

# WaterExp模板（WaterExport [水报告]）

Author:**linshao**

-- 内含42种常见**网站漏洞**，让你**水得快速，水得专业 --**

注：碰瓷类型漏洞脚本WaterExp.Py请关注公众号”黑客start”回复：”碰瓷“ 进行获取



## 碰瓷类型



1. Nginx版本泄露

|  |  |
| --- | --- |
| 风险名称 | Nginx版本泄露 |
| 风险级别 | **低风险** |
| 风险描述 | 暴露出版本号 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 暴露出来的版本号容易变成攻击者可利用的信息。从安全的角度来说，隐藏版本号会相对安全些！ |
| 加固建议 | 打开nginx配置文件nginx.conf，在http {...}里加上  server\_tokens off; |

1. tomcat默认报错页面

|  |  |
| --- | --- |
| 风险名称 | Nginx版本泄露 |
| 风险级别 | **低风险** |
| 风险描述 | 暴露出中间件，具体版本号 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 暴露出来的信息容易变成攻击者可利用的信息，如服务器中间件特性，支持的脚本语言等等。从安全的角度来说，隐藏版本号会相对安全些！ |
| 加固建议 | 打开nginx配置文件nginx.conf，在http {...}里加上server\_tokens off;  自定义错误页面 |

1. Iis版本泄露

|  |  |
| --- | --- |
| 风险名称 | Iis版本泄露 |
| 风险级别 | **低风险** |
| 风险描述 | 暴露出中间件，具体版本号 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 暴露出来的信息容易变成攻击者可利用的信息，如服务器中间件特性，支持的脚本语言等等。以及中间件头泄露可能会被资产测绘系统分类收录，在中间件出现漏洞的时候会优先受到攻击。从安全的角度来说，隐藏版本号较相对安全！ |
| 加固建议 | 修改iis配置隐藏版本号 |

1. jquery版本泄露

|  |  |
| --- | --- |
| 风险名称 | jquery版本泄露 |
| 风险级别 | **低风险** |
| 风险描述 | 泄露jQuery版本后黑客可根据jQuery当前版本进行查找相关版本漏洞从而进行利用 |
| 测试过程 |  |
| 影响地址 | [https://www.xxxxx.com.cn/app\_js/jquery.min.js](https://www.cib-fund.com.cn/app_js/jquery.min.js) |
| 风险分析 | 泄露jQuery版本后黑客可根据jQuery当前版本进行查找相关版本漏洞从而进行利用 |
| 加固建议 | 修改jQuery版本号 |

1. X-Powered-By信息泄露

|  |  |
| --- | --- |
| 风险名称 | X-Powered-By信息泄露 |
| 风险级别 | **低风险** |
| 风险描述 | 返回的响应头信息中暴露了具体的容器版本，攻击者可针对中间件的特性进行利用 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec/ |
| 加固建议 | 修改配置文件，取消响应包的X-Powered-By头字段 |

1. X-Frame-Options响应头丢失

|  |  |
| --- | --- |
| 风险名称 | 点击劫持：X-Frame-Options响应头丢失 |
| 风险级别 | **中风险** |
| 风险描述 | 返回的响应头信息中没有包含x-frame-options头信息设置，点击劫持(ClickJacking)允许攻击者使用一个透明的iframe，覆盖在一个网页上，然后诱使用户在该页面上进行操作，此时用户将在不知情的情况下点击透明的iframe页面； |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 配置nginx发送X-Frame-Options响应头，把下面这行添加到’http’,‘server’或者’location’的配置中:  add\_header X-Frame-Options SAMEORIGIN; |

1. Content-Security-Policy头缺失

|  |  |
| --- | --- |
| 风险名称 | 缺少HTTP标头”Content-Security-Policy”(网页安全政策) |
| 风险级别 | **低风险** |
| 风险描述 | 内容安全策略没有通过元标记或标头声明，因此可以利用浏览器对从服务器接收到的内容的信任。恶意脚本由受害者的浏览器执行，因为浏览器信任内容的来源，即使不是原始地点的。 |
| 测试过程 |  |
| 风险分析 | csp也是一种XSS解决方案，它可以通过配置来防止从外部加载资源，拒绝执行js等等，但错误的配置可能会引起网站功能异常 |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 可通过中间件容器配置添加响应包头  可在页面内直接添加meta标记 |

1. TRACE方法启用

|  |  |
| --- | --- |
| 风险名称 | TRACE方法启用 |
| 风险级别 | **中风险** |
| 风险描述 | TRACE方法是HTTP（超文本传输）协议定义的一种协议调试方法，该方法使得服务器原样返回任何客户端请求的内容（可能会附加路由中间的代理服务器的信息），由于该方法原样返回客户端提交的任意数据，因此，可用来进行跨站脚本（XSS）攻击，这种攻击方式又称为跨站跟踪攻击（XST）。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 即使网站对关键页面启用了HttpOnly头标记,攻击者也可以通过TRACE方法绕过进行XSS攻击，盗取会话cookie、获取账户、模拟其他用户身份，甚至可以修改网页呈现给其他用户的内容 |
| 加固建议 | 关闭不安全的传输方法，推荐只使用POST、GET方法！ |

1. 使用有漏洞的组件

|  |  |
| --- | --- |
| 风险名称 | 使用有漏洞的组件 |
| 风险级别 | **低风险** |
| 风险描述 | jQuery v1.12.4存在DOM型XSS |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 升级jQuery为新版本  或者修改字段1.12.4为3.52.4迷惑攻击者和误导扫描器识别。 |

1. 缓慢的HTTP拒绝服务攻击

|  |  |
| --- | --- |
| 风险名称 | 缓慢的HTTP拒绝服务攻击 |
| 风险级别 | **中风险** |
| 风险描述 | 由于HTTP请求底层使用TCP网络连接进行会话，因此如果中间件对会话超时时间设置不合理，并且HTTP在发送请求的时候采用慢速发HTTP请求，就会导致占用一个HTTP连接会话。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 如果发送大量慢速的HTTP包就会导致拒绝服务攻击DoS。 |
| 加固建议 | 限制HTTP头部传输的最大许可时间。超过指定时间HTTP Header还没有传输完成，直接判定源IP地址为慢速连接攻击，中断连接并加入黑名单。 |

1. 目标服务器启用了不安全HTTP方法

|  |  |
| --- | --- |
| 风险名称 | 目标服务器启用了不安全HTTP方法 |
| 风险级别 | **低风险** |
| 风险描述 | 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。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 有利于攻击者搜集服务器信息，实施下一步攻击 |
| 加固建议 | 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。 |

1. 缺少secure标识

|  |  |
| --- | --- |
| 风险名称 | 会话cookie中缺少Secure标识 |
| 风险级别 | **低风险** |
| 风险描述 | 会话cookie中缺少Secure标识会导致攻击者可以通过非HTTPS页面窃取到用户的cookie信息，造成用户cookie信息的泄露。设置附加的secure标识来提示浏览器只能通过Https(加密方式)方式来传输cookie，Http(未加密方式)方式则不可以。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 配置nginx配置文件，使请求头cookie后面带上secure字段 |

1. 敏感数据GET传输

|  |  |
| --- | --- |
| 风险名称 | 敏感数据GET传输 |
| 风险级别 | **低风险** |
| 风险描述 | GET方式传递的敏感数据更容易被截获，降低了攻击的门槛，攻击者通过简单的JavaScript脚本即可获得敏感数据的内容。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 使用POST方式传递敏感信息。 |

1. SSL证书无效

|  |  |
| --- | --- |
| 风险名称 | SSL证书无效 |
| 风险级别 | **低风险** |
| 风险描述 | SSL证书不受信任，SSL证书不是由受信的CA机构所签发的。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 攻击者可利用自签名SSL证书伪造钓鱼网站上，让用户信息泄露。  浏览器会弹出警告，易遭受攻击，让用户体验度大大降低。  自签名SSL证书是没有可访问的吊销列表的，所以它不具备让浏览器实时查验证书的状态，一旦证书丢失或者被盗而无法吊销，就很有可能被用于非法用途从而让用户蒙受损失。同时，浏览器还会发出“吊销列表不可用，是否继续？”的警告，不仅降低了网页的浏览速度，还大大降低了访问者对网站的信任度。 |
| 加固建议 | 进行证书的正确安装或者从可信机构申请证书 |

## 业务类型



1. 弱口令

|  |  |
| --- | --- |
| 风险名称 | 弱口令 |
| 风险级别 | **高风险** |
| 风险描述 | 由于企业管理员安全意识不强，为了图方便、好记使用弱口令，随着网络攻击的自动化、傻瓜化，弱口令攻击已成为一种最快速有效的攻击手段 |
| 漏洞截图 |  |
| 影响地址 | <http://10.80.90.46:8080/taompdata/a> |
| 加固建议 | 改为强口令 |

1. 用户无限注册

|  |  |
| --- | --- |
| 风险名称 | 用户无限注册 |
| 风险级别 | **中风险** |
| 风险描述 | 用户可无限提交请求注册用户，扰乱正常的业务 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 修改name字段放包即可注册成功。 |
| 加固建议 | 添加验证码。 |

1. 登录可爆破

|  |  |
| --- | --- |
| 风险名称 | 登录可爆破 |
| 风险级别 | **中风险** |
| 风险描述 | 无验证码以及传参加密，可直接爆破 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 使用burp加载字典可直接爆破用户密码。对用户admin爆破成功。 |
| 加固建议 | 添加验证码。  或者登录失败锁定账户30分钟 |

1. 用户无限注册

|  |  |
| --- | --- |
| 风险名称 | 用户无限注册 |
| 风险级别 | **中风险** |
| 风险描述 | 用户可无限提交请求注册用户，扰乱正常的业务 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 修改name字段放包即可注册成功。 |
| 加固建议 | 添加验证码。 |

1. 短信无限重放

|  |  |
| --- | --- |
| 风险名称 | 短信重放攻击 |
| 风险级别 | **高风险** |
| 风险描述 | 手机号验证时删除验证参数就可无限重发 |
| 测试过程 | 2694e399c60419870fc927675f495f2 |
| 影响地址 | https://github.com/linshaosec |
| 攻击载荷 | GET/xxx&mobile=手机号 HTTP/1.1  Host: xxx.xxxx.com.cn |
| 风险分析 | 企业对接sms验证码很多情况下是按条收费的，无限发送验证码可直接对企业财产造成损失 |
| 加固建议 | 修改后端代码，添加对验证码 |

1. 验证码不过期

|  |  |
| --- | --- |
| 风险名称 | 无视验证码爆破 |
| 风险级别 | **高风险** |
| 风险描述 | 验证码不失效，可配合明文传输账户密码进行爆破 |
| 测试过程 |  |
| 风险分析 | 后端没有对验证码参数进行过期处理，导致一个验证码反复使用 |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 修改代码，对验证码一旦使用则过期处理 |

1. 测试账号未删除

|  |  |
| --- | --- |
| 风险名称 | 测试账号未删除 |
| 风险级别 | **中风险** |
| 风险描述 | 测试账号 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 测试账户一般为弱密码，随手一试还存在system账户。存在比较大的概率进入系统内部从而进一步进行攻击。 |
| 加固建议 | 如非必要，删除测试账户，或采用强安全性用户名。  另外，统一更改错误提示，无论账户存在与否都显示为“用户名或密码错误”，防止账户枚举 |

1. 账户枚举

|  |  |
| --- | --- |
| 风险名称 | 用户枚举 |
| 风险级别 | **中风险** |
| 风险描述 | 获取用户名，减少爆破难度。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 输入admin用户不存在，输入1admin提示密码错误。 |
| 加固建议 | 对后端jsp代码if语句字符串进行修改。 |

1. 预约信息重放

|  |  |
| --- | --- |
| 风险名称 | 预约信息重放 |
| 风险级别 | **高风险** |
| 风险描述 | 攻击者利用该漏洞无限提交预约留言 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 攻击者利用该漏洞绕过验证码，消耗服务器系统资源占用，扰乱正常的预约业务。 |
| 加固建议 | 修复后端验证码校检代码 |

1. 任意账户密码重置

|  |  |
| --- | --- |
| 风险名称 | 任意账户密码重置 |
| 风险级别 | **高风险** |
| 风险描述 | 可以重置其他人包括管理员的账户密码 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 密码重置不应该使用时间戳来作为验证字段,修改高权限用户密码登录后攻击面会变大，危害也越大 |
| 加固建议 | 使用邮箱发送随机验证码验证来进行密码重置 |

**漏洞类型**



1. 越权访问

|  |  |
| --- | --- |
| 风险名称 | 越权访问 |
| 风险级别 | **高风险** |
| 风险描述 | 普通用户可以访问到不该访问的资源，即为越权访问 |
| 测试过程 |  |
| 影响地址 | https://xxx.xx.com/names.nsf/$users |
| 风险分析 | 用户登录后直接访问上述链接可以看到其他用户的用户名，邮箱及其登录密码散列等等信息，进而可以用john,hashcat等工具破解出明文密码或制作字典等等 |
| 加固建议 | 联系软件官方进行修复  临时建议:禁止访问该页面 |

1. 存储型XSS

|  |  |
| --- | --- |
| 风险名称 | 存储型XSS(跨站请求攻击) |
| 风险级别 | **高风险** |
| 风险描述 | 攻击者利用前端的漏洞，把恶意代码插入到html里面，执行恶意js代码 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 攻击荷载 | POST /xxx.json HTTP/1.1  Host: xxxx.com.cn  Payload:jsonpCallback=123312&email=<input/onauxclick="prompt(/HowAreYou/)">&channel=weixin&timestamp=1624688039869&\_=1624688024479 |
| 风险分析 | 攻击对象为用户或企业管理员，利用构造好的恶意js链接诱导，，执行恶意js代码，比如偷取cookcie令牌，获取经纬度，等操作 |
| 加固建议 | 对theme参数进行强过滤，<>等html标签符号进行实体化转义处理  使用waf等安全设备对请求进行检测 |

1. 文件上传漏洞

|  |  |
| --- | --- |
| 风险名称 | 文件上传漏洞 |
| 风险级别 | **高风险** |
| 风险描述 | 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得了执行服务器端命令的能力。 |
| 测试过程 |  |
| 影响地址 | Xxxxx.com.cn/uploader/upload.action |
| 风险分析 | 攻击者绕过上传机制上传恶意代码并执行从而控制服务器。 |
| 加固建议 | 修改后端代码对上传文件类型使用白名单验证 |

1. csrf跨站请求伪造

|  |  |
| --- | --- |
| 风险名称 | csrf跨站请求伪造 |
| 风险级别 | **中风险** |
| 风险描述 | 跨站请求伪造攻击，是攻击者通过一些技术手段如xss欺骗用户的浏览器去访问一个自己曾经认证过的网站并运行一些操作（如发邮件，发消息，甚至财产操作如转账和购买商品）。由于浏览器曾经认证过，所以被访问的网站会认为是真正的用户操作而去运行。这利用了web中用户身份验证的一个漏洞：简单的身份验证只能保证请求发自某个用户的浏览器，却不能保证请求本身是用户自愿发出的。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 攻击载荷 | GET /account/modifyInfo.json?jsonpCallback=modifyEmailCallback\_1111111111111111&handset=11&channel=111&timestamp=1111&\_=111HTTP/1.1  Host: xxxxx.com.cn |
| 加固建议 | 开发人员对页面添加token验证 |

1. 内网IP地址泄漏

|  |  |
| --- | --- |
| 风险名称 | 内网IP地址泄漏 |
| 风险级别 | **中风险** |
| 风险描述 | 网站的内部IP地址，常常被攻击者通过信息收集，得到其内网的IP地址，对于渗透攻击，打下良好基础，如内网Ip地址段，IP路由等等 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 攻击载荷 | POST /admin/xxxxx/insert HTTP/1.1  Host: xxxxxx.com.com  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0  Accept: application/json, text/javascript, \*/\*; q=0.01  Sec-Fetch-Mode: cors  Sec-Fetch-Site: same-origin  username=admin&password=111 |
| 加固建议 | 隐藏输出报错信息 |

1. SQL注入

|  |  |
| --- | --- |
| 风险名称 | SQL注入 |
| 风险级别 | **高风险** |
| 风险描述 | SQL注入即是指[web应用程序](https://baike.baidu.com/item/web%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/2498090" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)对用户输入数据的合法性没有判断或过滤不严，攻击者可以在web应用程序中事先定义好的查询语句的结尾上添加额外的[SQL语句](https://baike.baidu.com/item/SQL%E8%AF%AD%E5%8F%A5/5714895" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)，在管理员不知情的情况下实现非法操作，以此来实现欺骗[数据库服务器](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E6%9C%8D%E5%8A%A1%E5%99%A8/613818" \t "https://baike.baidu.com/item/sql%E6%B3%A8%E5%85%A5/_blank)执行非授权的任意查询，从而进一步得到相应的数据信息。 |
| 测试过程 |  |
| 影响地址 | https://xx.com/xxx.php?id=73375 |
| 风险分析 | 使用sqlmap直接爆出后台管理员账户密码 |
| 加固建议 | 联系网站开发人员进行修复  使用waf(web应用防火墙)对提交数据进行过滤 |

1. http明文传输

|  |  |
| --- | --- |
| 风险名称 | http明文传输 |
| 风险级别 | **中风险** |
| 风险描述 | http请求未加密，可以通过流量抓取获取到其敏感信息 |
| 测试过程 |  |
| 风险分析 | 请求包中的账户、密码应该加密处理 |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 请求包中的账户、密码应该加密处理  使用Https协议 |

1. 目录遍历

|  |  |
| --- | --- |
| 风险名称 | 目录遍历 |
| 风险级别 | **高风险** |
| 风险描述 | 目录浏览漏洞是由于网站存在配置缺陷，导致网站目录可以被任意浏览，这会导致网站很多隐私文件与目录泄露，比如数据库备份文件、配置文件等，攻击者利用该信息可以为进一步入侵网站做准备。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 寻找对应中间件的配置，关闭对目录浏览的支持。 |

1. springboot内存快照文件泄露

|  |  |
| --- | --- |
| 风险名称 | springboot内存快照文件泄露 |
| 风险级别 | **高风险** |
| 风险描述 | SprinBoot框架不安全的配置可能导致敏感信息泄露 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 攻击者通过下载此文件，可以得知系统详细信息，用户名，以及保存在内存里面的数据，如token，数据库用户名密码等等 |
| 加固建议 | 禁用此sprintboot路由或者设置路径不可访问 |

1. 后台源码泄露

|  |  |
| --- | --- |
| 风险名称 | 后台源码泄露 |
| 风险级别 | **高风险** |
| 风险描述 | 访问admin会将admin文件下载下来，它是一个压缩包打开即可看到脚本源码,源码泄漏导致攻击者阅读源码，更利于其发现及利用漏洞 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 将此文件移动到非web目录的不可访问地址 |

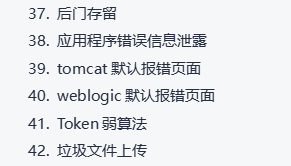
1. 页面源码泄露

|  |  |
| --- | --- |
| 风险名称 | 页面源码泄露 |
| 风险级别 | **中风险** |
| 风险描述 | 访问地址即可得知文件内容，有利于攻击者熟悉网站内部结构，以及获取敏感数据 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 加固建议 | 删除以上不必要文件  启用脚本支持，防止文件泄露 |

1. 25端口邮件系统用户枚举

|  |  |
| --- | --- |
| 风险名称 | 邮件系统用户枚举 |
| 风险级别 | **高风险** |
| 风险描述 | SMTP不安全的配置可能导致敏感信息泄露 |
| 测试过程 |  |
| 影响地址 | 192.168.1.1 |
| 风险分析 | 攻击者可以通过枚举用户名，可以爆破弱口令 |
| 加固建议 | 禁用RCPT TO命令 |

**其他类型**



1. 后门存留

|  |  |
| --- | --- |
| 风险名称 | 存在后门文件 |
| 风险级别 | **高风险** |
| 测试过程 |  |
| 后门位置 | C:/weaver/ecology/cloudstore/ |
| 风险分析 | 护网期间留下的后门文件未删除 |
| 加固建议 | 使用webshell查杀工具进行查杀，对比时间判断文件是否可疑 |

1. 应用程序错误信息泄露

|  |  |
| --- | --- |
| 风险名称 | 应用程序错误信息泄露 |
| 风险级别 | **低风险** |
| 风险描述 | 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，返回服务器敏感信息。 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 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。 |
| 加固建议 | 自定义错误页面或归一化错误页面信息提示！ |

1. tomcat默认报错页面

|  |  |
| --- | --- |
| 风险名称 | tomcat默认报错页面 |
| 风险级别 | **低风险** |
| 风险描述 | 暴露出中间件，具体版本号 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 暴露出来的信息容易变成攻击者可利用的信息，如服务器中间件特性，支持的脚本语言等等。从安全的角度来说，隐藏版本号会相对安全些！ |
| 加固建议 | 打开nginx配置文件nginx.conf，在http {...}里加上server\_tokens off;  自定义错误页面 |

1. weblogic默认报错页面

|  |  |
| --- | --- |
| 风险名称 | weblogic默认报错页面 |
| 风险级别 | **低风险** |
| 风险描述 | 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，返回服务器相关服务的敏感信息。如以下的weblogic，可以尝试使用其相关漏洞来进行利用 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 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。 |
| 加固建议 | 自定义错误页面或归一化错误页面信息提示！ |

1. Token弱算法

|  |  |
| --- | --- |
| 风险名称 | Token弱算法 |
| 风险级别 | **低风险** |
| 风险描述 | Token算法如下，按此算法编写代码对mobile字段爆破，并且生成对应token，即可得知存在用户的手机号码 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 重置密码疑似也使用的此算法，4位验证并不安全，数字验证码确保6位以上 |
| 加固建议 | 修改此加密算法，防止爆破 |

1. 垃圾文件上传

|  |  |
| --- | --- |
| 风险名称 | 垃圾文件上传 |
| 风险级别 | **中风险** |
| 风险描述 | 修改数据包内文件大小限制即可无限重放成功上传 |
| 测试过程 |  |
| 影响地址 | https://github.com/linshaosec |
| 风险分析 | 恶意攻击者可以通过无限上传塞满服务器磁盘，造成应用异常或者服务器宕机 |
| 加固建议 | 无服务端控制上传大小，无需此功能可以剔除掉上传功能 |