MiB</a></td><td align="right"><span
class="sortkey">30101889024 </span><a style="cursor:default" title="30 101 889 024
B">28&nbsp;GiB</a></td>
    <td align="center"><span class="sortkey">1.4946332160 </span><div class="PROGBOX"
style="width:200px;"><div class="PROGCOVER" style="width:98.51%;"></div><div class="PROGAVG"
style="width:1.49%"></div><div class="PROGVALUE"><a style="cursor:default" title="1.4946% =
(avg+0.0000%)">1.49</a></div></div></td>
    <td align="right">0</td><td align="right"><span class="sortkey">0 </span><a
style="cursor:default" title="0 B">0&nbsp;B</a></td><td align="right"><span class="sortkey">0
</span><a style="cursor:default" title="0 B">0&nbsp;B</a></td>
    <td align="center"><span class="sortkey">-1 </span><div class="PROGBOX"
style="width:200px;"><div class="PROGCOVER" style="width:100%;"></div><div class="PROGVALUE">-
</div></div></td>
  </tr>
</table>
<br/>
<table class="acid_tab acid_tab_zebra_C1_C2 acid_tab_storageid_mfsmbl" cellspacing="0">
  <tr><th colspan="4">Metadata Backup Loggers</th></tr>
  <tr>
    <th class="acid_tab_enumerate">#</th>
    <th>host</th>
    <th>ip</th>
    <th>version</th>
  </tr>
  <tr>
    <td align="right"></td><td align="left">(unresolved)</td><td align="center"><span
class="sortkey">010_093_216_120 </span>10.93.216.120</td><td align="center"><span
class="sortkey">00002_000_088_1 </span><span class="OKVERSION">2.0.88</span></td>
  </tr>
</table>
</div> <!-- end of container -->
</body>
</html>

```

Missing "X-XSS-Protection" header**严重性:** 低**CVSS 分数:** 5.0**URL:** http://10.62.233.181:9425/acidtab.js**实体:** acidtab.js (Page)**风险:** 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
可能会收集有关 Web 应用程序的敏感信息, 如用户名、密码、机器名和/或敏感文件位置**原因:** Web 应用程序编程或配置不安全**固定值:** [Config your server to use the "X-XSS-Protection" header](#)**差异:****推理:** AppScan detected that the X-XSS-Protection response header is missing, which may allow Cross-Site Scripting attacks**测试请求和响应:**

```

GET /acidtab.js HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/mfs.cgi
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: application/x-javascript
Content-Length: 13846

/*
 * Copyright (C) 2016 Jakub Kruszona-Zawadzki, Core Technology Sp. z o.o.
 *
 * This file is part of MooseFS.
 *
 * MooseFS is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, version 2 (only).
 *
 * MooseFS is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with MooseFS; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02111-1301, USA
 * or visit http://www.gnu.org/licenses/gpl-2.0.html
 */

if (typeof acid_ready == "undefined") {
  acid_ready = {
    ready : 0,
    functions : new Array(),
    docready : function() {
      if (acid_ready.ready==0) {
        acid_ready.ready = 1;
        var i;
        for (i=0 ; i<acid_ready.functions.length ; i++) {
          acid_ready.functions[i]();

```



```

    }
    },
    domloaded : function(e) {
        acid_ready.docready();
    },
    loaded : function(e) {
        acid_ready.docready();
    },
    readystate : function(e) {
        if (document.readyState=="complete" ||
document.readyState=="interactive") {
            acid_ready.docready();
        }
    },
    readytest : function() {
        if (acid_ready.ready==0) {
            if (typeof document.readyState!="undefined") {
                if (document.readyState=="complete" ||
document.readyState=="interactive") {
                    acid_ready.docready();
                } else {
                    setTimeout("acid_ready.readytest()",100);
                }
                return;
            }
            if (typeof document.documentElement!="undefined" && typeof
document.documentElement.doScroll!="undefined") {
                try {
                    document.documentElement.doScroll('left');
                } catch (e) {
                    setTimeout("acid_ready.readytest()",100);
                    return;
                }
            }
            acid_ready.docready();
        }
    },
    register : function(fn) {
        if (acid_ready.ready) {
            fn();
        } else {
            var l = acid_ready.functions.length;
            acid_ready.functions[l]=fn;
        }
    },
    init : function() {
        if (window.addEventListener) {
document.addEventListener("DOMContentLoaded",acid_ready.domloaded,false);
document.addEventListener("readystatechange",acid_ready.readystate,false);
            window.addEventListener("load",acid_ready.loaded,false);
        } else if (window.attachEvent) {
            document.attachEvent("onDOMContentLoaded",acid_ready.domloaded);
            document.attachEvent("onreadystatechange",acid_ready.readystate);
            window.attachEvent("onload",acid_ready.loaded);
        }
        setTimeout("acid_ready.readytest()",100);
    }
}
acid_ready.init();
}

if (typeof acid_tab == "undefined") {
    acid_tab = {
        init: function() {
            var tabs,i;
            if (!document.createElement || !document.getElementsByTagName) {
                return;
            }
            tabs = document.getElementsByTagName('table');
            if (tabs) {
                for (i=0 ; i<tabs.length ; i++) {
                    if (tabs[i].className.search(/\\bacid_tab\\b/) != -1) {
                        acid_tab.preparetab(tabs[i]);
                    }
                }
            }
        }
    }
}

```

```

    }
    ,

    preparetab: function(table) {
        var i,j,k,x,h,m,s,p,c,z,r;
        // find thead using 'TH' node names
        if (table.getElementsByTagName('thead').length == 0) {
            var thead = document.createElement('thead');
            while (table.rows.length>0 &&
table.rows[0].cells[0].nodeName=="TH") {
                thead.appendChild(table.rows[0]);
            }
            table.insertBefore(thead,table.firstChild);
        }
        // create tHead if necessary
        if (typeof table.tHead == "undefined") {
            table.tHead = table.getElementsByTagName('thead')[0];
        }
        // backup and remove extra bodies
        table.acid_tab_tbodybackup = new Array();
        for (i=0 ; i<table.tBodies.length ; i++) {
            table.acid_tab_tbodybackup[i] = table.tBodies[i];
        }
        while (table.tBodies.length>1) {
            table.removeChild(table.tBodies[1]);
        }
        // no body? - exit
        if (table.tBodies.length==0) {
            return;
        }
        // check settings
        m = table.className.match(/\\bacid_tab_zebra_([a-zA-Z0-9]+)_([a-zA-Z0-9-9]+)\\b/);
        if (m) {
            table.acid_tab_zebra = new Array();
            table.acid_tab_zebra[0] = m[1];
            table.acid_tab_zebra[1] = m[2];
        }
        if (table.className.search(/\\bacid_tab_noindicator\\b/)!==-1) {
            table.acid_tab_indicator = 0;
        } else {
            table.acid_tab_indicator = 1;
        }
        m = table.className.match(/\\bacid_tab_storageid_([a-zA-Z0-9]+)\\b/);
        if (m) {
            table.acid_tab_storageid = m[1];
        } else {
            table.acid_tab_storageid = ""
        }
        ...
        ...
        ...
    }

```

问题 1 / 1

TOC

发现内部 IP 泄露模式

严重性:

[参考信息](#)

CVSS 分数: 0.0

URL: http://10.62.233.181:9425/mfs.cgi

实体: mfs.cgi (Page)

风险: 可能会收集有关 Web 应用程序的敏感信息, 如用户名、密码、机器名和/或敏感文件位置

原因: Web 应用程序编程或配置不安全

固定值: 除去 Web 站点中的内部 IP 地址

差异:

推理: AppScan 在响应中发现了看似为内部 IP 地址的内容。

测试请求和响应:

```
GET /mfs.cgi HTTP/1.1
Accept-Language: en-US
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://10.62.233.181:9425/index.html
Host: 10.62.233.181:9425
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; Tablet PC 2.0)

HTTP/1.1 200 Ok
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>MFS Info (MooseFS)</title>
<link rel="stylesheet" href="mfs.css" type="text/css" />
<script src="acidtab.js" type="text/javascript"></script>
</head>
<body>
<div id="header">
<table class="HDR" cellpadding="0" cellspacing="0" border="0">
<tr>
<td class="LOGO"><a href="http://www.moosefs.org"></a></td>
<td class="MENU"><table class="MENU" cellspacing="0">
<tr>
<td class="US">Info &#8722;</td>
<td class="SU"><a href="mfs.cgi?sections=CS">Servers</a> <a href="mfs.cgi?sections=CS%7CIN"></a>
</td>
</tr>
</table>
</td>
</tr>
</table>
</div>
```

```

<td class="UU"><a href="mfs.cgi?sections=HD">Disks</a> <a href="mfs.cgi?sections=HD%7CIN">+</a>
</td>
<td class="UU"><a href="mfs.cgi?sections=EX">Exports</a> <a href="mfs.cgi?sections=EX%7CIN">+</a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MS">Mounts</a> <a href="mfs.cgi?sections=MS%7CIN">+</a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MO">Operations</a> <a href="mfs.cgi?sections=MO%7CIN">+
</a></td>
<td class="UU"><a href="mfs.cgi?sections=QU">Quotas</a> <a href="mfs.cgi?sections=QU%7CIN">+</a>
</td>
<td class="UU"><a href="mfs.cgi?sections=MC">Master Charts</a> <a href="mfs.cgi?
sections=MC%7CIN">+</a></td>
<td class="LUU"><a href="mfs.cgi?sections=CC">Server Charts</a> <a href="mfs.cgi?
sections=CC%7CIN">+</a></td>
</tr>
</table></td>
<td class="FILLER" style="white-space:nowrap;">
</td>
</tr>
</table>
</div>
<div id="container">
<table class="FR" cellspacing="0">
<tr><th>Notice</th></tr>
<tr><td>You are currently using GPL version of MooseFS. If you want to find out what great
features are available in MooseFS Pro go to <a
href="http://moosefs.com/products.html">http://moosefs.com/products.html</a></td></tr>
</table>
<br/>
<table class="acid_tab acid_tab_zebra_C1_C2 acid_tab_storageid_mfsmasters" cellspacing="0">
<tr><th colspan="10">Metadata Servers (masters)</th></tr>
<tr>
<th class="acid_tab_enumerate">#</th>
<th>ip</th>
<th>version</th>
<th>state</th>
<th>metadata version</th>
<th>RAM used</th>
<th>CPU used</th>
<th>last successful metadata save</th>
<th>last metadata save duration</th>
<th>last metadata save status</th>
</tr>
<tr>
<td align="right"></td><td align="center"><span class="sortkey">010_062_233_181
</span>10.62.233.181</td><td align="center"><span class="sortkey">00002_000_088_1 </span><span
class="OKVERSION">2.0.88</span></td>
<td align="center"><span class="STATECOLOR0">-</span></td>
<td align="right">4</td>
<td align="center"><a style="cursor:default" title="213 868 544
B">204&nbsp;MiB</a></td>
<td align="center"><a style="cursor:default" title="all:0.0800092% sys:0.0400046%
user:0.0400046%">all:0.08%&nbsp;sys:0.04%&nbsp;user:0.04%</a></td>
<td align="center">-</td><td align="center">-</td>
<td align="center">-</td>
</tr>
</table>
<br/>
<table class="FR" cellspacing="0">
<tr><th colspan="12">Metadata Info</th></tr>
<tr>
<th>total space</th>
<th>avail space</th>
<th>trash space</th>
<th>trash files</th>
<th>sustained space</th>
<th>sustained files</th>
<th>all fs objects</th>
<th>directories</th>
<th>files</th>
<th>chunks</th>
<th><a style="cursor:default" title="chunks from 'regular' hdd space and 'marked
for removal' hdd space">all chunk copies</a></th>
<th><a style="cursor:default" title="only chunks from 'regular' hdd
space">regular chunk copies</a></th>
</tr>
<tr>
<td align="center"><a style="cursor:default" title="30 101 889 024

```

```

B">28&nbsp;GiB</a></td>
  <td align="center"><a style="cursor:default" title="29 651 976 192
B">28&nbsp;GiB</a></td>
  <td align="center"><a style="cursor:default" title="0 B">0&nbsp;B</a></td>
  <td align="center">0</td>
  <td align="center"><a style="cursor:default" title="0 B">0&nbsp;B</a></td>
  <td align="center">0</td>
  <td align="center">1</td>
  <td align="center">1</td>
  <td align="center">0</td>
  <td align="center">0</td>
  <td align="center">0</td>
  <td align="center">0</td>
</tr>
</table>
<br/>
<table class="FR" cellpadding="0">
  <tr><th colspan="13">Memory usage detailed info</th></tr>
  <tr><th></th>
    <th>chunk hash</th>
    <th>chunks</th>
    <th>cs lists</th>
    <th>edge hash</th>
    <th>edges</th>
    <th>node hash</th>
    <th>nodes</th>
    <th>deleted nodes</th>
    <th>chunk
  ...
  ...
  ...

```

修订建议

高

查看危险字符注入的可能解决方案

TOC

该任务修复的问题类型

- 跨站点脚本编制
- 链接注入（便于跨站请求伪造）
- 通过框架钓鱼

一般

跨站点脚本编制

有多种减轻威胁的技巧：

[1] 策略：库或框架

使用不允许此弱点出现的经过审核的库或框架，或提供更容易避免此弱点的构造。

可用于更轻松生成正确编码的输出的库和框架示例包括 Microsoft 的 Anti-XSS 库、OWASP ESAPI 编码模块和 Apache Wicket。

[2] 了解将在其中使用数据的上下文，以及预期的编码。在不同组件之间传输数据时，或在生成可同时包含多个编码的输出（如 Web 页面或多部分邮件消息）时，这尤为重要。研究所有预期的通信协议和数据表示法以确定所需的编码策略。对于将输出到另一个 Web 页面的任何数据（尤其是从外部输入接收到的任何数据），请对所有非字母数字字符使用恰当的编码。

相同输出文档的某些部分可能需要不同的编码，具体取决于输出是在以下哪一项中：

[-] HTML 主体

[-] 元素属性（如 `src="XYZ"`）

[-] URI

[-] JavaScript 段

[-] 级联样式表和样式属性

请注意，“HTML 实体编码”仅适用于 HTML 主体。

请咨询 XSS Prevention Cheat Sheet

[http://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](http://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)

以获取有关所需编码和转义类型的更多详细信息。

[3] 策略：识别和减少攻击出现的机会

了解您的软件中可能出现不可信输入的所有潜在区域：参数或自变量、cookie、从网络读取的任何内容、环境变量、反向 DNS 查找、查询结果、请求头、URL 组成部分、电子邮件、文件、文件名、数据库以及向应用程序提供数据的任何外部系统。请记住，此类输入可通过 API 调用间接获取。

[4] 策略：输出编码

对于生成的每个 Web 页面，请使用并指定 ISO-8859-1 或 UTF-8 之类的字符编码。如果未指定编码，Web 浏览器可能通过猜测 Web 页面实际使用的编码来选择不同的编码。这可能导致 Web 浏览器将特定序列视为特殊序列，从而使客户机暴露在不易察觉的 XSS 攻击之下。请参阅 CWE-116 以获取与编码/转义相关的更多减轻威胁的方法。

[5] 策略：识别和减少攻击出现的机会

要帮助减轻针对用户会话 cookie 的 XSS 攻击带来的威胁，请将会话 cookie 设置为 HttpOnly。在支持 HttpOnly 功能的浏览器（如 Internet Explorer 和 Firefox 的较新版本）中，此属性可防止使用 document.cookie 的恶意客户端脚本访问用户的会话 cookie。这不是完整的解决方案，因为 HttpOnly 并不受所有浏览器支持。更重要的是，XMLHttpRequest 和其他功能强大的浏览器技术提供了对 HTTP 头的读访问权，包括在其中设置 HttpOnly 标志的 Set-Cookie 头。

[6] 策略：输入验证假定所有输入都是恶意的。使用“接受已知善意”输入验证策略：严格遵守规范的可接受输入的白名单。拒绝任何没有严格遵守规范的输入，或者将其转换为遵守规范的内容。不要完全依赖于将恶意或格式错误的输入加入黑名单。但是，黑名单可帮助检测潜在攻击，或者确定哪些输入格式不正确，以致应当将其彻底拒绝。

执行输入验证时，请考虑所有潜在相关属性，包括长度、输入类型、可接受值的完整范围、缺失或多余输入、语法、跨相关字段的一致性以及业务规则一致性。以业务规则逻辑为例，“boat”可能在语法上有效，因为它仅包含字母数字字符，但如果预期为颜色（如“red”或“blue”），那么它无效。

动态构造 Web 页面时，请使用严格的白名单以根据请求中参数的预期值来限制字符集。所有输入都应进行验证和清理，不仅限于用户应指定的参数，而是涉及请求中的所有数据，包括隐藏字段、cookie、头、URL 本身，等等。导致 XSS 脆弱性持续存在的一个常见错误是仅验证预期会由站点重新显示的字段。常见的情况是，在请求中出现由应用程序服务器或应用程序反射的其他数据，而开发团队却未能预料到此情况。另外，将来的开发者可能会使用当前未反映的字段。因此，建议验证 HTTP 请求的所有部分。请注意，适当的输出编码、转义和引用是防止 XSS 的最有效解决方案，虽然输入验证可能会提供一定的深度防御。输入验证会有效限制将在输出中出现的内容。它并不总是能够防止 XSS，尤其是在您需要支持可包含任意字符的自由格式文本字段的情况下。例如，在聊天应用程序中，心型表情图标（“<3”）可能会通过验证步骤，因为它的使用频率很高。但是，不能将其直接插入到 Web 页面中，因为它包含“<”字符，该字符需要转义或以其他方式进行处理。在此情况下，消除“<”可能会降低 XSS 的风险，但是这会产​​生不正确的行为，因为这样就不会记录表情图标。

这可能看起来只是略有不便，但在需要表示不等式的数学论坛中，这种情况就更为重要。即使在验证中出错（例如，在 100 个输入字段中忘记一个字段），相应的编码仍有可能针对基于注入的攻击为您提供防护。只要输入验证不是孤立完成的，便仍是有用的技巧，因为它可以大大减少攻击出现的机会，使您能够检测某些攻击，并提供正确编码所无法解决的其他安全性优势。请确保在应用程序内定义良好的界面中执行输入验证。即使某个组件进行了复用或移动到其他位置，这也将有助于保护应用程序。

链接注入（便于跨站请求伪造）

有多种减轻威胁的技巧：

[1] 策略：库或框架

使用不允许此弱点出现的经过审核的库或框架，或提供更容易避免此弱点的构造。

可用于更轻松生成正确编码的输出的库和框架示例包括 Microsoft 的 Anti-XSS 库、OWASP ESAPI 编码模块和 Apache Wicket。

[2] 了解将在其中使用数据的上下文，以及预期的编码。在不同组件之间传输数据时，或在生成可同时包含多个编码的输出（如 Web 页面或多部分邮件消息）时，这尤为重要。研究所有预期的通信协议和数据表示法以确定所需的编码策略。对于将输出到另一个 Web 页面的任何数据（尤其是从外部输入接收到的任何数据），请对所有非字母数字字符使用恰当的编码。

相同输出文档的某些部分可能需要不同的编码，具体取决于输出是在以下哪一项中：

[-] HTML 主体

[-] 元素属性（如 src="XYZ"）

[-] URI

[-] JavaScript 段

[-] 级联样式表和样式属性

请注意，“HTML 实体编码”仅适用于 HTML 主体。

请咨询 XSS Prevention Cheat Sheet

[http://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](http://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)

以获取有关所需编码和转义类型的更多详细信息。

[3] 策略：识别和减少攻击出现的机会

了解您的软件中可能出现不可信输入的所有潜在区域：参数或自变量、cookie、从网络读取的任何内容、环境变量、反向 DNS 查找、查询结果、请求头、URL 组成部分、电子邮件、文件、文件名、数据库以及向应用程序提供数据的任何外部系统。请记住，此类输入可通过 API 调用间接获取。

[4] 策略：输出编码

对于生成的每个 Web 页面，请使用并指定 ISO-8859-1 或 UTF-8 之类的字符编码。如果未指定编码，Web 浏览器可能通过猜测 Web 页面编码来选择不同的编码。这可能导致 Web 浏览器将特定序列视为特殊序列，从而使客户机暴露在不易察觉的 XSS 攻击之下。请参阅 CWE-116 以获取与编码/转义相关的更多减轻威胁的方法。

[5] 策略：识别和减少攻击出现的机会

要帮助减轻针对用户会话 cookie 的 XSS 攻击带来的威胁，请将会话 cookie 设置为 HttpOnly。在支持 HttpOnly 功能的浏览器（如 Internet Explorer 和 Firefox 的较新版本）中，此属性可防止使用 document.cookie 的恶意客户端脚本访问用户的会话 cookie。这不是完整的解决方案，因为 HttpOnly 并不受所有浏览器支持。更重要的是，

XMLHttpRequest 和其他功能强大的浏览器技术提供了对 HTTP 头的读访问权，包括在其中设置 HttpOnly 标志的 Set-Cookie 头。

[6] 策略：输入验证假定所有输入都是恶意的。使用“接受已知善意”输入验证策略：严格遵守规范的可接受输入的白名单。拒绝没有严格遵守规范的输入，或者将其变换为严格遵守规范的内容。不要完全依赖于针对恶意或格式错误的输入的黑名单。但是，黑名单可帮助检测潜在攻击，或者确定哪些输入格式不正确，以致应当将其彻底拒绝。

执行输入验证时，请考虑所有潜在相关属性，包括长度、输入类型、可接受值的完整范围、缺失或多余输入、语法、跨相关字段的一致性以及业务规则一致性。以业务规则逻辑为例，“boat”可能在语法上有效，因为它仅包含字母数字字符，但如果预期为颜色（如“red”或“blue”），那么它无效。

动态构造 Web 页面时，请使用严格的白名单以根据请求中参数的预期值来限制字符集。所有输入都应进行验证和清理：不仅限于用户应指定的参数，而是涉及请求中的所有数据，包括隐藏字段、cookie、头、URL 本身，等等。导致 XSS 脆弱性持续存在的一个常见错误是仅验证预期会由站点重新显示的字段。常见的情况是，在请求中出现由应用程序服务器或应用程序反射的数据，而开发团队却未能预料到此情况。另外，将来的开发者可能会使用当前未反映的字段。因此，建议验证 HTTP 请求的所有部分。请注意，适当的输出编码、转义和引用是防止 XSS 的最有效解决方案，虽然输入验证可能会提供一定的深度防御。这是因为它会有效限制将在输出中出现的内容。输入验证并不总是能够防止 XSS，尤其是在您需要支持可包含任意字符的自由格式文本字段的情况下。例如，在聊天应用程序中，心型表情图标（“<3”）可能会通过验证步骤，因为它的使用频率很高。但是，不能将其直接插入到 Web 页面中，因为它包含“<”字符，该字符需要转义或以其他方式进行处理。在此情况下，消除“<”可能会降低 XSS 的风险，但是这会产生不正确的行为，因为这样就不会记录表情图标。

这可能看起来只是略有不便，但在需要表示不等式的数学论坛中，这种情况就更为重要。即使在验证中出错（例如，在 100 个输入字段中忘记一个字段），相应的编码仍有可能针对基于注入的攻击为您提供防护。只要输入验证不是孤立完成的，便仍是有用的技巧，因为它可以大大减少攻击出现的机会，使您能够检测某些攻击，并提供正确编码所无法解决的其他安全性优势。请确保在应用程序内定义良好的界面中执行输入验证。即使某个组件进行了复用或移动到其他位置，这也将有助于保护应用程序。

通过框架钓鱼

有多种减轻威胁的技巧：

[1] 策略：库或框架

使用不允许此弱点出现的经过审核的库或框架，或提供更容易避免此弱点的构造。

可用于更轻松生成正确编码的输出的库和框架示例包括 Microsoft 的 Anti-XSS 库、OWASP ESAPI 编码模块和 Apache Wicket。

[2] 了解将在其中使用数据的上下文，以及预期的编码。在不同组件之间传输数据时，或在生成可同时包含多个编码的输出（如 Web 页面或多部分邮件消息）时，这尤为重要。研究所有预期的通信协议和数据表示法以确定所需的编码策略。对于将输出到另一个 Web 页面的任何数据（尤其是从外部输入接收到的任何数据），请对所有非字母数字字符使用恰当的编码。

相同输出文档的某些部分可能需要不同的编码，具体取决于输出是在以下哪一项中：

[-] HTML 主体

[-] 元素属性（如 src="XYZ"）

[-] URI

[-] JavaScript 段

[-] 级联样式表和样式属性

请注意，“HTML 实体编码”仅适用于 HTML 主体。

请咨询 XSS Prevention Cheat Sheet

[http://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](http://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)

以获取有关所需编码和转义类型的更多详细信息。

[3] 策略：识别和减少攻击出现的机会

了解您的软件中可能出现不可信输入的所有潜在区域：参数或自变量、cookie、从网络读取的任何内容、环境变量、反向 DNS 查找、查询结果、请求头、URL 组成部分、电子邮件、文件、文件名、数据库以及向应用程序提供数据的任何外部系统。请记住，此类输入可通过 API 调用间接获取。

[4] 策略：输出编码

对于生成的每个 Web 页面，请使用并指定 ISO-8859-1 或 UTF-8 之类的字符编码。如果未指定编码，Web 浏览器可能通过猜测 Web 页面实际使用的编码来选择不同的编码。这可能导致 Web 浏览器将特定序列视为特殊序列，从而使客户机暴露在不易察觉的 XSS 攻击之下。请参阅 CWE-116 以获取与编码/转义相关的更多减轻威胁的方法。

[5] 策略：识别和减少攻击出现的机会

要帮助减轻针对用户会话 cookie 的 XSS 攻击带来的威胁，请将会话 cookie 设置为 HttpOnly。在支持 HttpOnly 功能的浏览器（如 Internet Explorer 和 Firefox 的较新版本）中，此属性可防止使用 document.cookie 的恶意客户端脚本访问用户的会话 cookie。这不是完整的解决方案，因为 HttpOnly 并不受所有浏览器支持。更重要的是，XMLHttpRequest 和其他功能强大的浏览器技术提供了对 HTTP 头的读访问权，包括在其中设置 HttpOnly 标志的 Set-Cookie 头。

[6] 策略：输入验证假定所有输入都是恶意的。使用“接受已知善意”输入验证策略：严格遵守规范的可接受输入的白名

单。拒绝任何没有严格遵守规范的输入，或者将其转换为遵守规范的内容。不要完全依赖于针对恶意或格式错误的输入的黑名单。但是，黑名单可帮助检测潜在攻击，或者确定哪些输入格式不正确，以致应当将其彻底拒绝。

执行输入验证时，请考虑所有潜在相关属性，包括长度、输入类型、可接受值的完整范围、缺失或多余输入、语法、跨相关字段的一致性以及业务规则一致性。以业务规则逻辑为例，“boat”可能在语法上有效，因为它仅包含字母数字字符，但如果预期为颜色（如“red”或“blue”），那么它无效。

动态构造 Web 页面时，请使用严格的白名单以根据请求中参数的预期值来限制字符集。所有输入都应进行验证和清理，不仅限于用户应指定的参数，而是涉及请求中的所有数据，包括隐藏字段、cookie、头、URL 本身，等等。导致 XSS 脆弱性持续存在的一个常见错误是仅验证预期会由站点重新显示的字段。常见的情况是，在请求中出现由应用程序服务器或应用程序反射的数据，而开发团队却未能预料到此情况。另外，将来的开发者可能会使用当前未反映的字段。因此，建议验证 HTTP 请求的所有部分。请注意，适当的输出编码、转义和引用是防止 XSS 的最有效解决方案，虽然输入验证可能会提供一定的深度防御。这是因为它会有效限制将在输出中出现的内容。输入验证并不总是能够防止 XSS，尤其是在您需要支持可包含任意字符的自由格式文本字段的情况下。例如，在聊天应用程序中，心型表情图标（“<3”）可能会通过验证步骤，因为它的使用频率很高。但是，不能将其直接插入到 Web 页面中，因为它包含“<”字符，该字符需要转义或以其他方式进行处理。在此情况下，消除“<”可能会降低 XSS 的风险，但是这会产生不正确的行为，因为这样就不会记录表情图标。

这可能看起来只是略有不便，但在需要表示不等式的数学论坛中，这种情况就更为重要。即使在验证中出错（例如，在 100 个输入字段中忘记一个字段），相应的编码仍有可能针对基于注入的攻击为您提供防护。只要输入验证不是孤立完成的，便仍是有用的技巧，因为它可以大大减少攻击出现的机会，使您能够检测某些攻击，并提供正确编码所无法解决的其他安全性优势。请确保在应用程序内定义良好的界面中执行输入验证。即使某个组件进行了复用或移动到其他位置，这也将有助于保护应用程序。

.Net

跨站点脚本编制

[1] 我们建议您将服务器升级至 .NET Framework 2.0（或更新的版本），它本身就包括针对跨站点脚本编制攻击进行保护的安全检查。

[2] 您可以使用验证控件，将输入验证添加到“Web 表单”页面。验证控件提供适用于标准验证的所有常见类型的易用机制（例如，测试验证日期是否有效，或验证值是否在范围内）。另外，验证控件也支持定制编写验证，可让您完整定制向用户显示错误信息的方式。验证控件可以搭配“Web 表单”页面类文件中处理的任何控件来使用，其中包括 HTML 和 Web 服务器控件。

要确保用户输入仅包含有效值，您可以使用以下验证控件中的一种：

[1] “RangeValidator”：检查用户条目（值）是否在指定的上下界限之间。您可以检查配对数字、字母字符和日期内的范围。

[2] “RegularExpressionValidator”：检查条目是否与正则表达式定义的模式相匹配。此类型的验证使您能够检查可预见的字符序列，如社会保险号码、电子邮件地址、电话号码、邮政编码等中的字符序列。
有助于阻止跨站点脚本编制的正则表达式示例：

- 可以拒绝基本跨站点脚本编制变体的正则表达式可能如下：`^([<]|<[a-zA-Z])*[<]?$`

- 拒绝上述所有字符的一般正则表达式可能如下：`^([<|>|\"|'|\%|;|:|\\|&|+]*)$`

重要注意事项：验证控件不会阻止用户输入或更改页面处理流程；它们只会设置错误状态，并产生错误消息。程序员的职责是，在执行进一步的应用程序特定操作前，测试代码中控件的状态。

有两种方法可检查用户输入的有效性：

1. 测试常规错误状态：

在您的代码中，测试页面的 `IsValid` 属性。该属性会将页面上所有验证控件的 `IsValid` 属性值汇总（使用逻辑 AND）。如果将其中一个验证控件设置为无效，那么页面属性将会返回 `false`。

2. 测试个别控件的错误状态：

在页面的“验证器”集合中循环，该集合包含对所有验证控件的引用。然后，您就可以检查每个验证控件的 `IsValid` 属性。

最后，我们建议使用 Microsoft Anti-Cross Site Scripting Library（V1.5 更高版本）对不受信任的用户输入进行编码。

Anti-Cross Site Scripting Library 显现下列方法：

[1] `HtmlEncode` — 将在 HTML 中使用的输入字符串编码

[2] `HtmlAttributeEncode` — 将在 HTML 属性中使用的输入字符串编码

- [3] JavaScriptEncode — 将在 JavaScript 中使用的输入字符串编码
- [4] UriEncode — 将在“统一资源定位器 (URL)”中使用的输入字符串编码
- [5] VisualBasicScriptEncode — 将在 Visual Basic 脚本中使用的输入字符串编码
- [6] XmlEncode — 将在 XML 中使用的输入字符串编码
- [7] XmlAttributeEncode — 将在 XML 属性中使用的输入字符串编码

如果要适当使用 Microsoft Anti-Cross Site Scripting Library 来保护 ASP.NET Web 应用程序，您必须运行下列操作：

- 第 1 步：复查生成输出的 ASP.NET 代码
- 第 2 步：判断是否包括不受信任的输入参数
- 第 3 步：判断不受信任的输入的上下文是否作为输出，判断要使用哪个编码方法
- 第 4 步：编码输出

第 3 步骤的示例：

注意：如果要使用不受信任的输入来安装 HTML 属性，便应该使用 `Microsoft.Security.Application.HtmlAttributeEncode` 方法，将不受信任的输入编码。另外，如果要在 JavaScript 的上下文中使用不受信任的输入，便应该使用 `Microsoft.Security.Application.JavaScriptEncode` 来编码。

```
// Vulnerable code
// Note that untrusted input is being treated as an HTML attribute
Literal1.Text = "<hr noshade size=[untrusted input here]>";

// Modified code
Literal1.Text = "<hr noshade size="+Microsoft.Security.Application.AntiXss.HtmlAttributeEncode([untrusted
input here])+">";
```

第 4 步骤的示例：将输出编码时，必须记住的一些重要事项：

- [1] 输出应该编码一次。
- [2] 输出的编码与实际撰写，应该尽可能接近。例如，如果应用程序读取用户输入、处理输入，再用某种形式将它重新写出，便应该紧接在撰写输出之前进行编码。

```
// Incorrect sequence
protected void Button1_Click(object sender, EventArgs e)
{
    // Read input
    String Input = TextBox1.Text;
    // Encode untrusted input
    Input = Microsoft.Security.Application.AntiXss.HtmlEncode(Input);
    // Process input
    ...
    // Write Output
    Response.Write("The input you gave was"+Input);
}

// Correct Sequence
protected void Button1_Click(object sender, EventArgs e)
{
    // Read input
    String Input = TextBox1.Text;
    // Process input
    ...
    // Encode untrusted input and write output
    Response.Write("The input you gave was"+
        Microsoft.Security.Application.AntiXss.HtmlEncode(Input));
}
```

链接注入（便于跨站请求伪造）

通过框架钓鱼

J2EE

跨站点脚本编制

**** 输入数据验证：**虽然为了用户的方便，可以提供“客户端”层数据的数据验证，但必须使用 **Servlet** 在服务器层执行验证。客户端验证本身就不安全，因为这些验证可轻易绕过，例如，通过禁用 **Javascript**。一份好的设计通常需要 **Web** 应用程序框架，以提供服务器端实用程序例程，从而验证以下内容：[1] 必需字段[2] 字段数据类型（缺省情况下，所有 **HTTP** 请求参数都是“字符串”）[3] 字段长度[4] 字段范围[5] 字段选项[6] 字段模式[7] **cookie** 值[8] **HTTP** 响应好的做法是将以上例程作为“验证器”实用程序类中的静态方法实现。以下部分描述验证器类的一个示例。

[1] 必需字段“始终”检查字段不为空，并且其长度要大于零，不包括行距和后面的空格。

如何验证必需字段的示例：

```
// Java example to validate required fields
public Class Validator {
    ...
    public static boolean validateRequired(String value) {
        boolean isFieldValid = false;
        if (value != null && value.trim().length() > 0) {
            isFieldValid = true;
        }
        return isFieldValid;
    }
    ...
}
...
String fieldValue = request.getParameter("fieldName");
if (Validator.validateRequired(fieldValue)) {
    // fieldValue is valid, continue processing request
    ...
}
```

[2] 输入的 **Web** 应用程序中的字段数据类型和输入参数欠佳。例如，所有 **HTTP** 请求参数或 **cookie** 值的类型都是“字符串”。开发者负责验证输入的数据类型是否正确。使用 **Java** 基本包装程序类，来检查是否可将字段值安全地转换为所需的基本数据类型。

验证数字字段（**int** 类型）的方式的示例：

```
// Java example to validate that a field is an int number
public Class Validator {
    ...
    public static boolean validateInt(String value) {
        boolean isFieldValid = false;
        try {
            Integer.parseInt(value);
            isFieldValid = true;
        } catch (Exception e) {
            isFieldValid = false;
        }
        return isFieldValid;
    }
    ...
}
```

```
// check if the HTTP request parameter is of type int
String fieldValue = request.getParameter("fieldName");
if (Validator.validateInt(fieldValue)) {
    // fieldValue is valid, continue processing request
    ...
}
```

好的做法是将所有 HTTP 请求参数转换为其各自的数据类型。例如，开发者应将请求参数的“integerValue”存储在请求属性中，并按以下示例所示来使用：

```
// Example to convert the HTTP request parameter to a primitive wrapper data type
// and store this value in a request attribute for further processing
String fieldValue = request.getParameter("fieldName");
if (Validator.validateInt(fieldValue)) {
    // convert fieldValue to an Integer
    Integer integerValue = Integer.getInteger(fieldValue);
    // store integerValue in a request attribute
    request.setAttribute("fieldName", integerValue);
}
...
// Use the request attribute for further processing
Integer integerValue = (Integer)request.getAttribute("fieldName");
...
```

应用程序应处理的主要 Java 数据类型：

- Byte
- Short
- Integer
- Long
- Float
- Double
- Date

[3] 字段长度“始终”确保输入参数（HTTP 请求参数或 cookie 值）有最小长度和/或最大长度的限制。以下示例验证 userName 字段的长度是否在 8 至 20 个字符之间：

```
// Example to validate the field length
public Class Validator {
    ...
    public static boolean validateLength(String value, int minLength, int maxLength) {
        String validatedValue = value;
        if (!validateRequired(value)) {
            validatedValue = "";
        }
        return (validatedValue.length() >= minLength &&
            validatedValue.length() <= maxLength);
    }
    ...
}
...
String userName = request.getParameter("userName");
if (Validator.validateRequired(userName)) {
    if (Validator.validateLength(userName, 8, 20)) {
        // userName is valid, continue further processing
        ...
    }
}
}
```

[4] 字段范围

始终确保输入参数是在由功能需求定义的范围之内。

以下示例验证输入 `numberOfChoices` 是否在 10 至 20 之间：

```
// Example to validate the field range
public Class Validator {
    ...
    public static boolean validateRange(int value, int min, int max) {
        return (value >= min && value <= max);
    }
    ...
}
...
String fieldValue = request.getParameter("numberOfChoices");
if (Validator.validateRequired(fieldValue)) {
    if (Validator.validateInt(fieldValue)) {
        int numberOfChoices = Integer.parseInt(fieldValue);
        if (Validator.validateRange(numberOfChoices, 10, 20)) {
            // numberOfChoices is valid, continue processing request
            ...
        }
    }
}
}
```

[5] 字段选项 Web 应用程序通常会为用户显示一组可供选择的选项（例如，使用 **SELECT HTML** 标记），但不能执行服务器端验证以确保选定的值是其中一个允许的选项。请记住，恶意用户能够轻易修改任何选项值。始终针对由功能需求定义的受允许的选项来验证选定的用户值。以下示例验证用户针对允许的选项列表进行的选择：

```
// Example to validate user selection against a list of options
public Class Validator {
    ...
    public static boolean validateOption(Object[] options, Object value) {
        boolean isValidValue = false;
        try {
            List list = Arrays.asList(options);
            if (list != null) {
                isValidValue = list.contains(value);
            }
        } catch (Exception e) {
        }
        return isValidValue;
    }
    ...
}
...
// Allowed options
String[] options = {"option1", "option2", "option3"};
// Verify that the user selection is one of the allowed options
String userSelection = request.getParameter("userSelection");
if (Validator.validateOption(options, userSelection)) {
    // valid user selection, continue processing request
    ...
}
}
```

[6] 字段模式

始终检查用户输入与由功能需求定义的模式是否匹配。例如，如果 `userName` 字段应仅允许字母数字字符，且不区分大小写，那么请使用以下正则表达式：`^[a-zA-Z0-9]*$`

Java 1.3 或更早的版本不包含任何正则表达式包。建议将“Apache 正则表达式包”（请参阅以下“资源”）与 Java 1.3 一起使用，以解决该缺乏支持的问题。执行正则表达式验证的示例：

```
// Example to validate that a given value matches a specified pattern
```

```

// using the Apache regular expression package
import org.apache.regexp.RE;
import org.apache.regexp.RESyntaxException;
public Class Validator {
    ...
    public static boolean matchPattern(String value, String expression) {
        boolean match = false;
        if (validateRequired(expression)) {
            RE r = new RE(expression);
            match = r.match(value);
        }
        return match;
    }
    ...
}
...
// Verify that the userName request parameter is alphanumeric
String userName = request.getParameter("userName");
if (Validator.matchPattern(userName, "[a-zA-Z0-9]*$")) {
    // userName is valid, continue processing request
    ...
}

```

Java 1.4 引进了一种新的正则表达式包（`java.util.regex`）。以下是使用新的 Java 1.4 正则表达式包的 `Validator.matchPattern` 修订版：

```

// Example to validate that a given value matches a specified pattern
// using the Java 1.4 regular expression package
import java.util.regex.Pattern;
import java.util.regex.Matcher;
public Class Validator {
    ...
    public static boolean matchPattern(String value, String expression) {
        boolean match = false;
        if (validateRequired(expression)) {
            match = Pattern.matches(expression, value);
        }
        return match;
    }
    ...
}

```

[7] cookie 值使用 `javax.servlet.http.Cookie` 对象来验证 cookie 值。适用于 cookie 值的相同的验证规则（如上所述）取决于应用程序需求（如验证必需值、验证长度等）。

验证必需 cookie 值的示例：

```

// Example to validate a required cookie value
// First retrieve all available cookies submitted in the HTTP request
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    // find the "user" cookie
    for (int i=0; i<cookies.length; ++i) {
        if (cookies[i].getName().equals("user")) {
            // validate the cookie value
            if (Validator.validateRequired(cookies[i].getValue())) {
                // valid cookie value, continue processing request
                ...
            }
        }
    }
}
}

```

[8] HTTP 响应

[8-1] 过滤用户输入要保护应用程序免遭跨站点脚本编制的攻击，请通过将敏感字符转换为其对应的字符实体来清理 HTML。这些是 HTML 敏感字符：<>"'%;)(& +

以下示例通过将敏感字符转换为其对应的字符实体来过滤指定字符串：

```
// Example to filter sensitive data to prevent cross-site scripting
public Class Validator {
    ...
    public static String filter(String value) {
        if (value == null) {
            return null;
        }
        StringBuffer result = new StringBuffer(value.length());
        for (int i=0; i<value.length(); ++i) {
            switch (value.charAt(i)) {
                case '<':
                    result.append("&lt;");
                    break;
                case '>':
                    result.append("&gt;");
                    break;
                case '"':
                    result.append("&quot;");
                    break;
                case '\'':
                    result.append("&#39;");
                    break;
                case '%':
                    result.append("&#37;");
                    break;
                case ';':
                    result.append("&#59;");
                    break;
                case '(':
                    result.append("&#40;");
                    break;
                case ')':
                    result.append("&#41;");
                    break;
                case '&':
                    result.append("&amp;");
                    break;
                case '+':
                    result.append("&#43;");
                    break;
                default:
                    result.append(value.charAt(i));
                    break;
            }
        }
        return result;
    }
    ...
}
...
// Filter the HTTP response using Validator.filter
PrintWriter out = response.getWriter();
// set output response
out.write(Validator.filter(response));
out.close();
```

Java Servlet API 2.3 引进了过滤器，它支持拦截和转换 HTTP 请求或响应。

以下示例使用 Validator.filter 来用“Servlet 过滤器”清理响应：

```
// Example to filter all sensitive characters in the HTTP response using a Java Filter.
```

```
// This example is for illustration purposes since it will filter all content in the response, including
HTML tags!
public class SensitiveCharsFilter implements Filter {
    ...
    public void doFilter(ServletRequest request,
                        ServletResponse response,
                        FilterChain chain)
        throws IOException, ServletException {

        PrintWriter out = response.getWriter();
        ResponseWrapper wrapper = new ResponseWrapper((HttpServletResponse)response);
        chain.doFilter(request, wrapper);

        CharArrayWriter caw = new CharArrayWriter();
        caw.write(Validator.filter(wrapper.toString()));

        response.setContentType("text/html");
        response.setContentLength(caw.toString().length());
        out.write(caw.toString());
        out.close();
    }
    ...
    public class CharResponseWrapper extends HttpServletResponseWrapper {
        private CharArrayWriter output;

        public String toString() {
            return output.toString();
        }

        public CharResponseWrapper(HttpServletResponse response) {
            super(response);
            output = new CharArrayWriter();
        }

        public PrintWriter getWriter(){
            return new PrintWriter(output);
        }
    }
}
}
```

[8-2] 保护 cookie

在 cookie 中存储敏感数据时，确保使用 `Cookie.setSecure`（布尔标志）在 HTTP 响应中设置 cookie 的安全标志，以指导浏览器使用安全协议（如 HTTPS 或 SSL）发送 cookie。

保护“用户”cookie 的示例：

```
// Example to secure a cookie, i.e. instruct the browser to
// send the cookie using a secure protocol
Cookie cookie = new Cookie("user", "sensitive");
cookie.setSecure(true);
response.addCookie(cookie);
```

推荐使用的 JAVA 工具用于服务器端验证的两个主要 Java 框架是：

[1] Jakarta Commons Validator（与 Struts 1.1 集成）Jakarta Commons Validator 实施所有以上数据验证需求，是强大的框架。这些规则配置在定义表单字段的输入验证规则的 XML 文件中。在缺省情况下，Struts 支持在使用 Struts“bean:write”标记撰写的所有数据上，过滤 [8] HTTP 响应中输出的危险字符。可通过设置“filter=false”标志来禁用该过滤。

Struts 定义以下基本输入验证器，但也可定义定制的验证器：

required: 如果字段包含空格以外的任何字符，便告成功。

mask: 如果值与掩码属性给定的正则表达式相匹配，便告成功。

range: 如果值在 min 和 max 属性给定的值的范围内（(value >= min) & (value <= max)），便告成功。

maxLength: 如果字段长度小于或等于 max 属性，便告成功。

minLength: 如果字段长度大于或等于 min 属性, 便告成功。

byte、short、integer、long、float、double: 如果可将值转换为对应的基本类型, 便告成功。

date: 如果值代表有效日期, 便告成功。可能会提供日期模式。

creditCard: 如果值可以是有效的信用卡号码, 便告成功。

e-mail: 如果值可以是有效的电子邮件地址, 便告成功。

使用“Struts 验证器”来验证 loginForm 的 userName 字段的示例:

```
<form-validation>
  <global>
    ...
    <validator name="required"
      classname="org.apache.struts.validator.FieldChecks"
      method="validateRequired"
      msg="errors.required">
    </validator>
    <validator name="mask"
      classname="org.apache.struts.validator.FieldChecks"
      method="validateMask"
      msg="errors.invalid">
    </validator>
    ...
  </global>
  <formset>
    <form name="loginForm">
      <!-- userName is required and is alpha-numeric case insensitive -->
      <field property="userName" depends="required,mask">
        <!-- message resource key to display if validation fails -->
        <msg name="mask" key="login.userName.maskmsg"/>
        <arg0 key="login.userName.displayName"/>
        <var>
          <var-name>mask</var-name>
          <var-value>^[a-zA-Z0-9]*$</var-value>
        </var>
      </field>
      ...
    </form>
    ...
  </formset>
</form-validation>
```

[2] JavaServer Faces 技术

“JavaServer Faces 技术”是一组代表 UI 组件、管理组件状态、处理事件和验证输入的 Java API (JSR 127)。

JavaServer Faces API 实现以下基本验证器, 但可定义定制的验证器:

validate_doubleRange: 在组件上注册 DoubleRangeValidator。

validate_length: 在组件上注册 LengthValidator。

validate_longrange: 在组件上注册 LongRangeValidator。

validate_required: 在组件上注册 RequiredValidator。

validate_stringrange: 在组件上注册 StringRangeValidator。

validator: 在组件上注册定制的 Validator。

JavaServer Faces API 定义以下 UIInput 和 UIOutput 处理器 (标记):

input_date: 接受以 java.text.Date 实例格式化的 java.util.Date。

output_date: 显示以 java.text.Date 实例格式化的 java.util.Date。

input_datetime: 接受以 java.text.DateTime 实例格式化的 java.util.Date。

output_datetime: 显示以 java.text.DateTime 实例格式化的 java.util.Date。

input_number: 显示以 java.text.NumberFormat 格式化的数字数据类型 (java.lang.Number 或基本类型)。

output_number: 显示以 java.text.NumberFormat 格式化的数字数据类型 (java.lang.Number 或基本类型)。

input_text: 接受单行文本字符串。

output_text: 显示单行文本字符串。

input_time: 接受以 java.text.DateFormat 时间实例格式化的 java.util.Date。

output_time: 显示以 java.text.DateFormat 时间实例格式化的 java.util.Date。

input_hidden: 允许页面作者在页面中包括隐藏变量。

input_secret: 接受不含空格的单行文本, 并在输入时, 将其显示为一组星号。

input_textarea: 接受多行文本。

output_errors: 显示整个页面的错误消息, 或与指定的客户端标识相关联的错误消息。

output_label: 将嵌套的组件显示为指定输入字段的标签。

output_message: 显示本地化消息。

使用 JavaServer Faces 来验证 loginForm 的 userName 字段的示例:

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
...
<jsp:useBean id="UserBean"
    class="myApplication.UserBean" scope="session" />
<f:use_faces>
    <h:form formName="loginForm" >
        <h:input_text id="userName" size="20" modelReference="UserBean.userName">
            <f:validate_required/>
            <f:validate_length minimum="8" maximum="20"/>
        </h:input_text>
        <!-- display errors if present -->
        <h:output_errors id="loginErrors" clientId="userName"/>
        <h:command_button id="submit" label="Submit" commandName="submit" /><p>
    </h:form>
</f:use_faces>
```

引用

Java API 1.3 -

<http://java.sun.com/j2se/1.3/docs/api/>

Java API 1.4 -

<http://java.sun.com/j2se/1.4/docs/api/>

Java Servlet API 2.3 -

<http://java.sun.com/products/servlet/2.3/javadoc/>

Java 正则表达式包 -

<http://jakarta.apache.org/regexp/>

Jakarta 验证器 -

<http://jakarta.apache.org/commons/validator/>

JavaServer Faces 技术 -

<http://java.sun.com/j2ee/javaserverfaces/>

** 错误处理:

许多 J2EE Web 应用程序体系结构都遵循“模型视图控制器 (MVC)”模式。在该模式中, Servlet 扮演“控制器”的角色。Servlet 将应用程序处理委派给 EJB 会话 Bean (模型) 之类的 JavaBean。然后, Servlet 再将请求转发给 JSP (视图), 以呈现处理结果。Servlet 应检查所有的输入、输出、返回码、错误代码和已知的异常, 以确保实际处理按预期进行。

数据验证可保护应用程序免遭恶意数据篡改, 而有效的错误处理策略则是防止应用程序意外泄露内部错误消息 (如异常堆栈跟踪) 所不可或缺的。好的错误处理策略会处理以下项:

[1] 定义错误

[2] 报告错误

[3] 呈现错误

[4] 错误映射

[1] 定义错误

应避免在应用程序层 (如 Servlet) 中硬编码错误消息。相反地, 应用程序应该使用映射到已知应用程序故障的错误密钥。好的做法是定义错误密钥, 且该错误密钥映射到 HTML 表单字段或其他 Bean 属性的验证规则。例如, 如果需要“user_name”字段, 其内容为字母数字, 并且必须在数据库中是唯一的, 那么就应定义以下错误密钥:

(a) ERROR_USERNAME_REQUIRED: 该错误密钥用于显示消息, 以通知用户需要“user_name”字段;

(b) ERROR_USERNAME_ALPHANUMERIC: 该错误密钥用于显示消息, 以通知用户“user_name”字段应该是字母数字;

(c) ERROR_USERNAME_DUPLICATE: 该错误密钥用于显示消息, 以通知用户“user_name”值在数据库中重复;

(d) ERROR_USERNAME_INVALID: 该错误密钥用于显示一般消息, 以通知用户“user_name”值无效;

好的做法是定义用于存储和报告应用程序错误的以下框架 Java 类:

- ErrorKeys: 定义所有错误密钥

```
// Example: ErrorKeys defining the following error keys:
```

```
// - ERROR_USERNAME_REQUIRED
// - ERROR_USERNAME_ALPHANUMERIC
// - ERROR_USERNAME_DUPLICATE
// - ERROR_USERNAME_INVALID
// ...
public Class ErrorKeys {
    public static final String ERROR_USERNAME_REQUIRED = "error.username.required";
    public static final String ERROR_USERNAME_ALPHANUMERIC = "error.username.alphanumeric";
    public static final String ERROR_USERNAME_DUPLICATE = "error.username.duplicate";
    public static final String ERROR_USERNAME_INVALID = "error.username.invalid";
    ...
}
```

- Error: 封装个别错误

```
// Example: Error encapsulates an error key.
// Error is serializable to support code executing in multiple JVMs.
public Class Error implements Serializable {

    // Constructor given a specified error key
    public Error(String key) {
        this(key, null);
    }

    // Constructor given a specified error key and array of placeholder objects
    public Error(String key, Object[] values) {
        this.key = key;
        this.values = values;
    }

    // Returns the error key
    public String getKey() {
        return this.key;
    }

    // Returns the placeholder values
    public Object[] getValues() {
        return this.values;
    }

    private String key = null;
    private Object[] values = null;
}
```

- Errors: 封装错误的集合

```
// Example: Errors encapsulates the Error objects being reported to the presentation layer.
// Errors are stored in a HashMap where the key is the bean property name and value is an
// ArrayList of Error objects.
public Class Errors implements Serializable {

    // Adds an Error object to the Collection of errors for the specified bean property.
    public void addError(String property, Error error) {
        ArrayList propertyErrors = (ArrayList)errors.get(property);
        if (propertyErrors == null) {
            propertyErrors = new ArrayList();
            errors.put(property, propertyErrors);
        }
        propertyErrors.put(error);
    }

    // Returns true if there are any errors
    public boolean hasErrors() {
        return (errors.size > 0);
    }
}
```

```

    }

    // Returns the Errors for the specified property
    public ArrayList getErrors(String property) {
        return (ArrayList)errors.get(property);
    }

    private HashMap errors = new HashMap();
}

```

以下是使用上述框架类来处理“user_name”字段验证错误的示例：

```

// Example to process validation errors of the "user_name" field.
Errors errors = new Errors();
String userName = request.getParameter("user_name");
// (a) Required validation rule
if (!Validator.validateRequired(userName)) {
    errors.addError("user_name", new Error(ErrorKeys.ERROR_USERNAME_REQUIRED));
} // (b) Alpha-numeric validation rule
else if (!Validator.matchPattern(userName, "[a-zA-Z0-9]*$")) {
    errors.addError("user_name", new Error(ErrorKeys.ERROR_USERNAME_ALPHANUMERIC));
}
else
{
    // (c) Duplicate check validation rule
    // We assume that there is an existing UserValidationEJB session bean that implements
    // a checkIfDuplicate() method to verify if the user already exists in the database.
    try {
        ...
        if (UserValidationEJB.checkIfDuplicate(userName)) {
            errors.addError("user_name", new Error(ErrorKeys.ERROR_USERNAME_DUPLICATE));
        }
    } catch (RemoteException e) {
        // log the error
        logger.error("Could not validate user for specified userName: " + userName);
        errors.addError("user_name", new Error(ErrorKeys.ERROR_USERNAME_DUPLICATE));
    }
}
// set the errors object in a request attribute called "errors"
request.setAttribute("errors", errors);
...

```

[2] 报告错误

有两种方法可报告 web 层应用程序错误：

(a) Servlet 错误机制

(b) JSP 错误机制

[2-a] Servlet 错误机制

Servlet 可通过以下方式报告错误：

- 转发给输入 JSP（已将错误存储在请求属性中），或
- 使用 HTTP 错误代码参数来调用 `response.sendError`，或
- 抛出异常

好的做法是处理所有已知应用程序错误（如 [1] 部分所述），将这些错误存储在请求属性中，然后转发给输入 JSP。输入 JSP 应显示错误消息，并提示用户重新输入数据。以下示例阐明转发给输入 JSP（`userInput.jsp`）的方式：

```

// Example to forward to the userInput.jsp following user validation errors
RequestDispatcher rd = getServletContext().getRequestDispatcher("/user/userInput.jsp");
if (rd != null) {
    rd.forward(request, response);
}

```

如果 Servlet 无法转发给已知的 JSP 页面，那么第二个选项是使用 `response.sendError` 方法，将 `HttpServletResponse.SC_INTERNAL_SERVER_ERROR`（状态码 500）作为参数，来报告错误。请参阅 `javax.servlet.http.HttpServletResponse` 的 Javadoc，以获取有关各种 HTTP 状态码的更多详细信息。返回 HTTP 错误的示例：

```
// Example to return a HTTP error code
RequestDispatcher rd = getServletContext().getRequestDispatcher("/user/userInput.jsp");
if (rd == null) {
    // messages is a resource bundle with all message keys and values
    response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR,
        messages.getMessage(ErrorKeys.ERROR_USERNAME_INVALID));
}
```

作为最后的手段，Servlet 可以抛出异常，且该异常必须是以下其中一类的子类： - `RuntimeException` - `ServletException` - `IOException`

[2-b] JSP 错误机制

JSP 页面通过定义 `errorPage` 伪指令来提供机制，以处理运行时异常，如下示例所示：

```
<%@ page errorPage="/errors/userValidation.jsp" %>
```

未捕获的 JSP 异常被转发给指定的 `errorPage`，并且原始异常设置在名称为 `javax.servlet.jsp.jspException` 的请求参数中。错误页面必须包括 `isErrorPage` 伪指令：

```
<%@ page isErrorPage="true" %>
```

`isErrorPage` 伪指令导致“exception”变量初始化为所抛出的异常对象。

[3] 呈现错误

J2SE Internationalization API 提供使应用程序资源外部化以及将消息格式化的实用程序类，其中包括：

- (a) 资源束
- (b) 消息格式化

[3-a] 资源束

资源束通过将本地化数据从使用该数据的源代码中分离来支持国际化。每一资源束都会为特定的语言环境存储键/值对的映射。

`java.util.PropertyResourceBundle` 将内容存储在外部属性文件中，对其进行使用或扩展都很常见，如下示例所示：

```
#####
# ErrorMessages.properties
#####
# required user name error message
error.username.required=User name field is required

# invalid user name format
error.username.alphanumeric=User name must be alphanumeric

# duplicate user name error message
error.username.duplicate=User name {0} already exists, please choose another one
...

```

可定义多种资源，以支持不同的语言环境（因此名为资源束）。例如，可定义 `ErrorMessages_fr.properties` 以支持该束系列的法语成员。如果请求的语言环境的资源成员不存在，那么会使用缺省成员。在以上示例中，缺省资源是 `ErrorMessages.properties`。应用程序（JSP 或 Servlet）会根据用户的语言环境从适当的资源检索内容。

[3-b] 消息格式化

J2SE 标准类 `java.util.MessageFormat` 提供使用替换占位符来创建消息的常规方法。`MessageFormat` 对象包含嵌入了格式说明符的模式字符串，如下所示：

```
// Example to show how to format a message using placeholder parameters
String pattern = "User name {0} already exists, please choose another one";
String userName = request.getParameter("user_name");
Object[] args = new Object[1];
args[0] = userName;
String message = MessageFormat.format(pattern, args);
```

以下是使用 `ResourceBundle` 和 `MessageFormat` 来呈现错误消息的更加全面的示例：

```
// Example to render an error message from a localized ErrorMessages resource (properties file)
// Utility class to retrieve locale-specific error messages
public class ErrorMessageResource {

    // Returns the error message for the specified error key in the environment locale
    public String getErrorMessage(String errorKey) {
        return getErrorMessage(errorKey, defaultLocale);
    }

    // Returns the error message for the specified error key in the specified locale
    public String getErrorMessage(String errorKey, Locale locale) {
        return getErrorMessage(errorKey, null, locale);
    }

    // Returns a formatted error message for the specified error key in the specified locale
    public String getErrorMessage(String errorKey, Object[] args, Locale locale) {
        // Get localized ErrorMessageResource
        ResourceBundle errorMessageResource = ResourceBundle.getBundle("ErrorMessages", locale);
        // Get localized error message
        String errorMessage = errorMessageResource.getString(errorKey);
        if (args != null) {
            // Format the message using the specified placeholders args
            return MessageFormat.format(errorMessage, args);
        } else {
            return errorMessage;
        }
    }

    // default environment locale
    private Locale defaultLocale = Locale.getDefaultLocale();
}

...
// Get the user's locale
Locale userLocale = request.getLocale();
// Check if there were any validation errors
Errors errors = (Errors)request.getAttribute("errors");
if (errors != null && errors.hasErrors()) {
    // Iterate through errors and output error messages corresponding to the "user_name" property
    ArrayList userNameErrors = errors.getErrors("user_name");
    ListIterator iterator = userNameErrors.iterator();
    while (iterator.hasNext()) {
        // Get the next error object
        Error error = (Error)iterator.next();
        String errorMessage = ErrorMessageResource.getErrorMessage(error.getKey(), userLocale);
        output.write(errorMessage + "\r\n");
    }
}
```

建议定义定制 JSP 标记（如 `displayErrors`），以迭代处理并呈现错误消息，如以上示例所示。

[4] 错误映射

通常情况下，“Servlet 容器”会返回与响应状态码或异常相对应的缺省错误页面。可以使用定制错误页面来指定状态码或异常与 Web 资源之间的映射。好的做法是开发不会泄露内部错误状态的静态错误页面（缺省情况下，大部分 Servlet 容器都会报告内部错误消息）。该映射配置在“Web 部署描述符（`web.xml`）”中，如以下示例所指定：

```
<!-- Mapping of HTTP error codes and application exceptions to error pages -->
<error-page>
  <exception-type>UserValidationException</exception-type>
  <location>/errors/validationError.html</error-page>
</error-page>
<error-page>
  <error-code>500</error-code>
  <location>/errors/internalError.html</error-page>
</error-page>
<error-page>
  ...
</error-page>
...
```

推荐使用的 JAVA 工具用于服务器端验证的两个主要 Java 框架是：

[1] Jakarta Commons Validator（与 Struts 1.1 集成）Jakarta Commons Validator 是 Java 框架，定义如上所述的错误处理机制。验证规则配置在 XML 文件中，该文件定义了表单字段的输入验证规则以及对应的验证错误密钥。Struts 提供国际化支持以使用资源束和消息格式化来构建本地化应用程序。

使用“Struts 验证器”来验证 `loginForm` 的 `userName` 字段的示例：

```
<form-validation>
  <global>
    ...
    <validator name="required"
      classname="org.apache.struts.validator.FieldChecks"
      method="validateRequired"
      msg="errors.required">
    </validator>
    <validator name="mask"
      classname="org.apache.struts.validator.FieldChecks"
      method="validateMask"
      msg="errors.invalid">
    </validator>
    ...
  </global>
  <formset>
    <form name="loginForm">
      <!-- userName is required and is alpha-numeric case insensitive -->
      <field property="userName" depends="required,mask">
        <!-- message resource key to display if validation fails -->
        <msg name="mask" key="login.userName.maskmsg"/>
        <arg0 key="login.userName.displayName"/>
        <var>
          <var-name>mask</var-name>
          <var-value>^[a-zA-Z0-9]*$</var-value>
        </var>
      </field>
      ...
    </form>
    ...
  </formset>
</form-validation>
```

Struts JSP 标记库定义了有条件地显示一组累计错误消息的“errors”标记，如下示例所示：

```
<%@ page language="java" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<html:html>
<head>
<body>
  <html:form action="/logon.do">
    <table border="0" width="100%">
      <tr>
        <th align="right">
          <html:errors property="username"/>
          <bean:message key="prompt.username"/>
        </th>
        <td align="left">
          <html:text property="username" size="16"/>
        </td>
      </tr>
      <tr>
        <td align="right">
          <html:submit><bean:message key="button.submit"/></html:submit>
        </td>
        <td align="right">
          <html:reset><bean:message key="button.reset"/></html:reset>
        </td>
      </tr>
    </table>
  </html:form>
</body>
</html:html>
```

[2] JavaServer Faces 技术

“JavaServer Faces 技术”是一组代表 UI 组件、管理组件状态、处理事件、验证输入和支持国际化的 Java API（JSR 127）。

JavaServer Faces API 定义“output_errors”UIOutput 处理器，该处理器显示整个页面的错误消息，或与指定的客户端标识相关联的错误消息。

使用 JavaServer Faces 来验证 loginForm 的 userName 字段的示例：

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
...
<jsp:useBean id="UserBean"
  class="myApplication.UserBean" scope="session" />
<f:use_faces>
  <h:form formName="loginForm" >
    <h:input_text id="userName" size="20" modelReference="UserBean.userName">
      <f:validate_required/>
      <f:validate_length minimum="8" maximum="20"/>
    </h:input_text>
    <!-- display errors if present -->
    <h:output_errors id="loginErrors" clientId="userName"/>
    <h:command_button id="submit" label="Submit" commandName="submit" /><p>
  </h:form>
</f:use_faces>
```

引用

Java API 1.3 -

<http://java.sun.com/j2se/1.3/docs/api/>

Java API 1.4 -

<http://java.sun.com/j2se/1.4/docs/api/>

Java Servlet API 2.3 -

<http://java.sun.com/products/servlet/2.3/javadoc/>

Java 正则表达式包 -

<http://jakarta.apache.org/regexp/>

Jakarta 验证器 -

<http://jakarta.apache.org/commons/validator/>

JavaServer Faces 技术 -

<http://java.sun.com/j2ee/jaserverfaces/>

链接注入（便于跨站请求伪造）

通过框架钓鱼

PHP

跨站点脚本编制

**** 输入数据验证：**虽然为方便用户而在客户端层上提供数据验证，但仍必须始终在服务器层上执行数据验证。客户端验证本身就不安全，因为这些验证可轻易绕过，例如，通过禁用 **Javascript**。
一份好的设计通常需要 **Web** 应用程序框架，以提供服务器端实用程序例程，从而验证以下内容：**[1]** 必需字段**[2]** 字段数据类型（缺省情况下，所有 **HTTP** 请求参数都是“字符串”）**[3]** 字段长度**[4]** 字段范围**[5]** 字段选项**[6]** 字段模式**[7]** **cookie** 值**[8]** **HTTP** 响应好的做法是实现一个或多个验证每个应用程序参数的函数。以下部分描述一些检查的示例。
[1] 必需字段“始终”检查字段不为空，并且其长度要大于零，不包括行距和后面的空格。如何验证必需字段的示例：

```
// PHP example to validate required fields
function validateRequired($input) {
    ...
    $pass = false;
    if (strlen(trim($input))>0){
        $pass = true;
    }
    return $pass;
    ...
}
...
if (validateRequired($fieldName)) {
    // fieldName is valid, continue processing request
    ...
}
```

[2] 输入的 **Web** 应用程序中的字段数据类型和输入参数欠佳。例如，所有 **HTTP** 请求参数或 **cookie** 值的类型都是“字符串”。开发者负责验证输入的数据类型是否正确。**[3]** 字段长度“始终”确保输入参数（**HTTP** 请求参数或 **cookie** 值）有最小长度和/或最大长度的限制。**[4]** 字段范围始终确保输入参数是在由功能需求定义的范围內。

[5] 字段选项 **Web** 应用程序通常会为用户显示一组可供选择的选项（例如，使用 **SELECT HTML** 标记），但不能执行服务器端验证以确保选定的值是其中一个允许的选项。请记住，恶意用户能够轻易修改任何选项值。始终针对由功能需求定义的受允许的选项来验证选定的用户值。**[6]** 字段模式

始终检查用户输入与由功能需求定义的模式是否匹配。例如，如果 **userName** 字段应仅允许字母数字字符，且不区分大小写，那么请使用以下正则表达式：**^[a-zA-Z0-9]+\$**

[7] **cookie** 值

适用于 **cookie** 值的相同的验证规则（如上所述）取决于应用程序需求（如验证必需值、验证长度等）。

[8] **HTTP** 响应**[8-1]** 过滤用户输入要保护应用程序免遭跨站点脚本编制的攻击，开发者应通过将敏感字符转换为其对应的字符实体来清理 **HTML**。这些是 **HTML** 敏感字符：**< > " ' % ;) (& +**

PHP 包含一些自动化清理实用程序函数，如 **htmlentities()**：

```
$input = htmlentities($input, ENT_QUOTES, 'UTF-8');
```

此外，为了避免“跨站点脚本编制”的 UTF-7 变体，您应该显式定义响应的 Content-Type 头，例如：

```
<?php
header('Content-Type: text/html; charset=UTF-8');

?>
```

[8-2] 保护 cookie

在 cookie 中存储敏感数据且通过 SSL 来传输时，请确保先在 HTTP 响应中设置 cookie 的安全标志。这将会指示浏览器仅通过 SSL 连接来使用该 cookie。

为了保护 cookie，您可以使用以下代码示例：

```
<$php

$value = "some_value";
$time = time()+3600;
$path = "/application/";
$domain = ".example.com";
$secure = 1;

setcookie("CookieName", $value, $time, $path, $domain, $secure, TRUE);

?>
```

此外，我们建议您使用 HttpOnly 标志。当 HttpOnly 标志设置为 TRUE 时，将只能通过 HTTP 协议来访问 cookie。这意味着无法用脚本语言（如 JavaScript）来访问 cookie。该设置可有效地帮助减少通过 XSS 攻击盗用身份的情况（虽然并非所有浏览器都支持该设置）。

在 PHP 5.2.0 中添加了 HttpOnly 标志。

引用[1] 使用 HTTP 专用 cookie 来减轻“跨站点脚本编制”的影响：

<http://msdn2.microsoft.com/en-us/library/ms533046.aspx>

[2] PHP 安全协会：

<http://phpsec.org/>

[3] PHP 和 Web 应用程序安全博客（Chris Shiflett）：

<http://shiflett.org/>

链接注入（便于跨站请求伪造）

通过框架钓鱼

低

Config your server to use the "Content-Security-Policy" header

TOC

该任务修复的问题类型

- Missing "Content-Security-Policy" header

一般

Configure your server to send the "Content-Security-Policy" header.

For Apache, see:

http://httpd.apache.org/docs/2.2/mod/mod_headers.html

For IIS, see:

<https://technet.microsoft.com/pl-pl/library/cc753133%28v=ws.10%29.aspx>

For nginx, see:

http://nginx.org/en/docs/http/nginx_headers_module.html

低

Config your server to use the "X-Content-Type-Options" header

TOC

该任务修复的问题类型

- Missing "X-Content-Type-Options" header

一般

Configure your server to send the "X-Content-Type-Options" header with value "nosniff" on all outgoing requests.

For Apache, see:

http://httpd.apache.org/docs/2.2/mod/mod_headers.html

For IIS, see:

<https://technet.microsoft.com/pl-pl/library/cc753133%28v=ws.10%29.aspx>

For nginx, see:

http://nginx.org/en/docs/http/nginx_headers_module.html

低

Config your server to use the "X-XSS-Protection" header

TOC

该任务修复的问题类型

- Missing "X-XSS-Protection" header

一般

Configure your server to send the "X-XSS-Protection" header with value "1" (i.e. Enabled) on all outgoing requests.

For Apache, see:

http://httpd.apache.org/docs/2.2/mod/mod_headers.html

For IIS, see:

<https://technet.microsoft.com/pl-pl/library/cc753133%28v=ws.10%29.aspx>

For nginx, see:

http://nginx.org/en/docs/http/nginx_headers_module.html

低

除去 Web 站点中的内部 IP 地址

TOC

该任务修复的问题类型

- 发现内部 IP 泄露模式

一般

内部 IP 通常显现在 Web 应用程序/服务器所生成的错误消息中，或显现在 HTML/JavaScript 注释中。

[1] 关闭 Web 应用程序/服务器中有问题的详细错误消息。

[2] 确保已安装相关的补丁。

[3] 确保内部 IP 信息未留在 HTML/JavaScript 注释中。

咨询

跨站点脚本编制

TOC

测试类型:

应用程序级别测试

威胁分类:

跨站点脚本编制

原因:

未对用户输入正确执行危险字符清理

安全性风险:

可能会窃取或操纵客户会话和 **cookie**，它们可能用于模仿合法用户，从而使黑客能够以该用户身份查看或变更用户记录以及执行事务

受影响产品:

CWE:

79

X-Force:

6784

引用:

[CERT Advisory CA-2000-02](#)

[Microsoft How To: Prevent Cross-Site Scripting Security Issues \(Q252985\)](#)

[Microsoft How To: Prevent Cross-Site Scripting in ASP.NET](#)

[Microsoft How To: Protect From Injection Attacks in ASP.NET](#)

[Microsoft How To: Use Regular Expressions to Constrain Input in ASP.NET](#)

[Microsoft .NET Anti-Cross Site Scripting Library](#)

跨站点脚本编制培训模块

技术描述:

AppScan 检测到应用程序未对用户可控制的输入正确进行无害化处理，就将其放置到充当 Web 页面的输出中。这可被跨站点脚本编制攻击利用。

在以下情况下会发生跨站点脚本编制 (XSS) 脆弱性:

[1] 不可信数据进入 Web 应用程序, 通常来自 Web 请求。

[2] Web 应用程序动态生成了包含此不可信数据的 Web 页面。

[3] 页面生成期间, 应用程序不会禁止数据包含可由 Web 浏览器执行的内容, 例如 JavaScript、HTML 标记、HTML 属性、鼠标事件、Flash 和 ActiveX。

[4] 受害者通过 Web 浏览器访问生成的 Web 页面, 该页面包含已使用不可信数据注入的恶意脚本。

[5] 由于脚本来自 Web 服务器发送的 Web 页面, 因此受害者的 Web 浏览器在 Web 服务器的域的上下文中执行恶意脚本。

[6] 这实际违反了 Web 浏览器的同源策略的意图, 该策略声明一个域中的脚本不应该能够访问其他域中的资源或运行其他域中的代码。

一旦注入恶意脚本后, 攻击者就能够执行各种恶意活动。攻击者可能将私有信息 (例如可能包含会话信息的 cookie) 从受害者的机器传输给攻击者。攻击者可能以受害者的身份将恶意请求发送到 Web 站点, 如果受害者具有管理该站点的管理员特权, 这可能会对站点尤其危险。

网络钓鱼攻击可用于模仿可信站点, 并诱导受害者输入密码, 从而使攻击者能够危及受害者在该 Web 站点上的帐户。

最后, 脚本可利用 Web 浏览器本身中的脆弱性, 可能是接管受害者的机器 (有时称为“路过式入侵”)。

主要有三种类型的 XSS:

类型 1: 反射的 XSS (也称为“非持久性”)

服务器直接从 HTTP 请求中读取数据, 并将其反射回 HTTP 响应。在发生反射的 XSS 利用情况时, 攻击者会导致受害者向易受攻击的 Web 应用程序提供危险内容, 然后该内容会反射回受害者并由 Web 浏览器执行。传递恶意内容的最常用机制是将其作为参数包含在公共发布或通过电子邮件直接发送给受害者的 URL 中。以此方式构造的 URL 构成了许多网络钓鱼方案的核心, 攻击者借此骗取受害者的信任, 使其访问指向易受攻击的站点的 URL。在站点将攻击者的内容反射回受害者之后, 受害者的浏览器将执行该内容。

类型 2: 存储的 XSS (也称为“持久性”)

应用程序在数据库、消息论坛、访问者日志或其他可信数据存储中存储危险数据。在以后某个时间, 危险数据会读回到应用程序并包含在动态内容中。从攻击者的角度来看, 注入恶意内容的最佳位置是向许多用户或特别感兴趣的用户显示的区域。感兴趣的用户通常在应用程序中具有较高的特权, 或者他们会与对攻击者有价值的敏感数据进行交互。如果其中某个用户执行恶意内容, 那么攻击者就有可能能够以该用户的身份执行特权操作, 或者获取对属于该用户的敏感数据的访问权。例如, 攻击者可能在日志消息中注入 XSS, 而管理员查看日志时可能不会正确处理该消息。

类型 0: 基于 DOM 的 XSS

在基于 DOM 的 XSS 中, 客户机执行将 XSS 注入页面的操作; 在其他类型中, 注入操作由服务器执行。基于 DOM 的 XSS 中通常涉及发送到客户机的由服务器控制的可信脚本, 例如, 在用户提交表单之前对表单执行健全性检查的 Javascript。如果服务器提供的脚本处理用户提供的数据, 然后将数据注入回 Web 页面 (例如通过动态 HTML), 那么基于 DOM 的 XSS 就有可能发生。以下示例显示了在响应中返回参数值的脚本。

参数值通过使用 GET 请求发送到脚本, 然后在 HTML 中嵌入的响应中返回。

```
[REQUEST]
GET /index.aspx?name=JSmith HTTP/1.1
```

```
[RESPONSE]
HTTP/1.1 200 OK
Server: SomeServer
Date: Sun, 01 Jan 2002 00:31:19 GMT
Content-Type: text/html
Accept-Ranges: bytes
Content-Length: 27
```

```
<HTML>
Hello JSmith
</HTML>
```

攻击者可能会利用类似以下情况的攻击:

```
[ATTACK REQUEST]
GET /index.aspx?name=>"<script>alert('PWND')</script> HTTP/1.1
```

```
[ATTACK RESPONSE]
HTTP/1.1 200 OK
Server: SomeServer
Date: Sun, 01 Jan 2002 00:31:19 GMT
Content-Type: text/html
Accept-Ranges: bytes
Content-Length: 83

<HTML>
Hello >"'><script>alert('PWND')</script>
</HTML>
```

在这种情况下，JavaScript 代码将由浏览器执行（>"'> 部分在此处并不相关）。

链接注入（便于跨站请求伪造）

TOC

测试类型：

应用程序级别测试

威胁分类：

内容电子欺骗

原因：

未对用户输入正确执行危险字符清理

安全性风险：

- 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息
- 可能会窃取或操纵客户会话和 **cookie**，它们可能用于模仿合法用户，从而使黑客能够以该用户身份查看或变更用户记录以及执行事务
- 可能会在 **Web** 服务器上上载、修改或删除 **Web** 页面、脚本和文件

受影响产品：

CWE:

74

X-Force:

6784

引用:

[OWASP 文章](#)

[跨站点请求伪造常见问题 \(FAQ\)](#)

[跨站点请求伪造培训模块](#)

技术描述:

该软件使用受外部影响的输入来构造命令、数据结构或记录的全部或一部分，但未能对可能修改其解析或解释方式的元素进行无害化处理。

“链接注入”是通过在某个站点中嵌入外部站点的 URL，或者在易受攻击的站点中嵌入脚本的 URL，从而修改该站点的内容。在易受攻击的站点中嵌入 URL 后，攻击者能够将其作为发起针对其他站点（以及针对这个易受攻击的站点本身）的攻击的平台。

其中一些可能的攻击需要用户在攻击期间登录站点。通过从易受攻击的站点本身发起这些攻击，攻击者成功的可能性更高，因为用户更倾向于登录。

“链接注入”脆弱性是未对用户输入进行充分清理所导致的结果，该输入以后会在站点响应中返回给用户。这样一来，攻击者能够将危险字符注入响应中，从而有可能嵌入 URL，以及做出其他可能的内容修改。

以下是“链接注入”的示例（我们假设站点“[www.vulnerable.com](#)”有一个名为“name”的参数，用于问候用户）。

下列请求：[HTTP://www.vulnerable.com/greet.asp?name=John Smith](http://www.vulnerable.com/greet.asp?name=John+Smith)

会生成下列响应:

```
<HTML>
<BODY>
    Hello, John Smith.
</BODY>
</HTML>
```

然而，恶意的用户可以发送下列请求:

[HTTP://www.vulnerable.com/greet.asp?name=](http://www.vulnerable.com/greet.asp?name=)

这会返回下列响应:

```
<HTML>
<BODY>
    Hello, <IMG SRC="http://www.ANY-SITE.com/ANY-SCRIPT.asp">.
</BODY>
</HTML>
```

如以上示例所示，攻击者有可能导致用户浏览器向攻击者企图攻击的几乎任何站点发出自动请求。因此，“链接注入”脆弱性可用于发起几种类型的攻击:

[+] 跨站点请求伪造

[+] 跨站点脚本编制

[+] 网络钓鱼

通过框架钓鱼

TOC

测试类型:

应用程序级别测试

威胁分类:

内容电子欺骗

原因:

未对用户输入正确执行危险字符清理

安全性风险:

可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息

受影响产品:

CWE:

79

X-Force:

52829

引用:

FTC Consumer Alert - "How Not to Get Hooked by a 'Phishing' Scam"

技术描述:

网络钓鱼是一种社会工程技巧，其中攻击者伪装成受害者可能会与其进行业务往来的合法实体，以便提示用户透露某些机密信息（往往是认证凭证），而攻击者以后可以利用这些信息。网络钓鱼在本质上是一种信息收集形式，或者说是信息的“渔猎”。

攻击者有可能注入含有恶意内容的 **frame** 或 **iframe** 标记。如果用户不够谨慎，就有可能浏览该标记，却意识不到自己会离开原始站点而进入恶意的站点。之后，攻击者便可以诱导用户再次登录，然后获取其登录凭证。

由于伪造的站点嵌入在原始站点中，这样攻击者的网络钓鱼企图就披上了更容易让人轻信的外衣。

Missing "Content-Security-Policy" header

TOC

测试类型:

应用程序级别测试

威胁分类:

信息泄露

原因:

Web 应用程序编程或配置不安全

安全性风险:

- 可能会收集有关 Web 应用程序的敏感信息，如用户名、密码、机器名和/或敏感文件位置
- 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息

受影响产品:

引用:

[List of useful HTTP headers](#)

[An Introduction to Content Security Policy](#)

技术描述:

The "Content-Security-Policy" header is designed to modify the way browsers render pages, and thus to protect from various cross-site injections, including Cross-Site Scripting. It is important to set the header value correctly, in a way that will not prevent proper operation of the web site. For example, if the header is set to prevent execution of inline JavaScript, the web site must not use inline JavaScript in it's pages.

Missing "X-Content-Type-Options" header

TOC

测试类型:

应用程序级别测试

威胁分类:

信息泄露

原因:

Web 应用程序编程或配置不安全

安全性风险:

- 可能会收集有关 Web 应用程序的敏感信息，如用户名、密码、机器名和/或敏感文件位置
- 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息

受影响产品:

引用:

[List of useful HTTP headers](#)

[Reducing MIME type security risks](#)

技术描述:

The "X-Content-Type-Options" header (with "nosniff" value) prevents IE and Chrome from ignoring the content-type of a response.

This action may prevent untrusted content (e.g. user uploaded content) from being executed on the user browser (after a malicious naming, for example).

Missing "X-XSS-Protection" header

TOC

测试类型:

应用程序级别测试

威胁分类:

信息泄露

原因:

Web 应用程序编程或配置不安全

安全性风险:

- 可能会收集有关 **Web** 应用程序的敏感信息，如用户名、密码、机器名和/或敏感文件位置
- 可能会劝说初级用户提供诸如用户名、密码、信用卡号、社会保险号等敏感信息

受影响产品:

引用:

[List of useful HTTP headers](#)
[IE XSS Filter](#)

技术描述:

The "X-XSS-Protection" header forces the Cross-Site Scripting filter into Enable mode, even if disabled by the user. This filter is built into most recent web browsers (IE 8+, Chrome 4+), and is usually enabled by default. Although it is not designed as first and only defence against Cross-Site Scripting, it acts as an additional layer of protection.

发现内部 IP 泄露模式

TOC

测试类型:

应用程序级别测试

威胁分类:

信息泄露

原因:

Web 应用程序编程或配置不安全

安全性风险:

可能会收集有关 **Web** 应用程序的敏感信息，如用户名、密码、机器名和/或敏感文件位置

受影响产品:

CWE:

200

X-Force:

52657

引用:

CWE-200: 信息泄露（信息披露）

技术描述:

AppScan 检测到包含内部 IP 地址的响应。

内部 IP 定义为下列 IP 范围内的 IP:

10.0.0.0 - 10.255.255.255

172.16.0.0 - 172.31.255.255

192.168.0.0 - 192.168.255.255

对攻击者而言，泄露内部 IP 非常有价值，因为它显示了内部网络的 IP 地址方案。知道内部网络的 IP 地址方案，可以辅助攻击者策划出对内部网络进一步的攻击。

应用程序数据

已访问的 URL 8

TOC

URL
http://10.62.233.181:9425/
http://10.62.233.181:9425/index.html
http://10.62.233.181:9425/mfs.cgi
http://10.62.233.181:9425/mfs.cgi?masterhost=mfsmaster&masterport=9421&mastercontrolport=9419&mastername=MooseFS
http://10.62.233.181:9425/acidtab.js
http://10.62.233.181:9425/mfs.cgi?sections=HD
http://10.62.233.181:9425/mfs.cgi?sections=CS
http://10.62.233.181:9425/mfs.cgi?sections=CS IN

参数 5

TOC

名称	值	URL	类型
sections	HD CS CS IN	http://10.62.233.181:9425/mfs.cgi?sections=HD	简单链接
masterhost	mfsmaster	http://10.62.233.181:9425/mfs.cgi?masterhost=mfsmaster&masterport=9421&mastercontrolport=9419&mastername=MooseFS	简单链接
amp;mastercontrolport	9419	http://10.62.233.181:9425/mfs.cgi?masterhost=mfsmaster&masterport=9421&mastercontrolport=9419&mastername=MooseFS	简单链接
amp;mastername	MooseFS	http://10.62.233.181:9425/mfs.cgi?masterhost=mfsmaster&masterport=9421&mastercontrolport=9419&mastername=MooseFS	简单链接
amp;masterport	9421	http://10.62.233.181:9425/mfs.cgi?masterhost=mfsmaster&masterport=9421&mastercontrolport=9419&mastername=MooseFS	简单链接

失败的请求 0

TOC

已过滤的 URL 56

URL	原因
http://10.62.233.181:9425/mfs.css	文件扩展名
http://10.62.233.181:9425/logomini.png	文件扩展名
http://www.moosefs.org/	未测试的 Web Server
http://moosefs.com/products.html	未测试的 Web Server
http://10.62.233.181:9425/mfs.cgi?sections=HD IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=EX	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=EX IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MS	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MS IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MO	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MO IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=QU	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=QU IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MC	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MC IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CC	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CC IN	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=CS	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=CS IN	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=HD	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=HD IN	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=EX	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=EX IN	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=MS	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=MS IN	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=MO	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=MO IN	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419	路径限制

19&mastername=MooseFS§ions=QU	
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=QU IN	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=MC	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=MC IN	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=CC	路径限制
http://10.62.233.181:9425/mfs.cgi?masterport=9421&masterhost=mfsmaster&mastercontrolport=9419&mastername=MooseFS§ions=CC IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS HD	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=EX HD	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MS HD	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MO HD	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=QU HD	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=MC HD	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CC HD	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS EX	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS MS	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS MO	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS QU	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS MC	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS CC	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS&CSmaintenanceon=10.93.211.36:9422	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS HD IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS EX IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS MS IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS MO IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS QU IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS MC IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS CC IN	路径限制
http://10.62.233.181:9425/mfs.cgi?sections=CS IN&CSmaintenanceon=10.93.211.36:9422	路径限制

注释 1

TOC

URL	注释
http://10.62.233.181:9425/mfs.cgi	end of container

JavaScript 13

TOC

URL / 代码

<http://10.62.233.181:9425/index.html>

```
document.location.href="mfs.cgi"
// the above uses default parameter values; full invocation example is shown below
//
document.location.href="mfs.cgi?
masterhost=mfsmaster&masterport=9421&mastercontrolport=9419&mastername=MooseFS"
```

<http://10.62.233.181:9425/mfs.cgi>

```
//<!--//><![CDATA[//><!--
var bar_labels = ['node hash','others'];
var bar_tooltips = ['node hash (99.95 %)','other memory segments (0.05 %)'];
//--><![>>
```

<http://10.62.233.181:9425/mfs.cgi>

```
//<!--//><![CDATA[//><!--
function bar_refresh() {
    var b = document.getElementById("bar");
    var i,j,x;
    if (b) {
        var x = b.getElementsByTagName("td");
        for (i=0 ; i<x.length ; i++) {
            x[i].innerHTML = "";
        }
        for (i=0 ; i<x.length ; i++) {
            var width = x[i].clientWidth;
            var label = bar_labels[i];
            var tooltip = bar_tooltips[i];
            x[i].innerHTML = "<a title='" + tooltip + "'>" + label + "</a>";
            if (width<x[i].clientWidth) {
                x[i].innerHTML = "<a title='" + tooltip + "'>#8230;</a>";
                if (width<x[i].clientWidth) {
                    x[i].innerHTML = "<a title='" + tooltip + "'>#8226;</a>";
                    if (width<x[i].clientWidth) {
                        x[i].innerHTML = "<a title='" + tooltip + "'>.</a>";
                        if (width<x[i].clientWidth) {
                            x[i].innerHTML = "";
                        }
                    }
                } else {
                    for (j=1 ; j<bar_labels[i].length-1 ; j++) {
                        x[i].innerHTML = "<a title='" + tooltip + "
">" + label.substring(0,j) + "&#8230;</a>";
                        if (width<x[i].clientWidth) {
                            break ;
                        }
                    }
                    x[i].innerHTML = "<a title='" + tooltip + "'>" +
label.substring(0,j-1) + "&#8230;</a>";
                }
            }
        }
    }

    function bar_add_event(obj,type,fn) {
        if (obj.addEventListener) {
```



```
        obj.addEventListener(type, fn, false);
    } else if (obj.attachEvent) {
        obj.attachEvent('on'+type, fn);
    }
}

bar_add_event(window,"load",bar_refresh);
bar_add_event(window,"resize",bar_refresh);
//--><[!]]>
```

http://10.62.233.181:9425/mfs.cgi

```
acid_tab.switchdisplay('mfsmatrix','matrix_vis',1)
```

http://10.62.233.181:9425/mfs.cgi

```
acid_tab.switchdisplay('mfsmatrix','matrix_vis',0)
```

http://10.62.233.181:9425/mfs.cgi

```
acid_tab.switchdisplay('mfshdd','hddaddrname_vis',1)
```

http://10.62.233.181:9425/mfs.cgi

```
acid_tab.switchdisplay('mfshdd','hddaddrname_vis',0)
```

http://10.62.233.181:9425/mfs.cgi

```
acid_tab.switchdisplay('mfshdd','hddperiod_vis',1)
```

http://10.62.233.181:9425/mfs.cgi

```
acid_tab.switchdisplay('mfshdd','hddperiod_vis',2)
```

http://10.62.233.181:9425/mfs.cgi

```
acid_tab.switchdisplay('mfshdd','hddperiod_vis',0)
```

http://10.62.233.181:9425/mfs.cgi

```
acid_tab.switchdisplay('mfshdd','hddtime_vis',1)
```

<http://10.62.233.181:9425/mfs.cgi>

```
acid_tab.switchdisplay('mfshdd','hddtime_vis',0)
```

<http://10.62.233.181:9425/acidtab.js>

```
/*
 * Copyright (C) 2016 Jakub Kruszona-Zawadzki, Core Technology Sp. z o.o.
 *
 * This file is part of MooseFS.
 *
 * MooseFS is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, version 2 (only).
 *
 * MooseFS is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with MooseFS; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02111-1301, USA
 * or visit http://www.gnu.org/licenses/gpl-2.0.html
 */

if (typeof acid_ready == "undefined") {
  acid_ready = {
    ready : 0,
    functions : new Array(),
    docready : function() {
      if (acid_ready.ready==0) {
        acid_ready.ready = 1;
        var i;
        for (i=0 ; i<acid_ready.functions.length ; i++) {
          acid_ready.functions[i]();
        }
      }
    },
    domloaded : function(e) {
      acid_ready.docready();
    },
    loaded : function(e) {
      acid_ready.docready();
    },
    readystate : function(e) {
      if (document.readyState=="complete" || document.readyState=="interactive") {
        acid_ready.docready();
      }
    },
    readytest : function() {
      if (acid_ready.ready==0) {
        if (typeof document.readyState!="undefined") {
          if (document.readyState=="complete" ||
document.readyState=="interactive") {
            acid_ready.docready();
          } else {
            setTimeout("acid_ready.readytest()",100);
          }
        }
        return;
      }
      if (typeof document.documentElement!="undefined" && typeof
document.documentElement.doScroll!="undefined") {
        try {
          document.documentElement.doScroll('left');
```

```

        } catch (e) {
            setTimeout("acid_ready.readystatechange()",100);
            return;
        }
    }
    acid_ready.docready();
}

register : function(fn) {
    if (acid_ready.ready) {
        fn();
    } else {
        var l = acid_ready.functions.length;
        acid_ready.functions[l]=fn;
    }
}

init : function() {
    if (window.addEventListener) {
        document.addEventListener("DOMContentLoaded",acid_ready.domloaded, false);
        document.addEventListener("readystatechange",acid_ready.readystate, false);
        window.addEventListener("load",acid_ready.loaded, false);
    } else if (window.attachEvent) {
        document.attachEvent("onDOMContentLoaded",acid_ready.domloaded);
        document.attachEvent("onreadystatechange",acid_ready.readystate);
        window.attachEvent("onload",acid_ready.loaded);
    }
    setTimeout("acid_ready.readystatechange()",100);
}

acid_ready.init();
}

if (typeof acid_tab == "undefined") {
    acid_tab = {
        init: function() {
            var tabs,i;
            if (!document.createElement || !document.getElementsByTagName) {
                return;
            }
            tabs = document.getElementsByTagName('table');
            if (tabs) {
                for (i=0 ; i<tabs.length ; i++) {
                    if (tabs[i].className.search(/\\bacid_tab\\b/) != -1) {
                        acid_tab.preparetab(tabs[i]);
                    }
                }
            }
        },
        preparetab: function(table) {
            var i,j,k,x,h,m,s,p,c,z,r;
            // find thead using 'TH' node names
            if (table.getElementsByTagName('thead').length == 0) {
                var thead = document.createElement('thead');
                while (table.rows.length>0 && table.rows[0].cells[0].nodeName=="TH") {
                    thead.appendChild(table.rows[0]);
                }
                table.insertBefore(thead,table.firstChild);
            }
            // create tHead if necessary
            if (typeof table.tHead == "undefined") {
                table.tHead = table.getElementsByTagName('thead')[0];
            }
            // backup and remove extra bodies
            table.acid_tab_tbodybackup = new Array();
            for (i=0 ; i<table.tBodies.length ; i++) {
                table.acid_tab_tbodybackup[i] = table.tBodies[i];
            }
            while (table.tBodies.length>1) {
                table.removeChild(table.tBodies[1]);
            }
            // no body? - exit
            if (table.tBodies.length==0) {
                return;
            }
            // check settings
            m = table.className.match(/\\bacid_tab_zebra_([a-zA-Z0-9]+)_([a-zA-Z0-9]+)\\b/);
            if (m) {

```

```

        table.acid_tab_zebra = new Array();
        table.acid_tab_zebra[0] = m[1];
        table.acid_tab_zebra[1] = m[2];
    }
    if (table.className.search(/\bacid_tab_noindicator\b/)!=-1) {
        table.acid_tab_indicator = 0;
    } else {
        table.acid_tab_indicator = 1;
    }
    m = table.className.match(/\bacid_tab_storageid_([a-zA-Z0-9]+\b/);
    if (m) {
        table.acid_tab_storageid = m[1];
    } else {
        table.acid_tab_storageid = "";
    }
    z = -1;
    r = 0;
    // scan storage
    if (table.acid_tab_storageid!="" && typeof sessionStorage != "undefined") {
        p = "switchdisplay_"+table.acid_tab_storageid+"_";
        for (i=0 ; i<sessionStorage.length ; i++) {
            k = sessionStorage.key(i);
            if (k.slice(0,p.length)==p) {
                j = parseInt(sessionStorage.getItem(k));
                acid_tab.switchdisplay(table,k.slice(p.length),j);
            }
        }
        z = sessionStorage.getItem('switchbody_'+table.acid_t...

```

cookie 0

TOC

名称	首先设置	域	安全
值	请求的 URL		到期