English | 繁中版 | Português (Brasil) | Français | 한국어 | Nederlands | Indonesia | ไทย | Русский | Українська | Español | Italiano | 日本語 | Deutsch | Türkçe | Tiếng Việt | Монгол | हिंदी | العربية | Polski | Македонски | ລາວ

开发安全的 API 所需要核对的清单

以下是当你在设计, 测试以及发布你的 API 的时候所需要核对的重要安全措施.

身份认证

- [ ] 不要使用 `Basic Auth` 使用标准的认证协议 (如 JWT, OAuth).
- [ ] 不要再造 `Authentication`, `token generating`, `password storing` 这些轮子, 使用标准的.
- [ ] 在登录中使用 `Max Retry` 和自动封禁功能.
- [ ] 加密所有的敏感数据.

JWT (JSON Web Token)

- [ ] 使用随机复杂的密钥 (`JWT Secret`) 以增加暴力破解的难度.
- [ ] 不要在请求体中直接提取数据, 要对数据进行加密 (`HS256` 或 `RS256`).
- [ ] 使 token 的过期时间尽量的短 (`TTL`, `RTTL`).
- [ ] 不要在 JWT 的请求体中存放敏感数据, 它是可破解的.

OAuth 授权或认证协议

- [ ] 始终在后台验证 `redirect_uri`, 只允许白名单的 URL.
- [ ] 每次交换令牌的时候不要加 token (不允许 `response_type=token`).
- [ ] 使用 `state` 参数并填充随机的哈希数来防止跨站请求伪造(CSRF).
- [ ] 对不同的应用分别定义默认的作用域和各自有效的作用域参数.

访问

- [ ] 限制流量来防止 DDoS 攻击和暴力攻击.
- [ ] 在服务端使用 HTTPS 协议来防止 MITM 攻击.
- [ ] 使用 `HSTS` 协议防止 SSLStrip 攻击.

输入

- [ ] 使用与操作相符的 HTTP 操作函数, GET（读取), POST（创建), PUT（替换/更新）以及 DELETE（删除记录), 如果请求的方法不适用于请求的资源则返回 405 Method Not Allowed.
- [ ] 在请求头中的 content-type 字段使用内容验证来只允许支持的格式 (如 application/xml, application/json 等等) 并在不满足条件的时候返回 406 Not Acceptable.
- [ ] 验证 content-type 的发布数据和你收到的一样 (如 application/x-www-form-urlencoded, multipart/form-data, application/json 等等).
- [ ] 验证用户输入来避免一些普通的易受攻击缺陷 (如 XSS, SQL-注入, 远程代码执行 等等).
- [ ] 不要在 URL 中使用任何敏感的数据 (credentials, Passwords, security tokens, or API keys), 而是使用标准的认证请求头.
- [ ] 使用一个 API Gateway 服务来启用缓存、访问速率限制 (如 Quota, Spike Arrest, Concurrent Rate Limit) 以及动态地部署 APIs resources.

处理

- [ ] 检查是否所有的终端都在身份认证之后, 以避免被破坏了的认证体系.
- [ ] 避免使用特有的资源 id. 使用 /me/orders 替代 /user/654321/orders
- [ ] 使用 UUID 代替自增长的 id.
- [ ] 如果需要解析 XML 文件, 确保实体解析(entity parsing)是关闭的以避免 XXE 攻击.
- [ ] 如果需要解析 XML 文件, 确保实体扩展(entity expansion)是关闭的以避免通过指数实体扩展攻击实现的 Billion Laughs/XML bomb.
- [ ] 在文件上传中使用 CDN.
- [ ] 如果需要处理大量的数据, 使用 Workers 和 Queues 来快速响应, 从而避免 HTTP 阻塞.
- [ ] 不要忘了把 DEBUG 模式关掉.

输出

- [ ] 发送 X-Content-Type-Options: nosniff 头.
- [ ] 发送 X-Frame-Options: deny 头.
- [ ] 发送 Content-Security-Policy: default-src 'none' 头.
- [ ] 删除指纹头 - X-Powered-By, Server, X-AspNet-Version 等等.
- [ ] 在响应中强制使用 content-type, 如果你的类型是 application/json 那么你的 content-type 就是 application/json.
- [ ] 不要返回敏感的数据, 如 credentials, Passwords, security tokens.
- [ ] 在操作结束时返回恰当的状态码. (如 200 OK, 400 Bad Request, 401 Unauthorized, 405 Method Not Allowed 等等).

持续集成和持续部署

- [ ] 使用单元测试和集成测试来审计你的设计和实现.
- [ ] 引入代码审查流程, 不要自行批准更改.
- [ ] 在推送到生产环境之前确保服务的所有组件都用杀毒软件静态地扫描过, 包括第三方库和其它依赖.
- [ ] 为部署设计一个回滚方案.

也可以看看:

- yosriady/api-development-tools - 用于构建RESTful HTTP+JSON API的有用资源集合。

贡献

为此存储库创建一个 fork, 进行修改, 并提交 pull request 来贡献. 如果您有任何问题, 请发送邮件至 team@shieldfy.io.