<<黑客大曝光>>

图书基本信息

书名：<<黑客大曝光>>

13位ISBN编号：9787121066696

10位ISBN编号：7121066696

出版时间：2008-6

出版时间：电子工业出版社

作者：（美）斯卡姆布雷，（美）施玛，（美）西玛 著

页数：462

字数：512000

前言

经过近一年的辛苦工作，《黑客大曝光——Web应用安全机密与解决方案（第2版）》的翻译终于告一段落。
看着厚厚的书稿，无疑有一些轻松的感觉。
正是“纳清风台榭开怀，傍流水亭轩赏心。”
记得四年前，自己的第一本书《缓冲区溢出教程》出版时，漏洞满天飞，蠕虫肆虐，肉鸡成群。
几年过去了，现在可喜地看到，政府、企业、网管的安全意识越来越强，安全措施越来越完善，安全技术也水涨船高，利用一个漏洞扫描器就可以忽悠单子的时代一去不复返了。
不过，安全所处的大环境也急剧地变化，新型的网络应用越来越多，电子商务越来越发达；而另一

<<黑客大曝光>>

内容概要

在网络技术和电子商务飞速发展的今天，Web应用安全面临着前所未有的挑战。
本书凝聚了作者们超过30年的Web安全从业经验，详细剖析了Web应用的安全漏洞，攻击手法和对抗措施，一步步的教授如何防御邪恶的攻击，并协助读者理解黑客的思考过程。
　本书分为13章，书后带有附录和详细的英汉对照索引。
本书是网络管理员、系统管理员的必备宝典，也是电子商务从业者、网络爱好者和企业管理者的参考书籍。

作者简介

Joel Scambray拥有信息系统安全专家认证（CISSP），有15年的信息安全经验，包括：在微软和安永国际会计公司担任高级管理角色，与人创办Foundstone公司，担任“财富500强”企业的技术顾问，与人合著“黑客大曝光”（Hacking Exposed）系列畅销书。

Mike Shema是NT Object

<<黑客大曝光>>

书籍目录

进一步阅读附录A Web应用程序的安全检查列表附录B Web攻击工具和攻击技术清单附录C URLScan和ModSecurity附录D 关于本书的配套网站索引

章节摘录

第6章 输入验证攻击　　输入验证是Web应用程序安全防范的首道防线。
许多攻击，如SQL注入，脚本攻击（包括了跨站脚本），以及详细错误信息的泄漏，都是由于攻击者向程序提交了未曾预料到的输入类型而造成的。
输入验证就是为了确保输入数据的格式和类型都是程序所需的。
如果不进行严格的检查以减少误操作，程序的完整性和它的信息就有可能会受到损害。
想象一个应用程序中的购物车，我们考虑其信用卡字段。
首先，信用卡号码只包含数字；其次，大多数信用卡号码都是16位的数字，但也有一些是小于16位的。
因此，如果进行输入验证，第一项应该是长度检查：输入的数据是否为14~16位的字符；第二项检查应该是内容检查：输入的数据是否含非数字的字符。
我们可以在系统中加上另一项检查：输入的数据是否是一个合法的信用卡号码。
“0000111122223333”显然不是一个信用卡号码，但“4435786912639983”呢？
我们写一个简单的校验和程序就可以判断一个16位数字是否符合合法的信用卡号码要求。
另外，一些常识性知识也可以用来验证，比如，一个15位的信用卡号码应该以3开头，第二位应该为4或者7。
这个信用卡的例子演示了如何测试输入为一串数字的合法性。
但请注意，这个例子没有尝试判断信用卡号码是否匹配用户名或用户地址，而只是尝试验证号码本身的合法性。
本章内容着重于两点：一是完全信任用户提供的数据会带来的危险；二是如果没有恰当的限制期望的数据类型，攻击者攻击应用程序各种的方法。
数据验证可能会非常的复杂，但它却是应用程序安全基础中的基础。
应用程序的开发者应该事先预料到用户在表单字段中所有可能的输入值。
刚才我们提到了验证信用卡号码的三种简单方法：长度、内容和校验和。
这样的验证过程可以放在HTML页面中，以JavaScript实现，并运行在SSL之上。
基于JavaScript的解决方案非常简单，也的确是开发人员最常使用的方法之一。
但在接下来的几节里，我们将会看到，客户端的输入验证可以被绕过，而SSL只是起到保护Web传输的保密性。
换句话说，我们不能信任Web浏览器所做的安全检查工作，通过SSL加密连接不会对提交到应用程序的数据内容造成任何的影响。
6.1　预料意外的情况　　输入验证最大的一个安全漏洞就是将验证流程用JavaScript实现并放在浏览器中。
使用客户端的脚本语言实现验证流程似乎是合情合理的，原因首先是验证流程不必在服务端实现；其次是，客户端验证容易实现且被大多数Web浏览器支持（虽然也有很个别的浏览器不支持）；更重要的是，这样就把服务端所要做的大量工作移到了客户端，是应用程序的一种成功。
但Web浏览器是不可信、不可控的环境，所有传入传出的数据都可在传输途中被修改，不管是否存在输入验证流程。
因此，用于购买新的Web服务器来处理服务端输入验证的支出，远远小于恶意用户使用%0a的小手段来损害到应用程序安全带来的代价。
输入验证的攻击可能针对应用程序的不同方面。
理解黑客如何攻击不完整的验证流程是非常重要的，因为他们带来的危险远不只“垃圾数据”错误那么简单。
数据存储：包括SQL注入攻击中使用的字符。
这些字符可以改写数据库查询语句，导致执行攻击者定制的行为。
产生的错误能泄漏出各种信息，比如应用程序采用的编程语言，甚至是应用程序发送到数据库的具体SQL查询语句。

冒充其他用户：包括跨站脚本以及与“钓鱼”相关的攻击。
攻击者可以通过提交数据改写HTML，从而窃取其他用户的信息，或者引诱用户泄露他们的敏感信息。
控制Web服务器：此类攻击因操作系统不同而不同。
比如插入分号，能在UNIX的Web服务器上执行任意指令。
应用程序本来要在Web服务器上执行命令，但通过特殊的字符，可以欺骗它执行其他的命令。
泄露应用程序内容：攻击者能通过产生的错误来泄露程序的语言信息。
其他攻击方法也可以绕过浏览器对文件类型的限制。
比如许多Nimda蠕虫的变种，使用了斜线（用来分隔目录）其他的编码方式来绕过IIS的安全检查，从而可以访问Web根目录以外的文件。
缓冲区溢出攻击：缓冲区溢出攻击已经困扰着程序多年了，Web应用程序也不例外。
此类攻击包括向一个变量或者字段中填充尽可能多的字符，然后观察其结果。
这样可能会导致应用程序崩溃或者停止执行任意命令。
缓冲区溢出更多在编译语言，如C或C++中，被关注，而在Perl或Python等解释性语言中则很少注意。
基于.NET和Java的Web平台，由于不允许程序员直接操纵堆栈和堆分配（这是缓冲区溢出的条件），从而导致应用层面的缓冲区溢出非常困难。
缓冲区溢出可能会在特定的语言平台上长期存在。
获得任意数据的访问权限：一个用户可以访问另一个同级用户的信息，比如一个顾客可以查看另外一个顾客的账单信息；一个用户可以访问某些特权数据，比如匿名用户能够枚举、创建或者删除用户。
数据访问同样适用于受保护的文件或程序中的管理员领域。
6.2　在哪里寻找攻击载体　　每一个GET和POST的参数都可以用来做输入验证攻击，更改参数，不管它们来自表单还是应用程序，都是一项很细微的工作。
最易被攻击的点是输入字段。
通常这些字段包括登录名、密码、地址、电话、信用卡号以及搜索，其他使用下拉菜单的字段也不应该被忽略。
第一步需要枚举出这些字段以及它们大概的输入类型。
不要错误地认为输入验证攻击的目标只能是用户需要完成填写的字段，事实上，GET和POST请求的任何变量都有可能被攻击。
对一个很有价值的目标，攻击者在攻击前会深入全面的调查程序的文件、参数和表单字段。
Cookie是另一类攻击目标。
Cookie本来包含了用户不能故意操纵的值，但也有可能被用于SQL注入或冒充其他的用户。
Cookie只是一种特殊的HTTP头。
事实上，任何HTTP头都是输入验证攻击的载体。
HTTP响应头截断是另一种以HTTP头为攻击目标的例子，它将正常地响应截断，注入伪造的头部集（通常是Cookie或控制缓存，会给客户端带来很大的破坏）。
让我们仔细分析一下HTTP响应头截断攻击。
攻击的目标是使用参数作为转向指示的程序。
例如，一个存在潜在漏洞的URL如下：　　http://Website/redirect.cgi?page=http://Website/welcome.cgi
一个很好的输入验证流程需要确认page参数的值是一个合法的URL。
但如果可以包含任意的字符，那么参数可以被改写为如下的样子：　　http://Website/redirect.cgi?page=0d%0aContent-Type:%20text/　　html%0d%0aHTTP/1.1%20200%20OK%0d%0aContent-Type:%20text/
html%0d%0a%0d%0a%3chtml%3eHello, world!%3c/html%3e　　Page原来的值被一串字符所替换了，这些字符模拟了来自一个Web服务器的HTTP响应头，包括了一个简单的HTML字符串“Hello, world!”。
将编码后的字符替换掉，恶意载荷就更容易理解了：　　Content-Type: text/html　　HTTP/1.1 200 OK
Content-Type: text/html　　Hello, world!　　最终的攻击结果是浏览器显示了伪造的HTML内容，

而不是期望转向的网页。
这个例子似乎并无破坏作用，但恶意攻击可以包含JavaScript等内容，使其看起来需要用户提交密码、社会保险号、信用卡信息或其他敏感信息。
这里不是要讨论如何构造有效的钓鱼攻击，而是想说明参数的内容可以被操纵，而且会产生意想不到的效果。
6.3 绕过客户端验证 如果程序的输入验证只采用了基于JavaScript的方法，那么这个程序远没有想象中的安全。
客户端的JavaScript基本上是都可以被绕过的。
一些个人代理、个人防火墙以及Cookie管理软件都吹嘘它们可以过滤掉网站弹出窗口和其他入侵组件。
其实很多计算机专家也完全关闭JavaScript，以防止受到最新E-Mail病毒的攻击。
总而言之，有很多的理由和简单的方法让Internet用户禁用JavaScript。
当然，禁用JavaScript会使很多应用程序不完整。
不过现在有很多的工具可以移除JavaScript或允许我们在JavaScript检查后提交内容。
使用一个像Paros的代理，我们就可以在GET或POST请求发送到服务器之前将其截获，以这种方式可在浏览器中输入能通过验证的数据，而在代理中将数据改变为任意值。
6.4 常见的输入验证攻击 让我们看一下常见的输入验证攻击。
虽然大多数攻击只是向程序倾倒垃圾数据，但也有一些攻击含有精心构造的字符串。
在本章中，我们只是展示一下可能被攻击的漏洞，而把具体的攻击细节放在其他章节中讲解。
例如，SQL注入攻击也是输入验证攻击，但我们会在第8章中再详细讨论。
6.4.1 缓冲区溢出 缓冲区溢出很少会在解释性或高级编程语言中产生。
比如，使用PHP或Java语言编写的程序，产生漏洞的可能性就很小。
但还是有可能发生的，比如使用了某种语言的自带函数，而这个函数本身就含有缓冲区溢出漏洞。
不过，还是建议把精力放在其他输入验证，Session管理和其他的Web安全方面。
当然，如果程序包含了自己编写的IIS的ISAPI接口或Apache模块，那么还是需要进行缓冲区溢出的检查，或者进行更高效的代码安全评审。
实现缓冲区溢出攻击，只需要在一个输入字段中输入尽量多的数据，这是一种非常暴力和野蛮的攻击，但如果程序返回了一个错误，那么攻击就起作用了。
Perl非常适合做这种类型攻击，只需要一条指令就可以对一个参数产生任意长度的攻击。
$ perl -e print "a" x 500 aaaaaaa...重复500次 可以通过Perl脚本产生HTTP请求（使用LWP模块），或者使用Netcat产生输出。
当然不是用它们来提交正常的参数，而是用把参数替换成嵌入的Perl语句的输出。
比如一个正常的请求如下： $ echo –e “ GET /login.php?user=faustus\nHTTP/1.0\n\n ” |\ nc –vv Website 80 而在命令行中调用Perl进行缓冲区测试的例子如下： $ echo –e “ GET /login.php?user=\ > `perl –e ‘ print “ a ” x 500 ’ `\nHTTP/1.0\n\n ” |\ nc –vv Website 80 这里向login.php发送了长达500个"a "的user值。
这条语句可在任何类UNIX的环境（或Cygwin）下执行，而和curl程序结合起来使用，还可以减少SSL所带来的问题，如： $ curl https://Website/login.php?user=`perl –e " print" a" x 500"` 当使用不同的负载和不同的长度进行缓冲区溢出测试时，目标程序会返回不同的错误。
这些错误可能全部是“ 密码错误 ”，但也可能有一些会提示user参数的边界情况。
缓冲区测试的首要原则是使用基本的差异分析或异常检测，其步骤如下： 1. 向应用程序发送一个正常的请求并记录服务器的响应； 2. 向应用程序发送第一个缓冲区溢出测试数据，记录服务器此时的响应； 3. 向应用程序发送下一个测试数据并记录服务器响应； 4. 重复第3步。
当发现服务器的响应不同于对一个“ 正常 ”请求的响应时，就检查有哪些差别，这将帮助你跟踪到那些能产生错误的特定载荷（比如在URL中，7809个斜杠是可以接受的，但7810个就不可接受）。
在某些情况下，缓冲区溢出攻击能够使攻击者在服务器上执行任意指令。
编写这样的缓冲区溢出利用工具比较困难，但使用起来则非常简单。

换句话说，发现漏洞需要非常有经验的安全审核人员，但初级的攻击者也能够下载到和使用预先编写好的攻击工具。

注意　大多数时候，缓冲区溢出攻击都是“盲目”的，如果没有权限对应用附加调试器，或者查看日志或系统信息，构造出可执行任意命令的缓冲区溢出代码将是件非常困难的工作。

比如，IIS上的FrontPage Services Extension溢出漏洞，如果没有对系统完全的访问权限，就不能被利用而构成缓冲区溢出攻击。

<<黑客大曝光>>

编辑推荐

在网络技术和电子商务飞速发展的今天，Web应用安全面临着前所未有的挑战。
本书凝聚了作者们超过30年的Web安全从业经验，让读者了解黑客如何使用架构和应用剖析来进行探测和进入有漏洞的系统。
详细剖析了Web应用的安全漏洞，攻击手法和对抗措施，一步步的教授如何防御邪恶的攻击，并协助读者理解黑客的思考过程。
经典黑客图书，知名行业专家打造！
站在恶意入侵的角度来审视你的Web应用，以防范最新的Web攻击。
本书经过全面的修订和更新以覆盖最新Web利用技术，为你一步步地展示黑客们如何瞄准漏洞站点、获取权限、窃取关键数据以及进行毁灭性的攻击。
通过本书你将：　　了解黑客如何使用架构和应用剖析来进行探测和进入有漏洞的系统。
获得最流行的Web平台（包括IIS，Apache，PHP和ASP.NET）的漏洞利用代码、绕过技术和对抗措施的细节。
学习常见Web认证机制（包括基于密码的，多因素的和诸如Passport的单点登录机制）的优势和弱点。
理解如何通过高级会话分析、劫持和定制技术来操纵任何Web应用的访问控制的核心。
寻找和定位输入验证缺陷，包括跨站脚本（XSS）、SQL注入、HTTPB向应头截断、编码和特殊字符滥用。
进一步了解最新的SQL注入技术，包括盲注入攻击、子查询高级利用技术、Oracle利用技术和改进的对抗措施。
了解最新的XML Web服务攻击、Web管理攻击和DDos攻击（包括点击欺诈）。
遍历Firefox和IE的漏洞利用代码，以及最新的社会工程驱动的客户端攻击，诸如钓鱼和广告软件。
本书帮你参透Web攻击背后的玄机，练就一双大眼金睛，为你的Web应用保驾护航！

版权说明