目录

SQL注入攻击

SQL注入就是通过把SQL命令插入到Web表单递交或输入域名或页面请求的查询字符串，最终达到欺骗服务器执行恶意的SQL命令。它是利用现有应用程序，将（恶意）的SQL命令注入到后台数据库引擎执行的能力，它可以通过在Web表单中输入（恶意）SQL语句得到一个存在安全漏洞的网站上的数据库，而不是按照设计者意图去执行SQL语句。

SQL注入攻击可以非法读取、篡改、添加、删除数据库中的数据，盗取用户各类敏感信息，获取利益，可以修改数据库来改变网页的内容，可以私自添加或删除账号。通常攻击者有以下攻击技巧。

 错误回显

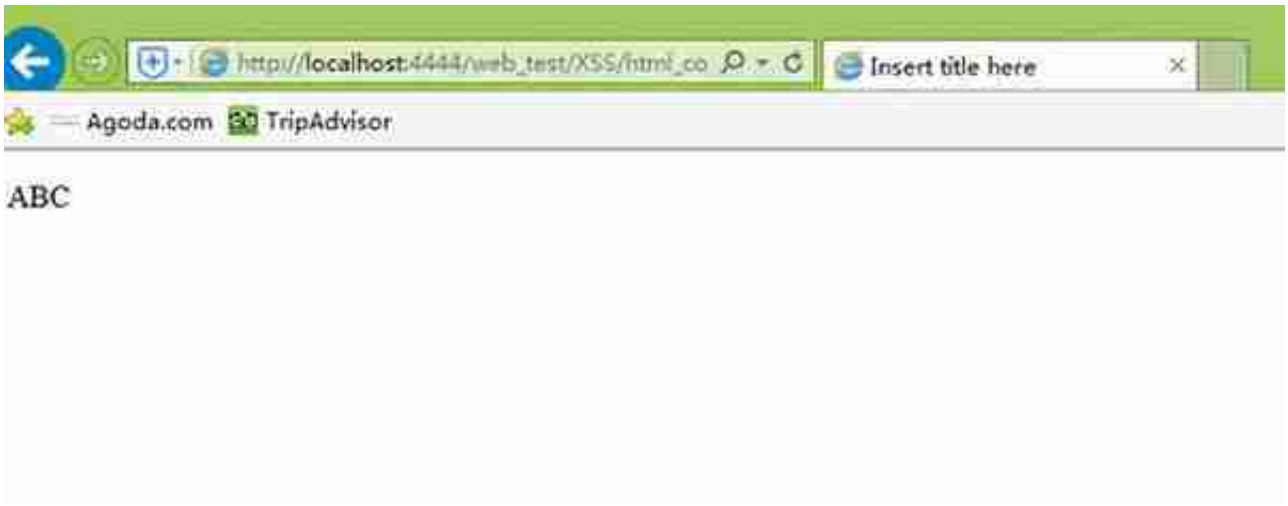
 基于布尔的盲注

 基于时间盲注

 联合查询注入

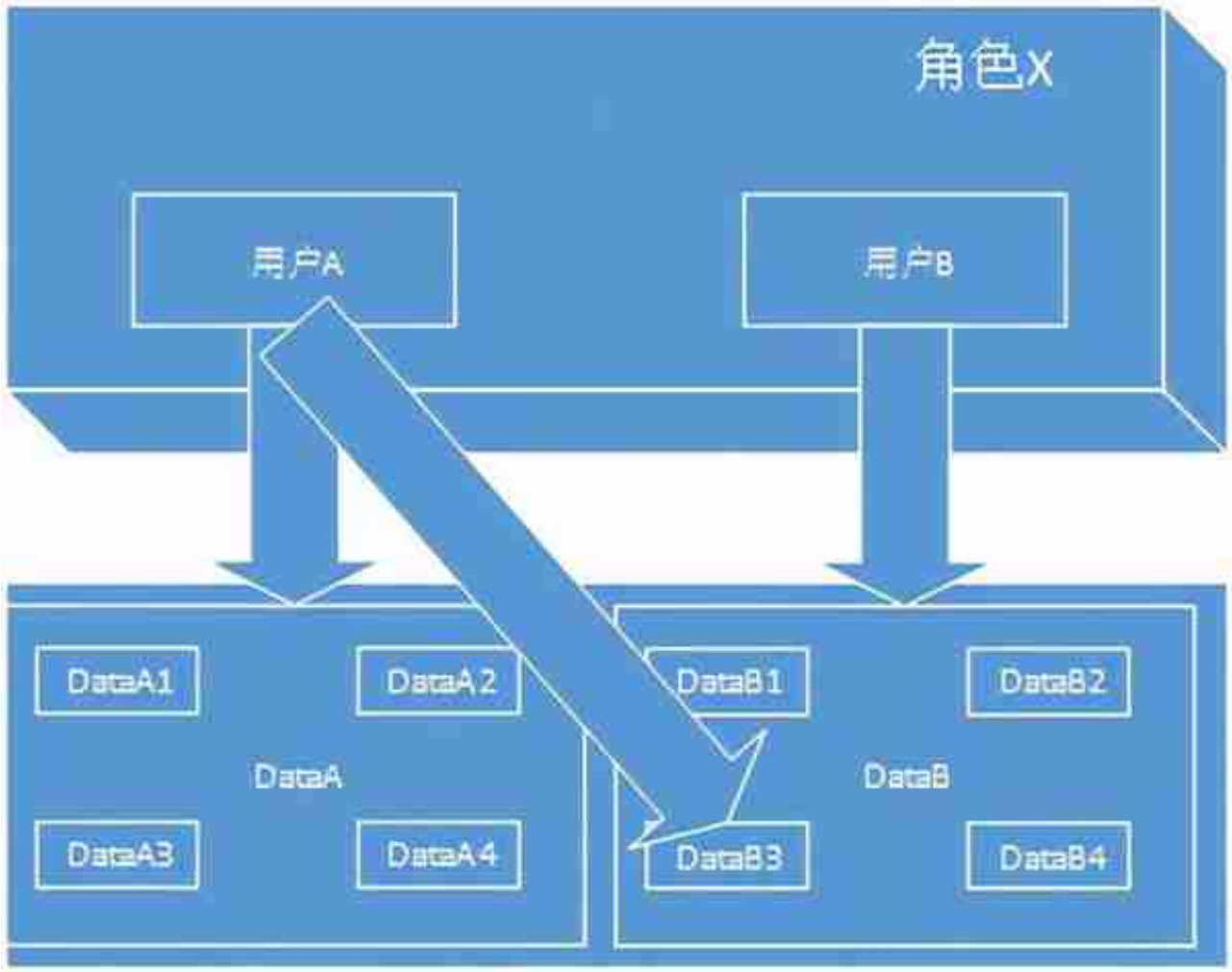
SQL注入攻击-攻击技巧之错误回显

如果攻击者在请求参数中输入的参数导致后台执行查询语句的语法错误，从服务器中直接返回了错误信息。则表明该请求有sql注入漏洞，并且可以在错误回显中获取服务器的数据库类型。



如果用户把输入的name参数改为一个js脚本，则A网站则会执行该脚本：

http://localhost:4444/web_test/XSS/html_content/b.jsp?name=<script>alert(1)</script>



失效的访问控制-水平越权

水平权限漏洞对应的防御措施如下。

 在逻辑层做鉴权，检查请求的操作者（从session中获取）和操作的对象（请求参数中获取）是否具有关联关系（查询数据库），如果无关联关系，则停止操作。这种防御的缺点使增加了一次数据库查询操作。

 把权限的控制转移到数据接口层中，即在进行sql查询更新操作时，在sql语句后加入userId的条件，而userId的值是从session中获取。

 在请求的参数中增加一个token值，如MD5（url+key），用做完整性校验，当攻击者修改参数时，拦截器将会拦截请求，从而完成防御。

安全配置错误

安全配置错误可以发生在一个应用程序堆栈的任何层面，包括平台、Web服务器、应用服务器、数据库、框架和自定义代码。开发人员和系统管理员需共同努力，以确保整个堆栈的正确配置。自动扫描器可用于检测未安装的补丁、错误的配置、默认帐户的使用、不必要的服务等。

- 服务器系统维护的安全配置
- 应用程序的安全配置

安全配置错误-服务器系统维护的安全配置

对于服务器系统维护方面的安全配置，主要有以下几个方面。

- 安装补丁程序
- 账号和密码保护
- 监测系统日志
- 使用HTTPS请求
- 隐藏IP地址
- 关闭不需要的服务和端口
- 部署检测系统和防御系统
- 定期对服务器进行备份

安全配置错误-应用程序的安全配置

关于应用程序的安全配置，主要是应用程序在部署到生产环境后的一些配置,主要有以下几个方面。

- 修改或删除后台默认账号
- 禁用服务器上的目录列表

 隐藏错误堆栈信息

敏感信息泄露

在这个领域最常见的漏洞是应该加密的数据不进行加密。在使用加密的情况下，常见的问题是不安全的密钥生成和管理和使用弱算法是很普遍的，特别是使用弱的哈希算法来保护密码。以下是对敏感信息的分类。

 个人信息，如姓名，身份证ID，电话号码，银行账户，驾驶证号码，社保卡号，护照号码等都是敏感数据。

 网站登录的用户名、密码，SSL证书，会话ID，加密使用的密钥等都属于敏感信息，这些信息一旦泄露，攻击者就可以以合法用户的身份访问Web系统，随意进行各种攻击操作。

 Web服务器的OS类型，版本信息，Web容器的名称，版本号，数据库类型，版本号，应用软件使用开源软件信息都属于敏感信息，因为攻击者知道这些软件信息，就会利用这些软件存在的公开漏洞进行专门攻击，提升了系统被攻破的可能性。

敏感信息泄露-保护敏感信息

对于数据信息需要保证数据安全的三要素：完整性、机密性和可用性。

 针对个人数据，必须加密存储，且要使用安全的加密算法。

 针对敏感数据传输，需要采用SSL加密通道，对每一个请求都应该使用SSL加密通道。

 应用程序运行出错容易造成敏感信息的泄露，定制统一出错页面，杜绝显示此类敏感信息到Web客户端。

 保存敏感信息的文件要严格控制访问权限。

攻击检测与防范不足

对攻击的防范措施所能覆盖的攻击类型，不应该只是XSS和SQL注入。
你可以通过如 WAFs, RASP, 和OWASP AppSensor
等技术来检测并阻止攻击。攻击保护有以下三个目标。

 检测攻击

有没有发生合法用户不可能产生的情况？ 应用程序是否以普通用户永远不会做的方式运行（例如，请求频率太高，非典型输入，异常使用模式，重复请求）？

 对攻击的响应

日志和通知对及时响应至关重要。考虑是否自动阻止请求，并确定阻止的IP地址或IP段。 考虑禁用或监控不良行为的用户账户。

 快速修复

如果开发或运维团队无法在一天内推出关键修补程序，需要部署一个可以分析HTTP流量，数据流或代码执行的虚拟补丁，并防止漏洞被利用。

攻击检测与防范不足-DDOS攻击方式

DDoS即分布式拒绝服务攻击， 是 指借助于客户/服务器技术，将多个计算机联合起来作为攻击平台，对一个或多个目标发动DOS攻击，从而成倍地提高拒绝服务攻击的威力。

 SYN Flood

SYN Flood (SYN洪水) 是种典型的DoS (Denial of Service，拒绝服务) 攻击。效果就是服务器TCP连接资源耗尽，停止响应正常的TCP连接请求。修改内核参数即可有效缓解,分别为启用SYN
Cookie、设置SYN最大队列长度以及设置SYN+ACK最大重试次数。

 UDP Flood

攻击者利用简单的TCP/IP服务，如Chargen和Echo来传送毫无用处的占满带宽的数据。通过伪造与某一主机的Chargen服务之间的一次的UDP连接，回复地址指向开着Echo服务的一台主机，这样就生成在两台主机之间存在很多的无用数据流，这些无用数据流就会导致带宽的服务攻击

防范配置方法：关闭Chargen服务。

 land 攻击

land 攻击是一种使用相同的源和目的主机和端口发送数据包到某台机器的攻击。结果通常使存在漏洞的机器崩溃。

 Smurf攻击

Smurf攻击是以最初发动这种攻击的程序名“Smurf”来命名的。这种攻击方法结合使用了IP欺骗和ICMP回复方法使大量网络传输充斥目标系统，引起目标系统拒绝为正常系统进行服务，对网络进行监控和统计发现, 若出现Smurf 攻击，则会出现大量的echo 报文。由于存在echo 应答风暴， 此时，echo 报文在所有报文中所占的比例大大增加。所以，如出现这种情况，就可能遭到了Smurf
攻击，挫败一个Smurf攻击的最简单的方法就是对边界路由器的回音应答（echo reply）信息包进行过滤，然后丢弃他们，使网络避免被湮没。

 利用服务漏洞

攻击检测与防范不足-DDOS检测方式

关于DDOS的检测方式主要有以下几种。

 基于流量的检测

基于流量大小进行检测的方法就是在被保护的网络边界路由器上部署流量检测算法，根据流量的突发检测DoS工具的发生

 基于源IP地址的检测

就是在被保护网络的边界路由器上部署源IP检测算法，根据源IP个数的突然增加来判断Dos的发生

 基于包属性的检测

绝服务攻击发生时，攻击数据包破坏了正常网络状况下进出数据包在IP数据包头字段的统计学稳定性，因此采用一定算法在正常情况下进行包属性字段学习判断进出数据包的危险度

攻击检测与防范不足-DDOS防范

关于DDOS的防御措施主要有以下几种。

 源端防御

增加路由器安全性能配置，建议在该网段的路由器上做配置调整，做到数据包过滤、反欺骗、异常识别、协议分析、流量限制多验证体系

 配置防火墙过滤规则

设计访问规则，阻止和杜绝一些恶意信息对主机的攻击

 加强终端防御，对受害者主机、受害者网络进行防御

提高主机系统和网络系统安全性，加强入口防御

 加强中端防御，攻击性的数据包在传输过程中采取的防御措施

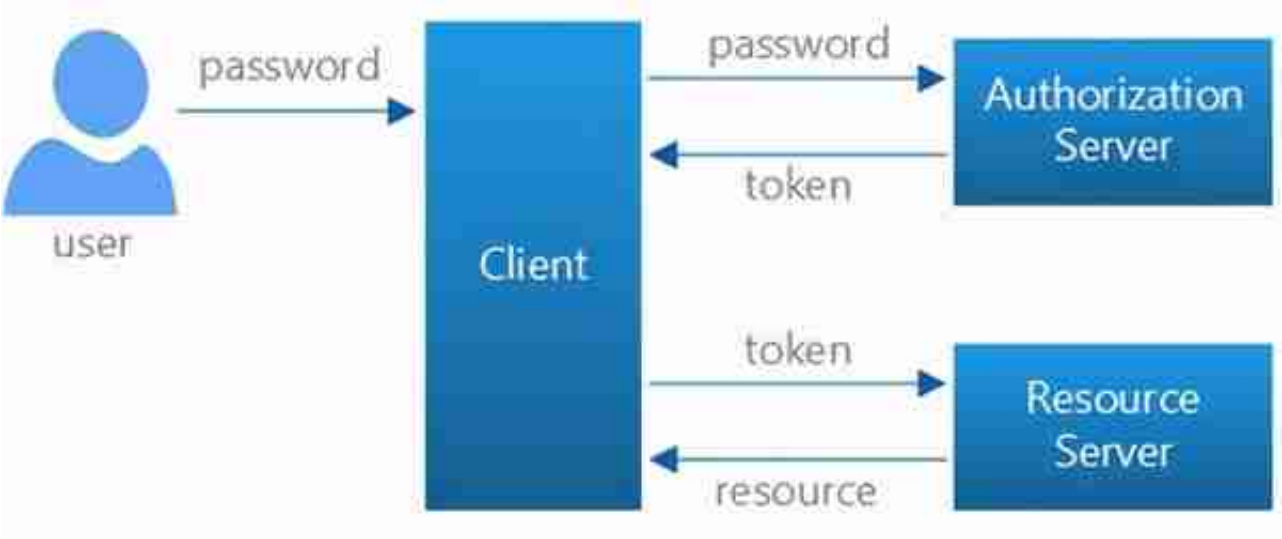
跨站请求伪造

跨站请求伪造（CSRF）攻击可以以你的名义发送恶意请求。CSRF则通过伪装来自受信任用户的请求来利用受信任的网站。

CSRF能够做的事情包括：以你名义发送邮件，发消息，盗取你的账号，甚至于购买商品，虚拟货币转账，造成的问题包括：个人隐私泄露以及财产安全。

跨站请求伪造-攻击原理

攻击原理如下图所示：



文件上传漏洞

文件上传漏洞是指用户上传了一个可执行的脚本文件，并通过此脚本文件获取了执行服务器端命令的能力。

文件上传漏洞有以下危害。

 上传文件是Web脚本语言，服务器的Web容器解释并执行了用户上传的脚本，导致代码执行。

 上传文件是病毒、木马文件，黑客用以诱骗用户或者管理员下载执行。

 上传文件是钓鱼图片或为包含了脚本的图片，在某些版本的浏览器中会被作为脚本执行，被用户钓鱼和欺诈。

文件上传漏洞-解析漏洞

攻击者在利用上传漏洞时，通常会与Web容器的解析漏洞配合在一起。所以我们了解一下解析漏洞。

 IIS解析漏洞

当建立*.asa、*.asp格式的文件夹时，其目录下的任意文件都将被IIS当做asp文件来解析。

当文件名为*.asp;1.jpg时，IIS同样会以ASP脚本来执行。

 Apache解析漏洞

在Apache 1.x和2.x中存在解析漏洞：当碰到不认识的扩展名是，将会从后向前解析，直到碰到认识的扩展名为止，如果都不认识，则会暴露其源代码。比如：a.php.aa.xa.bb，Apache首先会解析bb扩展名，如果不认识，将会解析xa扩展名，直到遍历到认识的扩展名为止，然后再将其进行解析。

文件上传漏洞-防御

对于文件上传漏洞的防御，有以下措施。

 文件上传的目录设置为不可执行

在实际应用中，很多大型网站的上传应用，文件上传后会放到独立的存储上，做静态文件处理。

 判断文件类型或对文件进行处理

在判断文件类型时，可以结合使用MIME Type、后缀检查等方式。此外，对于图片的处理，可以使用压缩函数或者resize函数，在处理图片的同时破坏图片中可能包含的HTML代码。

 使用随机数改写文件名和文件路径

文件上传如果要执行代码，则需要用户能够访问到这个文件。如果应用使用随机数改写了文件名和路径，将极大地增加攻击的成本。与此同时，像a.jsp;b.jpg或者a.php.aa.bb这种文件，都将会因为文件名被改写而无法进行攻击。