Web 网站安全技术分析

An Analysis of Website Security Technology

符凤平　(贵州省安顺市气象局 贵州 安顺 561000)

摘　要：本文从 Web 网站安全现状出发,分别从程序设计、Web 服务器、数据库等三方面对网站安全技术进行了较为详细的分析,并有针对性地提出相应的安全防范措施。主要介绍以下安全技术:SQL 注入攻击、md5 算法、Web 服务器操作系统的配置与管理、Access 数据库和 SQL Server 2000 数据库的安全管理等。

关键词：SQL 注入 md5 算法 Session 对象 Web 服务器 数据库管理

计算机网络技术的发展,给人类带来了极大的便利,但随之而来的是一系列安全问题,这些安全问题使得 Web 网站的开发与维护变得日趋复杂,网络安全管理面临着极大的技术挑战。加强网络安全技术尤其是网站安全技术方面的研究与应用,已成为当前计算机安全技术领域的研究重点。

1　引言

Web 网站一般采用 IIS 服务与数据库技术,常存在以下安全隐患:①操作系统未进行合理的配置与管理;②SQL 注入式攻击对网站可能造成的破坏,如数据库被入侵,重要信息被获取;③IIS 服务、服务器端脚本运行环境(如 ASP)本身存在的一些安全漏洞等等。为了排除上述系列安全隐患,在网站设计过程中,必须谨慎地编写程序代码,多方面地采取安全措施。本文主要从程序设计、Web 服务器、数据库等三方面对网站安全技术作了较为详细的分析,并有针对性地提出相应的防范措施。

2　Web 网站安全现状

计算机网络技术的迅猛发展,特别是 Internet 网的普及、Web 站点的增加,信息交互与共享已遍及整个世界。在这种互联性和开放性给社会带来极大效益的同时,网站安全问题愈来愈突出。这是因为:①计算机犯罪手段不断提高;②网站交互性成为安全的致命弱点,各种颇受用户欢迎的功能如聊天室、电子商务、E-mail 等,正是被攻击的主要对象;③自动化攻击工具在网络上大量泛滥,攻击技术不断得到普及与提高;④病毒泛滥带来的潜在危害。由此可见,网站安全面临着极大的挑战,主要安全问题可以分为以下几类:①服务器信息(如口令、密匙等)被破译,导致服务器被入侵;②浏览器功能强大,不仅为用户也为黑客的攻击提供了方便;③网站上的文件被非法访问,对文件的隐私性、机密性和完整性造成较大的威胁;④Web 服务器中的程序缺陷,成为黑客的攻击目标。综上所述,Web 网站的安全问题已成为当前计算机安全技术领域的主要研究方向,也正是本文讨论的重点。

3　Web 网站安全技术分析

3.1　程序设计

3.1.1　SQL 注入攻击

SQL 作为一种国际标准的数据库查询语言,在各种开发环境中得到了广泛的应用。SQL 注入原理,就是在客户端提交特殊代码,非法获取服务器信息。攻击者把破坏性的 SQL 命令输入到 Web 页面的表单域,导致 Web 服务器执行恶意的 SQL 命令。常见 SQL 注入攻击有以下两种情况:

(1) 用户登录漏洞。编写程序时,未对用户输入信息的合法性进行判断,这样用户就可以提交一段恶意性代码,以获得一些敏感的信息,或者控制整个服务器。通常使用以下 SQL 语句进行用户密码验证:

Sql = " Select * from 表名 where name = '" username" 'and pwd = '" password" '"

其中 username 和 password 为用户名和密码变

量。通过分析发现,当用户名输入字符串:asx'or'4 =4,密码输入:lxx,替换变量后,该 SQL 语句变为:Sql = " Select * from 表名 where name = 'asx'or'4 =4'and pwd ='lxx'。执行时,遇到或(or)操作就会忽略下面的与(and)操作,而逻辑表达式 4 =4 的值为 true,所以 SQL 语句会忽略后面的逻辑判断而通过密码验证。为避免用户登录漏洞,在执行验证之前,必须对用户名和密码进行合法性判断。若脚本语言为 VBScript,可使用字符串替换函数 replace( )进行处理。

(2) 用户身份验证被绕过的漏洞。对于需要通过身份验证后才能被访问的页面,如果攻击者知道了这些页面的路径和文件名,就可以绕过身份验证,直接进入到该页面。比如需要用户通过 login. asp 页面登录并经过身份验证才能打开 zyfw. asp 页面,攻击者可以通过 http://www. * * * /zyfw. asp 直接进入该页面。为防范此漏洞,可利用 Session 对象来实现安全控制。当访问者通过身份验证页面后,就把 Session 对象的 Sessionid 属性作为一个 Session 变量存储起来,当访问者试图登录到有效链接页面时,可将当前的 Sessionid 与存储在 Session 对象中的 ID 进行比较,如果不匹配,则拒绝访问。如在 Session("id")中保存着第一次链接的 Sessionid,则可使用以下语句判断用户能否访问:<% if session. sessionid < > session("id") then response. end% >。

3.1.2 Cookie 的安全性

为防止非法用户访问合法用户的会话变量,服务器为每个 Sessionid 指派一个随机生成号码。每当用户的 Web 浏览器返回一个 Sessionid Cookie 时,服务器取出 Sessionid 被赋予的数字,检查与存储在服务器上的生成号码是否一致,如果不一致则不允许用户访问会话变量。同时,应加密重要的 Sessionid Cookie,一旦黑客截获了用户的 Sessionid Cookie,就能假冒该用户开启一个活动会话。

3.1.3 页面缓存管理

如果浏览器设置了"浏览网页时首先查看本地缓冲区里的页面",就给非法用户提供了越权浏览的机会。因此,重要的 Web 页面(如身份验证页面)必须禁止页面缓存,强制浏览器每次向 Web 服务器请求新页面。Asp 环境下利用 Response 对象的 Expires 属性和 Clear 方法可实现禁止页面缓存,具体设置为:

```
<%
  Response. expires =0
  Rsponse. clear
% >
```

其中 expires 表示缓存页面的有效期,0 表示立即过期,clear 表示清空缓冲区。

3.1.4 使用 md5 算法对用户信息进行加密

md5 的全称是 message – digest algorithm 5(信息 – 摘要算法),经 md2、md3 和 md4 发展而来。它是一种加密算法,对任何文件能产生一个长度为 128 位的验证码,广泛用于数据完整性检查和数据签名。md5 具备以下特性:(1)不可逆性。从变换后的 md5 码无法获知原文件信息。(2)高度的离散性。原文件内容的细微变化就会导致其产生的 md5 验证码不同,而且 md5 码的产生不可预测。(3)代码唯一性。由于 md5 码长度为 128 位,具有相同 md5 码的可能性非常低,而且一旦原文件内容被损坏或被修改的话,其 md5 码就会发生很大的变化。在 ASP 环境中使用 md5 算法举例如下:

```
<%
  password = request. form(" password" )
  u_password = md5(password)
% >
```

其中 password = request. form(" password")使用 request 方法从 form 表单里获得用户提交的密码,u_password = md5(password)将用 md5 算法加密后的密码传递给 u_password,如将以上这段代码存为文件 md5. asp,则直接使用以下语句调用 md5. asp 文件:<! – – #include file = " md5. asp" – – >。

3.1.5 用户在线状态检测

如果用户已经断开连接或停止下载,就不用再浪费服务器的资源创建网页,因为缓冲区内容将被 IIS 服务丢弃。对需要大量时间计算或资源使用较多的网页来说,有必要在每一阶段都检查用户是否已离线,这样可以节省 Web 服务器资源,提高其运行效率。在 ASP 中,可使用 Response. IsClientConnected 属性进行检测,具体代码为:

```
<%
  If Response. IsClientConnected Then
    Response. Flush
```

```
    Else
      Response. End
    End If
  % >
```

3.1.6 设置服务器连接等待时间

对服务器操作活动的增长,大量增加建立数据库连接的时间,会加重服务器的负担,过长的连接延时将降低数据库的性能。在 ASP 中,用 Connection 对象的 ConnectionTimeout,可以限制放弃连接尝试并发出错误消息之前应用程序等待的时间。例如,以下语句利用 ConnectionTimeout 属性实现在取消连接尝试之前等待 20 秒:

```
  <%
    Set cn = Server. CreateObject ( " ADODB. Connection" )
    cn. ConnectionTimeout =20
    cn. Open " FILEDSN = MyDatabase. dsn"
  % >
```

默认的 ConnectionTimeout 属性是 30 秒。需要注意的是,在将 ConnectionTimeout 属性应用到数据库程序前,一定要确保数据源支持该属性。

3.2 Web 服务器

对操作系统进行合理的配置与管理,营造一个安全的操作系统环境,Web 网站才能稳定可靠地运行。对操作系统的配置与管理,主要有以下几个方面:

3.2.1 操作系统安装注意事项

目前,服务器采用的操作系统大多是 Windows 2000 Server。在安装过程中,须注意以下事项:

(1) 安装组件的选择。根据安全原则:最少的服务 + 最小的权限 = 最大的安全,应尽量减少安装需要的服务,如果开启某个服务,就要预防该服务可能引起的安全隐患。默认安装的几个服务如"Indexing Service"、"FrontPage 2000 Server Extensions"、"Internet Service Manager",存在着极大的安全隐患,应将其禁用。但有的服务却不能禁用,如关闭"Windows Installer"服务,系统将无法安装新的应用程序;关闭"Telephony"服务,远程用户将无法拨入服务器。

(2) 及时安装系统各种补丁。系统补丁最好在所有应用软件安装完之后再安装,因为补丁程序往往要替换或修改某些系统文件。开启 Windows Update 自动更新功能,及时从网上下载各种补丁。安装补丁程序是 Web 网站安全防范中必不可少的一项工作。

3.2.2 使用 NTFS 格式文件系统,加强访问权限管理

NTFS 格式文件系统可以设置文件目录访问权限,比 FAT32 格式文件系统更安全。缺省情况下,使用 NTFS 文件系统的硬盘分区,名称为 everyone 的用户具有完全控制权限。为了防止可能的非法入侵,必须合理地设置目录和文件的访问权限,仅给用户真正需要的权限,权限的最小化原则是系统安全的重要保障。除此之外,应删除默认帐号,将系统管理员(administrator)及时更名,修改其密码并定期进行更换,从而避免非法用户的攻击。

3.2.3 禁用共享

Windows 2000 server 每次重新启动后,会自动将 c $、d $ 等硬盘分区、ipc $、admin $ 设置为默认共享,这些共享对操作系统的安全存在着较大的威胁。禁用共享有多种方法,第一种方法是利用操作系统的管理工具进行设置,但在操作系统重新启动后,默认共享会恢复。另一种方法是建立一个批处理文件(.bat),将其放入启动组,这样每次启动操作系统后就禁用了默认共享。批处理文件内容举例如下:

```
net share c $ /delete
net share d $ /delete
net share admin $ /delete
net share ipc $ /delete
```

除了上述两种方法外,还可以通过修改注册表方式禁用默认共享,这里就不再详述。

3.2.4 使用审核策略,备份日志文件,关闭部分 TCP/IP 端口

审核策略不仅可以监视系统中各种与安全有关的事件,还会生成安全日志。通过分析安全日志,可以发现并阻止各种危及系统安全的行为。Windows 2000 Server 默认安装下,安全审核是关闭的。要进行审核,必须先确定审核策略,指定要审核的安全事件的类别。具体的设置:在"管理工具 – 本地安全策略 – 本地策略 – 审核策略"中打开必要的审核。除了安全日志,系统日志和应用程序日志也是很好的监视工具,它们记录了用户自登录开始直到退出的整个操作过程,通过查看和跟踪备份日志文件,可以了解系统活动情况,防止非法用户入侵,为网络安全分析提供可靠的依据。另

外,关闭部分不用的 TCP/IP 端口(如系统提供 www 服务,仅开启 80 端口; 提供 FTP 服务,仅开启 21 端口),以避免通过端口攻击为系统带来的安全隐患。

3.2.5 访问限制

(1) IP 地址限制。通过对 IIS 服务进行设置,可以实现 IP 地址访问限制。具体设置为:启动 IIS 服务,在 Web 站点的属性中选择"目录安全性"一栏,打开"IP 地址及域名限制",添加指定的 IP 地址即可。(2) 用户访问控制。IIS 服务提供了对站点资源进行匿名访问与验证控制设置,Web 服务器根据设置对用户的身份进行验证,阻止未授权用户与受限制内容建立 Http 连接。具体设置时,须对 Web 站点的"目录安全性"属性页进行编辑。另外,通过设置防火墙,也可实现部分访问限制功能。

3.3 数据库

数据库是网站的最核心部分,它的安全运行是保证整个网站正常运行的前提。常用的网站数据库是 Access 和 SQL Server,以下分别针对它们在 ASP 运行环境下的安全技术进行分析。

3.3.1 Access 数据库的安全

(1) 数据库文件名和存放路径应复杂。为防止数据库被入侵,可为 Access 数据库文件起一个复杂的名字,改变其存放路径。一般情况下,Access 数据库文件存放在 Web 目录中,很多黑客就是利用这种规律来查找并下载数据库文件,因此采用更改数据库文件存储路径的方法,将数据库文件存放在 Web 站点目录以外的某个文件夹中,可以有效提高 Access 数据库的安全性。

(2) 利用 ODBC 数据源。设计网站时,一般是将 Access 数据库文件的存储路径和文件名存放在数据库连接文件中,但如果连接文件的内容外泄,那么不管数据库文件名多么复杂,都存在着极大的安全隐患,这时就可以使用 ODBC 数据源方法,即使连接文件的内容外泄,他人也只能知道网站程序所使用的 ODBC 数据源名称,而数据库文件的存储路径和文件名却无法找到。手工修改数据库连接文件(如 conn. asp)中的内容,可以创建 ODBC 数据源的连接,如首先修改 conn. asp 文件:

```
<%
    DBPath = Server. MapPath(". /data/autostation. mdb")
    conn. Open" driver = {Microsoft Access Driver (*. mdb)};dbq = " &DBPath
%>
```

将以上内容修改为: conn. open " auto",其中"auto"是指 ODBC 数据源名称。接着在管理器中新建名为"auto"的 ODBC 数据源,并在其中指定"autostation. mdb"数据库文件的位置即可。

(3) 为数据库文件编码及加密。为防止他人使用其它工具查看数据库文件,可以对数据库文件进行编码,具体做法是选择"工具→安全→编码/解码数据库"。为数据库设置密码的具体做法:以"独占"的方式打开数据库,在功能表中选择"工具→安全→设置数据库密码"。但是 Access 的加密机制比较简单,还须借助于其它安全措施(如身份认证和权限控制)来保证数据库的安全。

3.3.2 SQL Server 数据库的安全

(1) 及时安装 SQL Server 数据库系统最新升级包,加强数据库访问日志的监视。为了提高数据库安全性,须定期对 SQL Server 数据库软件进行升级,安装已发布的安全更新。定期审核数据库的登录事件,在实例属性中选择"安全性",将其中的审核级别选定为全部,这样在数据库系统日志里就详细记录了所有帐号的登录事件,一旦出现问题能够查出原因,及时补救。

(2) 控制访问权限。定义用户和角色对数据库、数据表和数据列的访问权限,限制用户对表拥有直接的查询、更改、插入、删除权限,可以通过给用户访问视图和执行存储过程的权限,以保证数据库的安全。为 sa 分配一个复杂的密码,取消 guest 帐号。sa 具有对 SQL Server 数据库操作的全部权限,但在安装 SQL Server 时 sa 缺省口令为空,为 SQL Server 带来了潜在的安全隐患,应把 sa 的口令换为更安全的口令,同时不能把 sa 帐号的密码写在应用程序或者脚本中。

(3) 隔离数据库服务器,并定期备份。物理和逻辑上的隔离组成了 SQL Server 的安全基础,安装数据库的机器应该处于一个从物理形式上受到保护的地方。数据库应该安装在单位内部网的安全区域中,不直接与 Internet 网相连。同时,定期对数据库进行备份,并将备份保存在安全的地方,在必要的时候能够实现对数据库的恢复。

(下转第 131 页)

(上接第 165 页)

4 结语

本文主要从程序设计、服务器、数据库等三方面对 Web 网站安全技术进行了比较详细地分析。随着新的安全问题不断出现,构建一个面面俱到的网站安全体系,仍需要不断地学习与实践。网站的安全稳定运行,应侧重于预防,不断增强安全意识,及时地堵上各种安全漏洞,采取各种预防措施,才能及时有效地排除安全隐患。

参考文献

1 姬武军,牛光. 网站平台安全防入侵问题解决方法与实现. 电脑应用技术,2007,69:24 - 27.

2 张人意. Web 网站系统的安全性研究[硕士学位论文],长沙:湖南师范大学,2005.

3 卜英奇. 网站安全技术的分析与应用[硕士学位论文],吉林:吉林大学,2007.