**1、Apache样例文件泄漏**

**漏洞描述**

apache一些样例文件没有删除，可能存在cookie、session伪造，进行后台登录操作

**修复建议**

1、删除样例文件

2、对apache中web.xml进行相关设置

**2、弱口令**

**漏洞描述**

由于系统中存在有弱口令，导致攻击者通过弱口令可轻松登录系统中，从而进行下一步的攻击，如上传webshell，获取敏感数据！

另外攻击者利用弱口令登录网站管理后台，可任意增删改等操作，从而造成负面影响！

**修复建议**

1、建议强制用户首次登录时修改默认口令，或是使用用户自定义初始密码的策略；

2、 完善密码策略，信息安全最佳实践的密码策略为8位（包括）以上字符，包含数字、大小写字母、特殊字符中的至少3种。

3、对管理后台进行访问控制，修改后台弱口令，加强口令强度并定期修改。

4、增加验证机制，防爆破机制，限制ip＋cookie访问次数。

**3、明文传输登录口令**

**漏洞描述**

用户登录过程中使用明文传输用户登录信息，若用户遭受中间人攻击时，攻击者可直接获取该用户登录账户，从而进行进一步渗透。

**修复建议**

1、用户登录信息使用加密传输，如密码在传输前使用安全的算法加密后传输，可采用的算法包括：不可逆hash算法加盐（4位及以上随机数，由服务器端产生）；安全对称加密算法，如AES(128、192、256位)，且必须保证客户端密钥安全，不可被破解或读出；非对称加密算法，如RSA(不低于1024位)、SM2等。

2、使用https来保证传输的安全。

**4、暴力破解**

**漏洞描述**

由于没有对登录页面进行相关的防暴力破解机制，如无验证码、有验证码但验证码未在服务器端校验以及无登录错误次数限制等，导致攻击者可通过暴力破解获取用户登录账户及口令，从而获取网站登录访问权限！

**修复建议**

1、添加验证码机制，加入图片（验证码动态生成且满足随机性）或者短信验证码（验证码具备超时时限一般为1分钟，且在该时限内错误次数超过3次则进行锁定1分钟后方能重新获取验证码，超时后验证码自动失效）！

2、验证码必须在服务器端进行校验，客户端的一切校验都是不安全的！

**5、sql注入漏洞**

**漏洞描述**

Web程序代码中对于用户提交的参数未做过滤就直接放到SQL语句中执行，导致参数中的特殊字符打破了SQL语句原有逻辑，黑客可以利用该漏洞执行任意SQL语句，如查询数据、下载数据、写入webshell、执行系统命令以及绕过登录限制等。

**修复建议**

代码层最佳防御sql漏洞方案：采用sql语句预编译和绑定变量，是防御sql注入的最佳方法。

1）所有的查询语句都使用数据库提供的参数化查询接口，参数化的语句使用参数而不是将用户输入变量嵌入到SQL语句中。当前几乎所有的数据库系统都提供了参数化SQL语句执行接口，使用此接口可以非常有效的防止SQL注入攻击。

2）对进入数据库的特殊字符（’”<>&\*;等）进行转义处理，或编码转换。

3）确认每种数据的类型，比如数字型的数据就必须是数字，数据库中的存储字段必须对应为int型。

4）数据长度应该严格规定，能在一定程度上防止比较长的SQL注入语句无法正确执行。

5）网站每个数据层的编码统一，建议全部使用UTF-8编码，上下层编码不一致有可能导致一些过滤模型被绕过。

6）严格限制网站用户的数据库的操作权限，给此用户提供仅仅能够满足其工作的权限，从而最大限度的减少注入攻击对数据库的危害。

7）避免网站显示SQL错误信息，比如类型错误、字段不匹配等，防止攻击者利用这些错误信息进行一些判断。

**6、跨站脚本攻击（xss）漏洞**

**漏洞描述**

1）、Web程序代码中把用户提交的参数未做过滤或过了不严就直接输出到页面，参数中的特殊字符打破了HTML页面的原有逻辑，黑客可以利用该漏洞执行恶意HTML/JS代码、构造蠕虫传播、篡改页面实施钓鱼攻击、诱以及导用户再次登录，然后获取其登录凭证等。

2）、XSS攻击对Web服务器本身虽无直接危害，但是它借助网站进行传播，对网站用户进行攻击，窃取网站用户账号信息等，从而也会对网站产生较严重的危害。XSS攻击可导致以下危害：

3）、钓鱼欺骗：最典型的就是利用目标网站的反射型跨站脚本漏洞将目标网站重定向到钓鱼网站，或者通过注入钓鱼JavaScript脚本以监控目标网站的表单输入，甚至攻击者基于DHTML技术发起更高级的钓鱼攻击。

4）、网站挂马：跨站时，攻击者利用Iframe标签嵌入隐藏的恶意网站，将被攻击者定向到恶意网站上、或弹出恶意网站窗口等方式，进行挂马攻击。

5）、身份盗用：Cookie是用户对于特定网站的身份验证标志，XSS攻击可以盗取用户的cookie，从而利用该cookie盗取用户对该网站的操作权限。如果一个网站管理员用户的cookie被窃取，将会对网站引发巨大的危害。

6）、盗取网站用户信息：当窃取到用户cookie从而获取到用户身份时，攻击者可以盗取到用户对网站的操作权限，从而查看用户隐私信息。

7）、垃圾信息发送：在社交网站社区中，利用XSS漏洞借用被攻击者的身份发送大量的垃圾信息给特定的目标群。

8）、劫持用户Web行为：一些高级的XSS攻击甚至可以劫持用户的Web行为，从而监视用户的浏览历史、发送与接收的数据等等。

9）、XSS蠕虫：借助XSS蠕虫病毒还可以用来打广告、刷流量、挂马、恶作剧、破坏网上数据、实施DDoS攻击等。

**修复建议**

xss漏洞本质上是一种html注入，也就是将html代码注入到网页中。那么其防御的根本就是在将用户提交的代码显示到页面上时做好一系列的过滤与转义

1）假定所有输入都是可疑的，必须对所有输入中的script、iframe等字样进行严格的检查。这里的输入不仅仅是用户可以直接交互的输入接口，也包括HTTP请求中的Cookie中的变量，HTTP请求头部中的变量等。

2）不仅要验证数据的类型，还要验证其格式、长度、范围和内容。

3）不要仅仅在客户端做数据的验证与过滤，关键的过滤步骤在服务端进行。

4）对输出的数据也要检查，数据库里的值有可能会在一个大网站的多处都有输出，即使在输入做了编码等操作，在各处的输出点时也要进行安全检查。

**7、目标服务器启用了不安全HTTP方法**

**漏洞描述**

目标服务器启用了不安全的传输方法，如PUT、TRACE、DELETE、MOVE等，这些方法表示可能在服务器上使用了 WebDAV，由于dav方法允许客户端操纵服务器上的文件，如上传、修改、删除相关文件等危险操作，如果没有合理配置dav，有可能允许未授权的用户对其进行利用，修改服务器上的文件。

**修复建议**

1）、关闭不安全的传输方法，推荐只使用POST、GET方法！

2）、如果服务器不需要支持 WebDAV，请务必禁用它。

或者为允许webdav的目录配置严格的访问权限，如认证方法，认证需要的用户名，密码。

**8、任意文件上传**

**漏洞描述**

文件上传漏洞通常由于网页代码中的文件上传路径变量过滤不严或webserver相关解析漏洞未修复而造成的，如果文件上传功能实现代码没有严格限制用户上传的文件后缀以及文件类型，攻击者可通过 Web 访问的目录上传任意文件，包括网站后门文件（webshell），进而远程控制网站服务器。

攻击者可通过此漏洞上传恶意脚本文件，对服务器的正常运行造成安全威胁！

**修复建议**

1）、对上传文件类型进行限制，并且不能只做前端的限制，而要前端和后端一起限制，后端可以进行扩展名检测，重命名文件，MIME类型检测以及限制上传文件的大小，或是将上传的文件放在安全的路径下，尽量放于webserver之外的远程服务器等。

2）、严格限制和校验上传的文件，禁止上传恶意代码的文件。同时限制相关目录的执行权限，防范webshell攻击。

3）、对上传文件格式进行严格校验及安全扫描，防止上传恶意脚本文件；

4）、设置权限限制，禁止上传目录的执行权限；

5）、严格限制可上传的文件类型；

6）、严格限制上传的文件路径。

7）、文件扩展名服务端白名单校验。

8）、文件内容服务端校验。

9）、上传文件重命名。

10）、隐藏上传文件路径。

**9、测试页面泄漏在外网**

**漏洞描述**

一些测试页面泄漏到外网，导致外界误传公司被黑客入侵，影响公司声誉。

**修复建议**

删除测试页面以及无用文件，例如test.cgi，phpinfo.php，info.pho， .svn/entries等。

**10、目录浏览**

**漏洞描述**

由于服务器端配置不当，开启了目录浏览，黑客可获得服务器上的文件目录结构，从而下载敏感文件。

**修复建议**

通过修改配置文件，去除中间件（如IIS、apache、tomcat）的文件目录索引功能

2.设置目录权限

3.在每个目录下创建一个空的index.html页面。

**11、phpinfo信息泄漏**

**漏洞描述**

Web站点的某些测试页面可能会使用到PHP的phpinfo()函数，会输出服务器的关键信息，从而造成信息泄露，通过获取的信息可进行下一步的攻击计划！

**修复建议**

删除该PHP文件！

**12、未授权访问**

**漏洞描述**

由于没有对相关敏感页面进行访问权限的检查，导致攻击者可未授权访问，从而获取敏感信息及进行未授权操作等！

**修复建议**

对相关页面进行严格的访问权限的控制以及对访问角色进行权限检查！

**13、越权访问**

**漏洞描述**

由于没有对用户访问角色的权限进行严格的检查及限制，导致当前账号可对其他账号进行相关操作，如查看、修改等！

**修复建议**

对用户访问角色的权限进行严格的检查及限制！

**14、命令执行漏洞**

**漏洞描述**

命令执行漏洞是指代码未对用户可控参数做过滤，导致直接带入执行命令的代码中，对恶意构造的语句，可被用来执行任意命令。黑客可在服务器上执行任意命令，写入后门，从而入侵服务器，获取服务器的管理员权限，危害巨大。

**修复建议**

严格过滤用户输入的数据，禁止执行非预期系统命令！

**15、应用程序错误信息泄露**

漏洞描述

黑客可通过特殊的攻击向量，使web服务器出现500、404等相关错误，导致信息泄漏如绝对路径、webserver版本、源代码、sql语句等敏感信息，恶意攻击者很有可能利用这些信息实施进一步的攻击。

修复建议

1、自定义错误页面或归一化错误页面信息提示！

2、修正代码！

**16、LDAP注入**

**漏洞描述**

由于Web 应用程序没有对用户提供的输入进行适当过滤和检查，攻击者便有可能修改LDAP 语句的结构，并且以（例如：数据库服务器、Web 应用程序服务器、Web 服务器）的权限执行任意命令，许可权可能会允许查询、修改或除去 LDAP 树状构造内任何数据。

**修复建议**

对用户的输入进行严格的过滤及检查，并且对类型也进行检查！

**17、文件包含漏洞**

**漏洞描述**

本地文件包含是指程序代码在处理包含文件的时候没有严格控制。利用这个漏洞，攻击者可以先把上传的静态文件，或网站日志文件作为代码执行，或者包含远程服务器上的恶意文件，进而获取到服务器权限。

**修复建议**

1、严格检查变量是否已经初始化。

2、对所有输入提交可能包含的文件地址，包括服务器本地文件及远程文件，进行严格的检查，参数中不允许出现../之类的目录跳转符。

3、严格检查include类的文件包含函数中的参数是否外界可控。

4、不要仅仅在客户端做数据的验证与过滤，关键的过滤步骤在服务端进行。

**18、网站敏感压缩文件泄露**

**漏洞描述**

误将网站备份文件或是敏感信息文件存放在某个网站目录下，外部黑客可通过暴力破解文件名等方法下载该备份文件，导致网站敏感信息泄露。

**修复建议**

1、不要在网站目录下存放网站备份文件或敏感信息的文件。

2、如需存放该类文件，请将文件名命名为难以猜解的字符串。

**19、CRLF HTTP 头部注入漏洞**

**漏洞描述**

CRLF 是“回车 +换行”（\r\n）的简称。在 HTTP 协议中，HTTPHeader 与 HTTP Body 是用两个 CRLF 符号进行分隔的，**浏览器**根据这两个 CRLF 符号来获取 HTTP 内容并显示。因此，一旦攻击者能够控制 HTTP 消息头中的字符，注入一些恶意的换行，就能注入一些会话 Cookie 或者 HTML 代码。

**修复建议**

过滤 \r 、\n 之类的换行符，避免输入的数据污染到其他 HTTP 消息头。

**20、URL 跳转漏洞**

**漏洞描述**

Web 程序直接跳转到参数中的 URL ，或页面引入任意的开发者 URL，被攻击者利用可实施钓鱼攻击等操作。

**修复建议**

在控制页面转向的地方校验传入的URL是否为可信域名。

**21、Crossdomain.xml 配置不当**

**漏洞描述**

网站根目录下的 crossdomain.xml 文件指明了远程Flash 是否可以加载当前网站的资源（图片、网页内容、Flash等）。如果配置不当，可能导致遭受跨站请求伪造（CSRF）攻击。

**修复建议**

对于不需要从外部加载资源的网站，在 crossdomain.xml 文件中更改allow-access-from的domain属性为域名白名单。

**22、敏感信息泄露**

**漏洞描述**

在页面中或者返回的响应包中泄露了敏感信息，通过这些信息，攻击者可进一步渗透。

**修复建议**

1、如果是探针或测试页面等无用的程序建议删除，或者修改不易被猜到的名字。

2、禁用泄露敏感信息的页面或应用。

3、对相关敏感信息进行模糊化处理，在服务器端进行！

4、对服务器端返回的数据进行严格的检查，满足查询数据与页面显示数据一致，切勿返回多于的数据！

**23、任意文件下载**

**漏洞描述**

文件下载处由于未对下载路径进行过滤，利用路径回溯符../跳出程序本身的限制目录实现来下载任意文件，如下载系统密码文件等！

**修复建议**

对下载路径进行过滤，如下载前对传入的参数进行过滤，并且对下载文件类型进行检查，是否是允许下载的类型，另外禁止使用回溯符../！

**24、weblogic SSRF服务器请求伪造**

**漏洞描述**

目标存在weblogic SSRF服务器请求伪造漏洞。WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。SSRF(Server-Side Request Forgery:服务器端请求伪造) 是一种由攻击者构造形成由服务端发起请求的一个安全漏洞。一般情况下，SSRF攻击的目标是从外网无法访问的内部系统。（正是因为它是由服务端发起的，所以它能够请求到与它相连而与外网隔离的内部系统）。Weblogic中间件默认带有“UDDI 目录浏览器”且为未授权访问，通过该应用，可进行无回显的SSRF请求。攻击者可利用该漏洞对企业内网进行大规模扫描，了解内网结构，并可能结合内网漏洞直接获取服务器权限。

**修复建议**

1.删除uddiexplorer文件夹

2.限制uddiexplorer应用只能内网访问

**25、目标站点存在网马**

**漏洞描述**

经渗透测试发现目标站点存在有webshell，攻击者可进行远程连接操作，进行恶意操作！

**修复建议**

1）删除可疑文件，并进行本地文件漏洞扫描排查是否还存在有其他木马！

2）使用相关手段发现并及时修复已存在的漏洞！

**26、IIS短文件名泄露漏洞**

**漏洞描述**

Internet Information Services（IIS，互联网信息服务）是由微软公司提供的基于运行Microsoft windows的互联网基本服务。Microsoft IIS在实现上存在文件枚举漏洞，攻击者可利用此漏洞枚举网络服务器根目录中的文件。

危害：攻击者可以利用“~”字符猜解或遍历服务器中的文件名，或对IIS服务器中的.Net Framework进行拒绝服务攻击。

黑客可通过该漏洞尝试获取网站服务器下存放文件的文件名，达到获取更多信息来入侵服务器的目的。

**修复建议**

修改Windows配置，关闭短文件名功能。

1.关闭NTFS 8.3文件格式的支持。该功能默认是开启的，对于大多数用户来说无需开启。

2.如果是虚拟主机空间用户,可采用以下修复方案：

1）修改注册列表HKLM\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisable8dot3NameCreation的值为1(此修改只能禁止NTFS8.3格式文件名创建,已经存在的文件的短文件名无法移除)。

2）如果你的web环境不需要asp.net的支持你可以进入Internet 信息服务(IIS)管理器 --- Web 服务扩展 - ASP.NET 选择禁止此功能。

3）升级net framework 至4.0以上版本。

3.将web文件夹的内容拷贝到另一个位置，比如D:\www到D:\www.back，然后删除原文件夹D:\www，再重命名D:\www.back到D:\www。如果不重新复制，已经存在的短文件名则是不会消失的。

**27、Apache Struts2 远程代码执行漏洞（S2-019）**

**漏洞描述**

Apache Struts2的“Dynamic MethodInvocation”机制是默认开启的，仅提醒用户如果可能的情况下关闭此机制，如果未关闭此机制将导致远程代码执行漏洞，远程攻击者可利用此漏洞在受影响应用上下文中执行任意代码。

**修复建议**

1、目前厂商已经发布了升级补丁以修复这个安全问题，请到厂商的主页下载！

2、或者手工设置struts.xml文件<constantname="struts.enable.DynamicMethodInvocation"value="false"/>

**28、Apache Struts2 远程代码执行漏洞（S2-037）**

**漏洞描述**

Apache Struts2在使用REST插件时，攻击者可以绕过动态方法执行的限制，调用恶意表达式执行远程代码。

**修复建议**

建议用户到官方获取最新补丁或者最新版本程序！

**29、Apache Struts2 DevMode 远程代码执行漏洞**

**漏洞描述**

为了便于开发人员调试程序，Struts2提供了一个devMode模式，可以方便查看程序错误以及日志等信息。当Struts2中的devMode模式设置为true时，存在严重远程代码执行漏洞。如果WebService 启动权限为最高权限时，可远程执行任意命令，包括关机、建立新用户、以及删除服务器上所有文件等等。

**修复建议**

建议用户到官方获取最新补丁或者最新版本程序！

或者将struts.properties中的devMode设置为false，或是在struts.xml中添加如下代码：<constant name="struts.devMode"value="false"/>。

**30、Apache Struts2 远程代码执行漏洞（S2-045）**

**漏洞描述**

Apache Struts2的Jakarta Multipartparser插件存在远程代码执行漏洞，漏洞编号为CVE-2017-5638。攻击者可以在使用该插件上传文件时，修改HTTP请求头中的Content-Type值来触发该漏洞，导致远程执行代码。

**修复建议**

检测方式查看web目录下/WEB-INF/lib/目录下的struts-core.x.x.jar ，如果这个版本在Struts2.3.5 到 Struts2.3.31 以及 Struts2.5 到 Struts2.5.10之间则存在漏洞！

1、建议用户到官方获取最新补丁或者最新版本程序！

2、更新至Strusts2.3.32或者Strusts2.5.10.1，或使用第三方的防护设备进行防护。

3、临时解决方案：删除commons-fileupload-x.x.x.jar文件（会造成上传功能不可用）。

4、修改WEB-INF/classes目录下的配置

在WEB-INF/classes目录下的struts.xml中的struts 标签下添加

<constantname=”struts.custom.i18n.resources”value=”global”/>；

在WEB-INF/classes/目录下添加global.properties，文件内容如下:

struts.messages.upload.error.InvalidContentTypeException=1

**31、Apache Struts2 远程代码执行漏洞（S2-033）**

**漏洞描述**

Apache Struts2在开启动态方法调用（DynamicMethod Invocation）的情况下，攻击者使用REST插件调用恶意表达式可以远程执行代码。

**修复建议**

1、用户到官方获取最新补丁或者最新版本程序！

2、或者在允许的情况下禁用动态方法调用（Dynamic Method Invocation），修改Struts2的配置文件struts.xml，将struts.enable.DynamicMethodInvocation设置为“false”。

**32、目标URL存在httphost头攻击漏洞**

**漏洞描述**

为了方便的获得网站域名，开发人员一般依赖于HTTP Host header，但是这个header是不可信赖的，如果应用程序没有对host header值进行处理，就有可能造成恶意代码的传入。

**修复建议**

web应用程序应该使用SERVER\_NAME而不是host header。

在Apache和Nginx里可以通过设置一个虚拟机来记录所有的非法host header。在Nginx里还可以通过指定一个SERVER\_NAME名单，Apache也可以通过指定一个SERVER\_NAME名单并开启UseCanonicalName选项。

**33、登录绕过漏洞**

**漏洞描述**

由于对登录的账号及口令校验存在逻辑缺陷，或再次使用服务器端返回的相关参数作为最终登录凭证，导致可绕过登录限制，如服务器返回一个flag参数作为登录是否成功的标准，但是由于代码最后登录是否成功是通过获取这个flag参数来作为最终的验证，导致攻击者通过修改flag参数即可绕过登录的限制！

**修复建议**

修改验证逻辑，如是否登录成功服务器端返回一个参数，但是到此就是最终验证，不需要再对返回的参数进行使用并作为登录是否成功的最终判断依据！

**34、短信/邮件轰炸**

**漏洞描述**

由于没有对短信或者邮件发送次数进行限制，导致可无限次发送短信或邮件给用户，从而造成短信轰炸，进而可能被大量用户投诉，从而影响公司声誉！

**修复建议**

对发送短信或邮件的次数进行限制，如1分钟只能发送1次短信或邮件，并且需要在服务器进行限制！

**35、slow http Dos拒绝服务**

**漏洞描述**

按照设计，HTTP协议要求服务器在处理之前完全接收请求。如果HTTP请求没有完成，或者传输速率非常低，服务器会保持其资源忙于等待其余数据。如果服务器保持太多的资源忙，这将造成一个拒绝服务。严重者一台主机即可让web运行缓慢甚至是崩溃！

**修复建议**

对于 Apache 可以做以下优化：

设置合适的 timeout 时间（Apache 已默认启用了 reqtimeout 模块），规定了 Header 发送的时间以及频率和 Body 发送的时间以及频率

增大 MaxClients(MaxRequestWorkers)：增加最大的连接数。根据官方文档，两个参数是一回事，版本不同，MaxRequestWorkers was called MaxClients before version 2.3.13.Theold name is still supported.

默认安装的 Apache 存在 Slow Attack 的威胁，原因就是虽然设置的 timeoute，但是最大连接数不够，如果攻击的请求频率足够大，仍然会占满Apache的所有连接。

**36、web服务测试页面信息泄露**

**漏洞描述**

由于没有删除默认的且与业务无关的页面，导致信息泄露，如：webserver版本信息、中间件类型及版本信息等，通过对这类信息的收集，攻击者可制定具有针对性的攻击计划！

**修复建议**

删除与业务无关的页面，如果是必须需要使用的中间件管理页面，建议对该页面进行访问权限的控制！