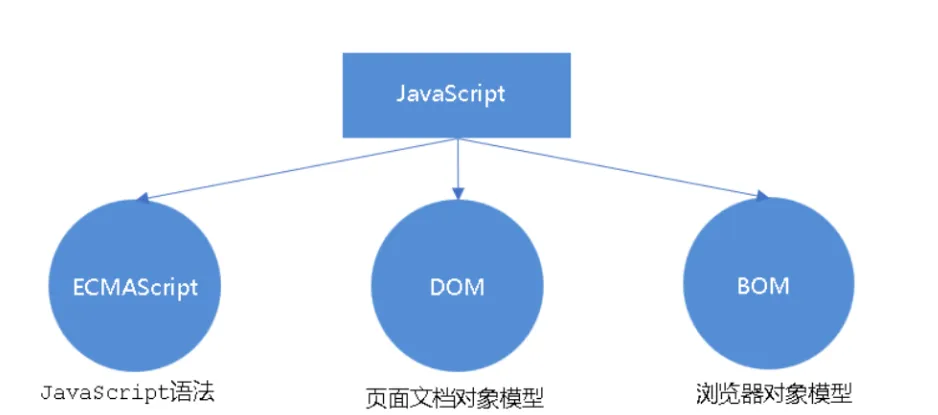
Web开发-JS应用&原生代码&Ajax技术&JQuery库&Axios库&前端校验&安全结合



#知识点：  
1、安全开发-原生PHP-超级全局变量  
2、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
0、安全开发-原生PHP-数据库通讯  
1、安全开发-原生PHP-身份验证技术  
2、安全开发-Cookie&Session&Token  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-弱类型脆弱  
2、安全开发-原生PHP-函数&数据类型  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-开发组件集合  
2、安全开发-原生PHP-模版引擎渲染  
3、安全开发-原生PHP-第三方编辑器  
---------------------------------------------  
1、安全开发-框架技术-ThinkPHP  
2、安全开发-框架安全-版本&写法  
3、安全开发-ThinkPHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-文件安全操作  
2、安全开发-原生PHP-上传读取删除包含等  
3、安全开发-原生PHP-代码审计文件安全---------------------------------------------  
1、安全开发-原生PHP-RCE安全  
2、安全开发-原生PHP-代码执行&命令注入  
3、安全开发-原生PHP-代码审计案例  
---------------------------------------------  
1、安全开发-原生PHP-PHP.INI安全  
2、安全开发-原生PHP-全局文件&单函数  
3、安全开发-原生PHP-流量检测&AI算法  
---------------------------------------------  
1、安全开发-原生JS-语法模型概念  
2、安全开发-Ajax技术-JQuery&Axios  
3、安全开发-Ajax应用-文件上传&用户登录  
  
#章节点：(待补充)  
#具体点：(待补充)



演示案例：

* WEB开发-原生JS-语法模型概念
* WEB开发-Ajax技术-JQuery&Axios
* WEB开发-Ajax前端验证-上传&登录

JavaScript被广泛用于Web应用开发，常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。嵌入动态文本于HTML页面；对浏览器事件做出响应，读写HTML元素，在数据被提交到服务器之前验证数据；检测访客的浏览器信息；控制用户凭据，包括创建和修改等。  
  
#安全结合：  
1、发现更多的有利用价值的信息（URL、域名、路径等等）  
测试站、后台路径、未公开的路径、api地址等等  
2、发现敏感信息（硬编码的帐号、pass、API密钥、注释等等）  
硬编码帐号可登录、测试帐号可被登录、密钥泄露、注释中开发信息等等  
3、发现危险的代码（eval、dangerouslySetInnerHTML等等）  
URL跳转，XSS跨站、模版注入（SSTI）等  
4、了解网站的逻辑校验功能  
前端检测，加密逆向，数据走向等  
  
#学习文档：  
1、原生JS教程  
https://www.w3school.com.cn/js/index.asp  
2、jQuery库教程  
https://www.w3school.com.cn/jquery/index.asp  
3、Axios库教程  
https://www.axios-http.cn/docs/intro  
  
#Ajax技术  
概念：Asynchronous JavaScript And XML：异步的 JavaScript 和 XML  
AJAX作用：  
1、数据交换：通过AJAX可给服务器发送请求，并获取服务器响应的数据  
2、后台发送：浏览器的请求是后台js发送给服务器的，js会创建单独的线程发送异步请求，这个线程不会影响浏览器的线程运行。  
3、局部刷新：浏览器接收到结果以后进行页面局部刷新  
JS-AJAX知识点：  
1、变量，函数，运算，事件等  
2、引用库，Ajax，JQuery，Axios等  
  
#文件上传  
1、布置前端页面  
2、JS获取提交数据  
3、JS对上传格式判断  
4、后端对上传数据处理  
前端JS进行后缀过滤，后端PHP进行上传处理  
安全问题：  
1、过滤代码能看到分析绕过   
2、禁用JS或删除过滤代码绕过  
  
#登录验证  
0、布置前端页面  
1、获取登录事件  
2、配置Ajax请求  
3、后端代码验证  
4、成功回调判断  
后端PHP进行帐号判断，前端JS进行登录处理  
安全问题：  
1、过滤代码能看到分析绕过   
2、禁用JS或修改返回分析绕过  
  
#测试案例  
JS代码分析验证逻辑

涉及资源：[资源下载地址](https://docs.qq.com/doc/DQ3Z6RkNpaUtMcEFr)