# 绿盟科技"网络安全漏洞扫描系统"
# 安全评估报告-站点报表

报 表 生 成 时 间　　2025-04-17　09:02:04

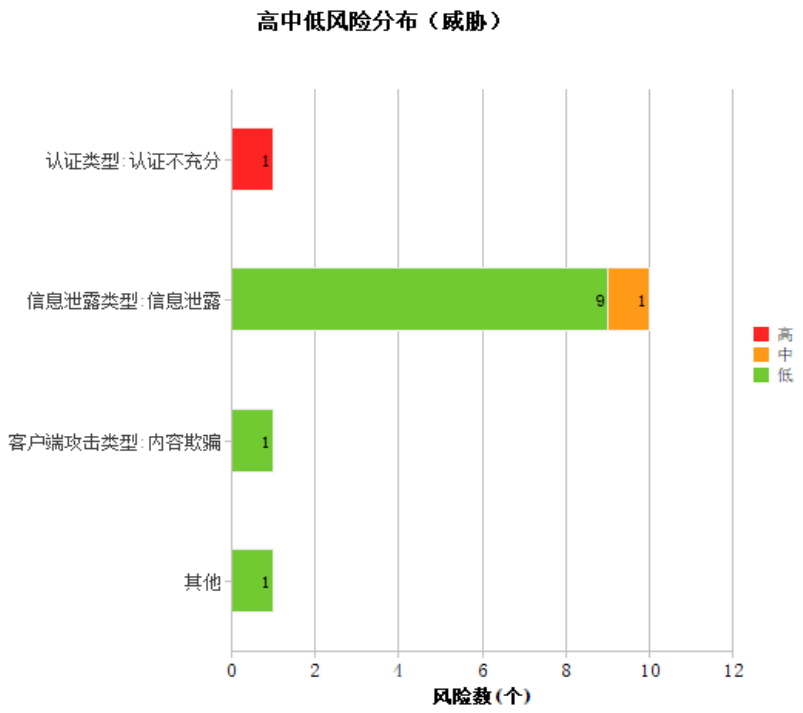
# 绿盟科技"网络安全漏洞扫描系统"安全评估报告-站点报表

## 目录

## 1. 站点概况

| | |
|---|---|
| **站点名称** | http://172.18.3.195/ |
| **站点 IP 地址** | 172.18.3.195 |
| **风险等级** | 非常危险 |
| **风险值** | 9.1 |
| **漏洞模板** | 自动匹配扫描 |
| **当前状态** | 扫描完成 |
| **漏洞信息** | 高风险：1 个<br>中风险：1 个<br>低风险：14 个 |
| **扫描链接数** | 65 |
| **时间统计** | 开始：2025-04-17 08:17:13<br>结束：2025-04-17 08:26:40<br>扫描耗时 9 分 27 秒 |
| **版本信息** | 系统版本：V6.0R04F04SP05<br>Web 插件版本：V6.0R02F00.3710 |

## 2. 风险分类统计




| 威胁分类 | 高风险 | 中风险 | 低风险 | 总计 |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 认证类型:认证不充分 | 1 | 0 | 0 | 1 |
| 信息泄露类型:信息泄露 | 0 | 1 | 9 | 10 |
| 客户端攻击类型:内容欺骗 | 0 | 0 | 1 | 1 |
| 其他 | 0 | 0 | 1 | 1 |
| 合计 | 1 | 1 | 11 | 13 |

# 3. Web 风险分布

## 3.1. Web 应用漏洞

### 3.1.1. 检测到目标 Redis 数据库未授权访问

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | redis-cli -h 172.18.3.195 |

| | |
|---|---|
| **详细描述** | Redis 是一个 NoSQL 的数据库(NoSQl 泛指非关系型的数据库,常用的 mysql 是关系型数据库)，数据通过键/值对存储在内存中。默认配置中，在服务运行的时候，会开放一个没有验证的 TCP/6379 端口,用来进行远程连接数据库。<br>由于 Redis 默认是运行在 TCP 的 6379 端口上的，而且默认安装没有设置用户访问认证，也没有访问 IP 限制，这样导致确定端口开放后，对 Redis 服务直接匿名访问,可对数据库进行匿名操作，信息泄露，执行命令等操作，造成严重问题。 |
| **解决办法** | 在 Redus 的配置文件中进行如下修改:<br>配置 port，修改端口号。<br>配置 bind 选项，限定可以连接 Redis 服务器的 IP。<br>配置 requirepass 选项，设置密码。<br>配置 rename-command CONFIG ""，禁用一些命令。 |
| **威胁分值** | 10 |
| **危险插件** | 否 |
| **发现日期** | 2014-12-16 |
| **CVSS 评分** | 7.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L) |

### 3.1.2. Hashicorp Consul Web UI 和 API 未授权访问

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195:8500/v1/agent/self (GET) |

| | |
|---|---|
| **详细描述** | HashiCorp Consul 是美国 HashiCorp 公司的一套分布式、高可用数据中心感知解决方案。该产品用于跨动态分布式基础架构连接和配置应用程序。 |

| | |
|---|---|
| | 未经身份验证的远程攻击者可能通过访问 Hashicorp Consul Web UI 和 API 来收集数据，注册服务并获得远程访问权限。 |
| **解决办法** | 只允许本地主机连接 Hashicorp Consul Web UI 和 API，设置防火墙和 ACL 进行限制。 |
| **威胁分值** | 5 |
| **危险插件** | 否 |
| **发现日期** | 2021-08-02 |
| **CVSS 评分** | 4.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.3. 检测到目标 URL 存在内部 IP 地址泄露

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/app.config.js?v=1.0.0-1744792831793 |
| **问题参数** | |
| **参考（验证）** | 172.18.3.195 |

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/entry/index-e413f910.js |
| **问题参数** | |
| **参考（验证）** | 192.168.0.234 |

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/app.config.js |
| **问题参数** | |
| **参考（验证）** | 172.18.3.195 |

| | |
|---|---|
| **详细描述** | 内部 IP 定义为下列 IP 范围内的 IP:<br>10.0.0.0 - 10.255.255.255<br>172.16.0.0 - 172.31.255.255<br>192.168.0.0 - 192.168.255.255<br><br>对攻击者而言，泄露内部 IP 非常有价值，因为它显示了内部网络的 IP 地址方案。知道内部网络的 IP 地址方案，可以辅助攻击者策划出对内部网络进一步的攻击。 |
| **解决办法** | 内部 IP 通常显现在 Web 应用程序/服务器所生成的错误消息中，或显现在 HTML/JavaScript 注释中。<br>[1] 关闭 Web 应用程序/服务器中有问题的详细错误消息。<br>[2] 确保已安装相关的补丁。<br>[3] 确保内部 IP 信息未留在 HTML/JavaScript 注释中。 |
| **威胁分值** | 3 |

| | |
|---|---|
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 5.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N) |

### 3.1.4. 检测到目标 X-Content-Type-Options 响应头缺失

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/ |

| | |
|---|---|
| **详细描述** | X-Content-Type-Options HTTP 消息头相当于一个提示标志，被服务器用来提示客户端一定要遵循在 Content-Type 首部中对 MIME 类型 的设定，而不能对其进行修改。这就禁用了客户端的 MIME 类型嗅探行为，换句话说，也就是意味着网站管理员确定自己的设置没有问题。<br><br>X-Content-Type-Options 响应头的缺失使得目标 URL 更易遭受跨站脚本攻击。 |
| **解决办法** | 将您的服务器配置为在所有传出请求上发送值为“nosniff”的“X-Content-Type-Options”头。对于 Apache，请参阅：http://httpd.apache.org/docs/2.2/mod/mod_headers.html<br>对于 IIS，请参阅：https://technet.microsoft.com/pl-pl/library/cc753133%28v=ws.10%29.aspx<br>对于 nginx，请参阅：http://nginx.org/en/docs/http/ngx_http_headers_module.html |
| **威胁分值** | 2 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 4.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.5. 检测到目标 X-XSS-Protection 响应头缺失

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/ |

| | |
|---|---|
| **详细描述** | HTTP X-XSS-Protection 响应头是 Internet Explorer，Chrome 和 Safari 的一个特性，当检测到跨站脚本攻击 (XSS)时，浏览器将停止加载页面。<br><br>X-XSS-Protection 响应头的缺失使得目标 URL 更易遭受跨站脚本攻击。 |
| **解决办法** | 将您的服务器配置为在所有传出请求上发送值为“1”（例如已启用）的“X-XSS-Protection”头。对于 Apache，请参阅：http://httpd.apache.org/docs/2.2/mod/mod_headers.html<br>对于 IIS，请参阅： |

| | |
|---|---|
| | https://technet.microsoft.com/pl-pl/library/cc753133%28v=ws.10%29.aspx<br>对于 nginx，请参阅：<br>http://nginx.org/en/docs/http/ngx_http_headers_module.html |
| **威胁分值** | 2 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 4.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.6. 检测到目标 URL 存在客户端（JavaScript）Cookie 引用

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/chunk/axios-36de57be.js |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/chunk/axios-36de57be.js |

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/entry/index-e413f910.js |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/entry/index-e413f910.js |

| | |
|---|---|
| **详细描述** | Cookie 通常由 Web 服务器创建并存储在客户端浏览器中，用来在客户端保存用户的身份标识、Session 信息，甚至授权信息等。客户端 JavaScript 代码可以操作 Cookie 数据。<br><br>如果在客户端使用 JavaScript 创建或修改站点的 cookie，那么攻击者就可以查看到这些代码，通过阅读代码了解其逻辑，甚至根据自己所了解的知识将其用来修改 cookie。一旦 cookie 包含了很重要的信息，譬如包含了权限信息等，攻击者很容易利用这些漏洞进行特权升级等攻击。 |
| **解决办法** | 1、避免在客户端放置业务/安全逻辑。<br>2、查找并除去客户端不安全的 JavaScript 代码，该代码可能会对站点造成安全威胁。 |
| **威胁分值** | 2 |
| **危险插件** | 否 |
| **发现日期** | 2010-01-01 |
| **CVSS 评分** | 2.6(CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.7. 检测到目标 Content-Security-Policy 响应头缺失

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/ |

| | |
|---|---|
| **详细描述** | HTTP 响应头 Content-Security-Policy 允许站点管理者控制用户代理能够为指定的页面加载哪些资源。除了少数例外情况，设置的政策主要涉及指定服务器的源和脚本结束点。<br><br>Content-Security-Policy 响应头的缺失使得目标 URL 更易遭受跨站脚本攻击。 |
| **解决办法** | 将您的服务器配置为发送 “Content-Security-Policy” 头。对于 Apache，请参阅：http://httpd.apache.org/docs/2.2/mod/mod_headers.html<br>对于 IIS，请参阅：https://technet.microsoft.com/pl-pl/library/cc753133%28v=ws.10%29.aspx<br>对于 nginx，请参阅：http://nginx.org/en/docs/http/ngx_http_headers_module.html |
| **威胁分值** | 2 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 4.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.8. 检测到目标 Strict-Transport-Security 响应头缺失

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/ |

| | |
|---|---|
| **详细描述** | Web 服务器对于 HTTP 请求的响应头中缺少 Strict-Transport-Security，这将导致浏览器提供的安全特性失效。 当 Web 服务器的 HTTP 头中包含 Strict-Transport-Security 头时，浏览器将持续使用 HTTPS 来访问 Web 站点，可以用来对抗协议降级攻击和 Cookie 劫持攻击。<br>其可选的值有： max-age=SECONDS，表示本次命令在未来的生效时间 includeSubDomains，可以用来指定是否对子域名生效 漏洞危害： Web 服务器对于 HTTP 请求的响应头中缺少 Strict-Transport-Security，这将导致浏览器提供的安全特性失效，更容易遭受 Web 前端黑客攻击的影响。 |
| **解决办法** | 1）修改服务端程序，给 HTTP 响应头加上 Strict-Transport-Security 如果是 java 服务端，可以使用如下方式添加 HTTP 响应头 response.setHeader("Strict-Transport-Security", "value") 如果是 php 服务端，可以使用如下方式添加 HTTP 响应头 header("Strict-Transport-Security: value") 如果是 asp 服务端，可以使用如下方式添加 HTTP 响应头 Response.AddHeader "Strict-Transport-Security", "value" 如果是 python django 服务端，可以使用如下方式添加 HTTP 响应头 response = HttpResponse() response["Strict-Transport-Security"] = "value" 如果是 python flask 服务端，可以使用如下方式添加 HTTP 响应头 response = make_response() response.headers["Strict-Transport-Security"] = "value"；<br>2）修改负载均衡或反向代理服务器，给 HTTP 响应头加上 Strict-Transport-Security 如果使用 Nginx、Tengine、Openresty 等作为代理服务器，在配置文件中写入如下内容即可添加 HTTP 响应头： add_header |

| | |
|---|---|
| | Strict-Transport-Security value; 如果使用 Apache 作为代理服务器，在配置文件中写入如下内容即可添加 HTTP 响应头： Header add Strict-Transport-Security "value"。 |
| **威胁分值** | 1 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 4.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.9. 检测到目标 Referrer-Policy 响应头缺失

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/ |

| | |
|---|---|
| **详细描述** | Web 服务器对于 HTTP 请求的响应头中缺少 Referrer-Policy，这将导致浏览器提供的安全特性失效。 当用户在浏览器上点击一个链接时，会产生一个 HTTP 请求，用于获取新的页面内容，而在该请求的报头中，会包含一个 Referrer，用以指定该请求是从哪个页面跳转页来的，常被用于分析用户来源等信息。但是也成为了一个不安全的因素，所以就有了 Referrer-Policy，用于过滤 Referrer 报头内容，其可选的项有： no-referrer no-referrer-when-downgrade origin origin-when-cross-origin same-origin strict-origin strict-origin-when-cross-origin unsafe-url 漏洞危害： Web 服务器对于 HTTP 请求的响应头中缺少 Referrer-Policy，这将导致浏览器提供的安全特性失效，更容易遭受 Web 前端黑客攻击的影响。 |
| **解决办法** | 1）修改服务端程序，给 HTTP 响应头加上 Referrer-Policy 如果是 java 服务端，可以使用如下方式添加 HTTP 响应头 response.setHeader("Referrer-Policy", "value") 如果是 php 服务端，可以使用如下方式添加 HTTP 响应头 header("Referrer-Policy: value") 如果是 asp 服务端，可以使用如下方式添加 HTTP 响应头 Response.AddHeader "Referrer-Policy", "value" 如果是 python django 服务端，可以使用如下方式添加 HTTP 响应头 response = HttpResponse( response["Referrer-Policy"] = "value" 如果是 python flask 服务端，可以使用如下方式添加 HTTP 响应头 response = make_response() response.headers["Referrer-Policy"] = "value"；<br>2）修改负载均衡或反向代理服务器，给 HTTP 响应头加上 Referrer-Policy 如果使用 Nginx、Tengine、Openresty 等作为代理服务器，在配置文件中写入如下内容即可添加 HTTP 响应头： add_header Referrer-Policy value; 如果使用 Apach 作为代理服务器，在配置文件中写入如下内容即可添加 HTTP 响应头： Header adc Referrer-Policy "value"。 |
| **威胁分值** | 1 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 4.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.10. 检测到目标 X-Permitted-Cross-Domain-Policies 响应头缺失

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/ |

| | |
|---|---|
| **详细描述** | Web 服务器对于 HTTP 请求的响应头中缺少 X-Permitted-Cross-Domain-Policies，这将导致浏览器提供的安全特性失效。 当一些在线的 Web Flash 需要加载其他域的内容时，很多 Web 会通过设置一个 crossdomain.xml 文件的方式来控制其跨域方式。很有可能有些开发者并没有修改 crossdomain.xml 文件的权限，但是又有和跨域的 Flash 共享数据的需求，这时候可以通过设置 X-Permitted-Cross-Domain-Policies 头的方式来替代 crossdomain.xml 文件，其可选的值有： none master-only by-content-type by-ftp-filename all 漏洞危害： Web 服务器对于 HTTP 请求的响应头中缺少 X-Permitted-Cross-Domain-Policies，这将导致浏览器提供的安全特性失效，更容易遭受 Web 前端黑客攻击的影响。 |
| **解决办法** | 1）修改服务端程序，给 HTTP 响应头加上 X-Permitted-Cross-Domain-Policies 如果是 java 服务端，可以使用如下方式添加 HTTP 响应头 response.setHeader("X-Permitted-Cross-Domain-Policies", "value") 如果是 php 服务端，可以使用如下方式添加 HTTP 响应头 header("X-Permitted-Cross-Domain-Policies: value") 如果是 asp 服务端，可以使用如下方式添加 HTTP 响应头 Response.AddHeader "X-Permitted-Cross-Domain-Policies", "value" 如果是 python django 服务端，可以使用如下方式添加 HTTP 响应头 response = HttpResponse() response["X-Permitted-Cross-Domain-Policies"] = "value" 如果是 python flask 服务端，可以使用如下方式添加 HTTP 响应头 response = make_response() response.headers["X-Permitted-Cross-Domain-Policies"] = "value"; 2）修改负载均衡或反向代理服务器，给 HTTP 响应头加上 X-Permitted-Cross-Domain-Policies 如果使用 Nginx、Tengine、Openresty 等作为代理服务器，在配置文件中写入如下内容即可添加 HTTP 响应头: add_header X-Permitted-Cross-Domain-Policies value; 如果使用 Apache 作为代理服务器，在配置文件中写入如下内容即可添加 HTTP 响应头： Header add X-Permitted-Cross-Domain-Policies "value"。 |
| **威胁分值** | 1 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 4.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.11. 检测到目标 X-Download-Options 响应头缺失

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |

| | |
|---|---|
| **参考（验证）** | http://172.18.3.195/ |

| | |
|---|---|
| **详细描述** | Web 服务器对于 HTTP 请求的响应头中缺少 X-Download-Options，这将导致浏览器提供的安全特性失效。 漏洞危害: Web 服务器对于 HTTP 请求的响应头中缺少 X-Download-Options，这将导致浏览器提供的安全特性失效，更容易遭受 Web 前端黑客攻击的影响。 |
| **解决办法** | 1）修改服务端程序，给 HTTP 响应头加上 X-Download-Options 如果是 java 服务端，可以使用如下方式添加 HTTP 响应头 response.setHeader("X-Download-Options", "value") 如果是 php 服务端，可以使用如下方式添加 HTTP 响应头 header("X-Download-Options: value") 如果是 asp 服务端，可以使用如下方式添加 HTTP 响应头 Response.AddHeader "X-Download-Options", "value" 如果是 python django 服务端，可以使用如下方式添加 HTTP 响应头 response = HttpResponse() response["X-Download-Options"] = "value" 如果是 python flask 服务端，可以使用如下方式添加 HTTP 响应头 response = make_response() response.headers["X-Download-Options"] = "value"；<br>2）修改负载均衡或反向代理服务器，给 HTTP 响应头加上 X-Download-Options 如果使用 Nginx、Tengine、Openresty 等作为代理服务器，在配置文件中写入如下内容即可添加 HTTP 响应头： add_header X-Download-Options value; 如果使用 Apache 作为代理服务器，在配置文件中写入如下内容即可添加 HTTP 响应头： Header add X-Download-Options "value"。 |
| **威胁分值** | 1 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 4.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N) |

### 3.1.12. 点击劫持：X-Frame-Options 未配置

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/ |

| | |
|---|---|
| **详细描述** | 点击劫持（ClickJacking）是一种视觉上的欺骗手段。攻击者使用一个透明的、不可见的 iframe，覆盖在一个网页上，然后诱使用户在该网页上进行操作，此时用户将在不知情的情况下点击透明的 iframe 页面。通过调整 iframe 页面的位置，可以诱使用户恰好点击在 iframe 页面的一些功能性按钮上。<br>HTTP 响应头信息中的 X-Frame-Options，可以指示浏览器是否应该加载一个 iframe 中的页面。如果服务器响应头信息中没有 X-Frame-Options，则该网站存在 ClickJacking 攻击风险。网站可以通过设置 X-Frame-Options 阻止站点内的页面被其他页面嵌入从而防止点击劫持。 |
| **解决办法** | 修改 web 服务器配置，添加 X-Frame-Options 响应头。赋值有如下三种：<br>1、DENY：不能被嵌入到任何 iframe 或者 frame 中。<br>2、SAMEORIGIN:页面只能被本站页面嵌入到 iframe 或者 frame 中。 |

| | |
|---|---|
| | 3、ALLOW-FROM uri：只能被嵌入到指定域名的框架中。<br>例如：<br>apache 可配置 http.conf 如下：<br><IfModule headers_module><br>Header always append X-Frame-Options "DENY"<br></IfModule><br><br>IIS 可配置相关网站的 Web.config 如下：<br><system.webServer><br>...<br><br><httpProtocol><br><customHeaders><br><add name="X-Frame-Options" value="deny" /><br></customHeaders><br></httpProtocol><br><br>...<br></system.webServer> |
| **威胁分值** | 1 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |
| **CVSS 评分** | 5.3(CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L) |

### 3.1.13. HTTP 缺少安全头漏洞

| | |
|---|---|
| **请求方式** | GET |
| **URL** | http://172.18.3.195/ |
| **问题参数** | |
| **参考（验证）** | http://172.18.3.195/ (Missing) permissions-policy, cross-origin-resource-policy, clear-site-data, access-control-allow-origin, cross-origin-embedder-policy, cross-origin-opener-policy, |

| | |
|---|---|
| **详细描述** | 目标未配置关键安全响应头（permissions-policy、clear-site-data、cross-origin-embedder-policy、cross-origin-opener-policy、cross-origin-resource-policy、access-control-allow-origin），可能导致浏览器安全策略缺失，增加跨站点攻击和资源共享风险。 |
| **解决办法** | 在服务器配置中添加以下安全响应头：<br><br>permissions-policy：限制特定功能的使用范围。<br>clear-site-data：清除浏览器存储的数据以防信息泄露。<br>cross-origin-embedder-policy：增强跨源资源加载安全性。<br>cross-origin-opener-policy：防止跨源窗口劫持。<br>cross-origin-resource-policy：限制跨源资源访问。 |

| | |
|---|---|
| | access-control-allow-origin：明确允许的跨域请求来源。<br>确保头信息内容根据应用需求正确配置。 |
| **威胁分值** | 1 |
| **危险插件** | 否 |
| **发现日期** | 2001-01-01 |

## 4. 外部链接列表

本站点外部链接总数为:16

| 域名 | 外链数目 |
|---|---|
| https://fontawesome.com | 2 |
| https://github.com | 4 |
| http://192.168.0.234:10200 | 1 |
| http://www.w3.org | 2 |
| https://vanilla-picker.js.org | 1 |
| https://raw.githubusercontent.com | 1 |
| https://vxeui.com | 2 |
| http://json-schema.org | 2 |
| http://192.168.0.234:10000 | 1 |
| 合计 | 16 |

## 5. 参考标准

### 5.1. 单一漏洞风险等级评定标准

| 危险程度 | 危险值区域 | 危险程度说明 |
|---|---|---|
| 高 | 7 <= 漏洞风险值 <= 10 | 攻击者可以远程操作系统文件、读写后台数据库、执行任意命令或进行远程拒绝服务攻击。 |
| 中 | 4 <= 漏洞风险值 < 7 | 攻击者可以利用 Web 网站攻击其他用户，读取系统文件或后台数据库。 |
| 低 | 0 <= 漏洞风险值 < 4 | 攻击者可以获取某些系统、文件的信息或冒用身份。 |

| 分值 | 评估标准 |
|---|---|
| 1 | 可远程获取 Web 服务器组件的版本信息。 |
| 2 | 目标 Web 服务器开放了不必要的服务。 |
| 3 | 可远程访问到某些不在目录树中的文件或读取服务器动态脚本的源码。 |
| 4 | 可远程因为会话管理的问题导致身份冒用。 |
| 5 | 可远程利用受影响的 Web 服务器攻击其他浏览网站的用户。 |
| 6 | 可远程读取系统文件或后台数据库。 |
| 7 | 可远程读取系统文件或后台数据库。 |
| 8 | 可远程以普通用户身份执行命令或进行拒绝服务攻击。 |

| | |
|---|---|
| 9 | 可远程以普通用户身份执行命令或进行拒绝服务攻击。 |
| 10 | 可远程以管理用户身份执行命令（不受限、容易利用）。 |

### 5.2. 站点风险等级评定标准

| 站点风险等级 | 站点风险值区域 |
|---|---|
| 非常危险 | 8 <= 站点风险值 <= 10 |
| 比较危险 | 5 <= 站点风险值 < 8 |
| 比较安全 | 1 <= 站点风险值 < 5 |
| 非常安全 | 0 <= 站点风险值 < 1 |

说明：
按照网络安全漏洞扫描系统的站点风险评估模型计算站点风险值。根据得到的站点风险值参考“站点风险等级评定标准”标识站点风险等级。
将站点风险等级按照风险值的高低进行排序，得到非常危险、比较危险、比较安全、非常安全四种网络风险等级。

### 5.3. 安全建议

随着越来越多的网络访问通过 Web 界面进行操作，Web 安全已经成为互联网安全的一个热点，基于 Web 的攻击广为流行，SQL 注入、跨站脚本等 Web 应用层漏洞的存在使得网站沦陷、页面篡改、网页挂马等攻击行为困扰着网站管理者并威胁着网站以及直接用户的安全。基于此，我们可从如下几个方面来消除这些风险，做到防患于未然：

1. 对网站的开发人员进行安全编码方面的培训，在开发过程避免漏洞的引入能起到事半功倍的效果。
2. 请专业的安全研究人员或安全公司对架构网站的程序和代码做全面的源码审计，修补所有发现的安全漏洞，这种白盒安全测试比较全面、深入，能发现绝大部分的安全问题。
3. 在网站上线前，使用 Web 应用漏洞扫描系统进行安全评估，并修补发现的问题；在网站上线后，坚持更新并使用网站安全监测系统，对整站以及关键页面进行周期和实时监测，及时消除发现的隐患。
4. 采用专业的 Web 安全防火墙产品，可以在不修改网站本身的情况下对大多数的 Web 攻击起到有效的阻断作用，绿盟科技提供了功能强大的 WAF 产品，可以满足用户在这方面的需求。
5. 建议网络管理员、系统管理员、安全管理员关注安全信息、安全动态及最新的严重漏洞，特别是影响到 Web 站点所使用的系统和软件的漏洞，应该在事前设计好应对规划，一旦发现系统受漏洞影响及时采取措施。