TURING 图灵原创

李建熠◎编著

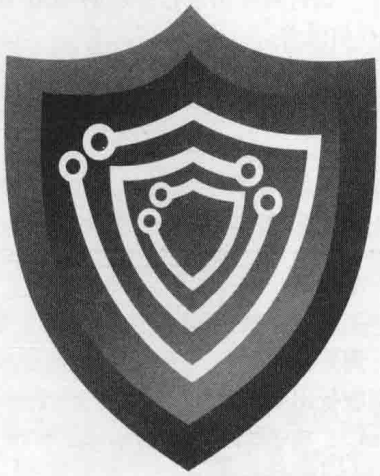
Web漏洞防护

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创

Web漏洞防护

李建熠◎编著



人民邮电出版社
北 京

图书在版编目（CIP）数据

Web漏洞防护 / 李建熠编著. -- 北京 : 人民邮电出版社, 2019.5
（图灵原创）
ISBN 978-7-115-51016-7

Ⅰ. ①W… Ⅱ. ①李… Ⅲ. ①计算机网络－网络安全－研究 Ⅳ. ①TP393.08

中国版本图书馆CIP数据核字(2019)第054765号

内 容 提 要

本书以 OWASP Top 10 2017 中涉及的漏洞为基础，系统阐述了常见的 Web 漏洞的防护方式。书中首先介绍了漏洞演示平台及一些常用的安全防护工具，然后对 OWASP Top 10 2017 中涉及的漏洞防护方式及防护工具进行了说明，接着介绍了如何通过 HTTP 响应头提升 Web 客户端自身对漏洞的防护能力，最后讨论了在无法更改应用程序源码的情况下，如何对应用进行外层的 WAF 防护。

本书适合关注于 Web 漏洞防护的任何读者。

◆ 编　　著　李建熠
责任编辑　王军花
责任印制　周昇亮

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京市艺辉印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：18
字数：431千字　　2019年5月第1版
印数：1－3 000册　　2019年5月北京第1次印刷

定价：79.00元

读者服务热线：**(010)51095183转600**　印装质量热线：**(010)81055316**
反盗版热线：**(010)81055315**
广告经营许可证：京东工商广登字 **20170147** 号

作者

李建熠

毕业于北京邮电大学，安全从业者及爱好者，曾任职于美团点评，参与过美团点评安全从0到1的构建。

微信连接

回复“安全”查看相关书单

微博连接

关注@图灵教育 每日分享IT好书

QQ连接

图灵读者官方群I：218139230
图灵读者官方群II：164939616

图灵社区
iTuring.cn

在线出版，电子书，《码农》杂志，图灵访谈

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

前　言

概述

目前大多数公司发现并处理安全漏洞的方式如下：利用上线前的安全检测、安全扫描、应急响应中心（SRC）等各种机制发现业务中存在的安全漏洞，然后提交给业务方进行修复，最后安全部门会对相关的漏洞进行复检以确认修复情况。整个过程初看并不存在什么缺陷，但是在实际的执行过程中往往存在下面这些问题。

- 安全人员只给出漏洞详情，并未向业务方提供相关漏洞的具体修复方案，造成业务方无法对漏洞进行修补。
- 安全人员给出了修复方案，但是过于简略，如对输入进行过滤、添加CSRF Token等，造成业务方无法理解修复方案的具体含义，同时后期的开发人员也无法将其作为一种安全编码规范来遵守。
- 安全人员给出了详细的修复方案，但是未给出修复漏洞的工具，业务方需自行查找相关工具，这给他们带来了很大的麻烦，可能会造成漏洞修复不及时，同时他们选用的工具也可能存在缺陷或漏洞，会造成漏洞修复不完全或引入新的漏洞。

目前，国内鲜有图书针对安全漏洞修复方案及修复工具进行详细介绍，网上相关文章大多只是简单介绍，缺少全面且详尽的讲解。

本书通过归纳国内外Web安全防护的相关文章（特别是与OWASP相关的文章），总结工作中的一些经验，对漏洞防护的方法进行说明。开发人员可以将这些防护方法作为日常编码的一种规范来遵从，以减少同类漏洞的出现。为了使漏洞修复更加便利、快捷，书中将会介绍漏洞防护的相关工具及其使用方法。同时，考虑到开发人员对于安全漏洞及其原理不甚了解，本书将会结合相关实例进行具体的说明，方便读者对安全漏洞有相对直观的认识。

关于Web安全防护，我们不能奢求通过某种手段防护所有的漏洞，而需要采取多种防护手段才能产生良好的防护效果。本书共介绍了3种防护手段。

- 在能够更改程序源码的情况下，对问题代码进行修改，从而对漏洞进行防护。这是防护Web漏洞的主要方式，也是本书介绍的重点，对应本书的第2章到第13章。

- 提升 Web 客户端自身防护漏洞的能力，以缓解潜在的漏洞危害。这种方式主要通过 HTTP 响应头进行设置，对应本书的第 14 章。
- 通过客户端与服务器之间的通信数据，分析出潜在的恶意流量并及时拦截，防止 Web 漏洞的产生。这种方式主要通过 WAF 进行防护，对应本书的第 15 章。

安全漏洞的种类繁多，一本书不可能囊括所有的漏洞。本书对漏洞的普遍性和严重性这两个因素进行考量，涵盖的范围主要是 OWASP Top 10 2017 中所涉及的漏洞。同时，考虑到目前大多数企业都将 Java 作为主要的开发语言，因此在漏洞说明及工具介绍上都将以 Java 为主，但是漏洞防护的方法是通用的，可以将它们应用到其他语言上。

本书涉及的内容比较庞杂，且本人在对某些知识的认识上存在一定的局限性，书中的内容难免会有些不足，甚至错误之处。如果大家在阅读中发现不足或错误之处，欢迎通过邮件方式（我的邮箱：balckarbiter@gmail.com）向本人指正。

本书结构

全书共包含 15 章。

第 1 章主要介绍了漏洞演示平台及书中将多次用到的安全防护工具，同时简要介绍了 OWASP Top 10 的内容。

第 2 章到第 13 章是本书的重点部分，详细介绍 OWASP Top 10 2017 中涉及的漏洞防护方式及防护工具，读者可以通过目录进一步了解其内容。

第 14 章主要介绍了如何通过 HTTP 响应头提升 Web 客户端自身对漏洞的防护能力。

第 15 章主要介绍了在无法更改应用程序源码的情况下，如何对应用进行外层的 WAF 防护。

致谢

感谢 OWASP 基金会项目与技术总监 Harold L. Blankenship 先生对 OWASP 相关文章引用的授权，本书最后将会详细列出所引用的文章，以示感谢。

感谢带我走上安全之路并给予我指导的陈伟先生、李晖老师、祁麟先生和张文师兄，感谢在写作过程中给予我支持与帮助的朋友。

感谢王军花和武芮欣两位女士，是她们的努力才使本书最终与广大读者见面，她们的专业意见给我了极大的帮助。

感谢我的父母、妻子和兄长，是他们的支持与陪伴，才使我最终完成本书。

目　录

第 1 章 使用工具介绍

本章首先介绍了漏洞演示工具 WebGoat 的安装及使用，然后讨论了书中将多次使用的安全工具 ESAPI、Apache Shiro 和 Spring Security，最后简述了 OWASP Top 10 的相关内容和本书结构。

1.1 WebGoat

WebGoat 是由 OWASP 组织使用 Java 语言研发的一款 Web 应用，用于演示 Web 应用中常见的漏洞。该项目是开源的，可以从 GitHub 上获取其最新源码（https://github.com/WebGoat/WebGoat）。WebGoat 中的漏洞类型与最新的 OWASP Top 10 保持一致，如最新版本的 WebGoat 8.0 与近期发布的 OWASP Top 10 2017 保持一致，是我们用于漏洞演示及学习的重要工具。注意不要将 WebGoat 部署到企业的生产环境中，因为攻击者可以利用上面的漏洞对生产环境造成损害。

本次对漏洞实例的展示使用最新版的 WebGoat 8.0，目前共有 11 个漏洞课程，包括 SQL 注入攻击、XXE（XML External Entity Attack，XML 外部实体攻击）、认证缺陷（认证绕过）、XSS（Cross-Site Scripting，跨站脚本）攻击、访问控制缺陷［包括不安全对象直接引用（IDOR）和功能级访问控制缺失］、不安全的登录、跨站请求伪造（CSRF）、组件缺陷及一些其他课程。除主体课程外，WebGoat 8.0 还提供了一个辅助工具 WebWolf，用于协助学习相关的课程。它主要包括 3 个功能：文件上传、邮件服务和请求消息记录。

下面我们简要介绍如何搭建 WebGoat 环境。这可以使用多种方式，如 Docker、Jar 包和源码，其中最简单的方式就是直接使用 Jar 包，具体的搭建步骤如下所示。

(1) 下载最新版的 Jar 包（地址：https://github.com/WebGoat/WebGoat/releases），这里需要用到 webgoat-server.jar 和 webwolf.jar。

(2) 使用命令 `java -jar webgoat-server.jar` 运行 WebGoat 主体课程，访问 http://localhost:8080/WebGoat/login 进行注册，然后使用注册的用户名和密码登录，即可开始学习课程，WebGoat 运行截图如图 1-1 所示。从运行截图中可以看出，左侧为课程的导航栏，右侧为课程的具体内容，此处不展开说明，第 2 章至第 13 章将详细介绍课程内容。

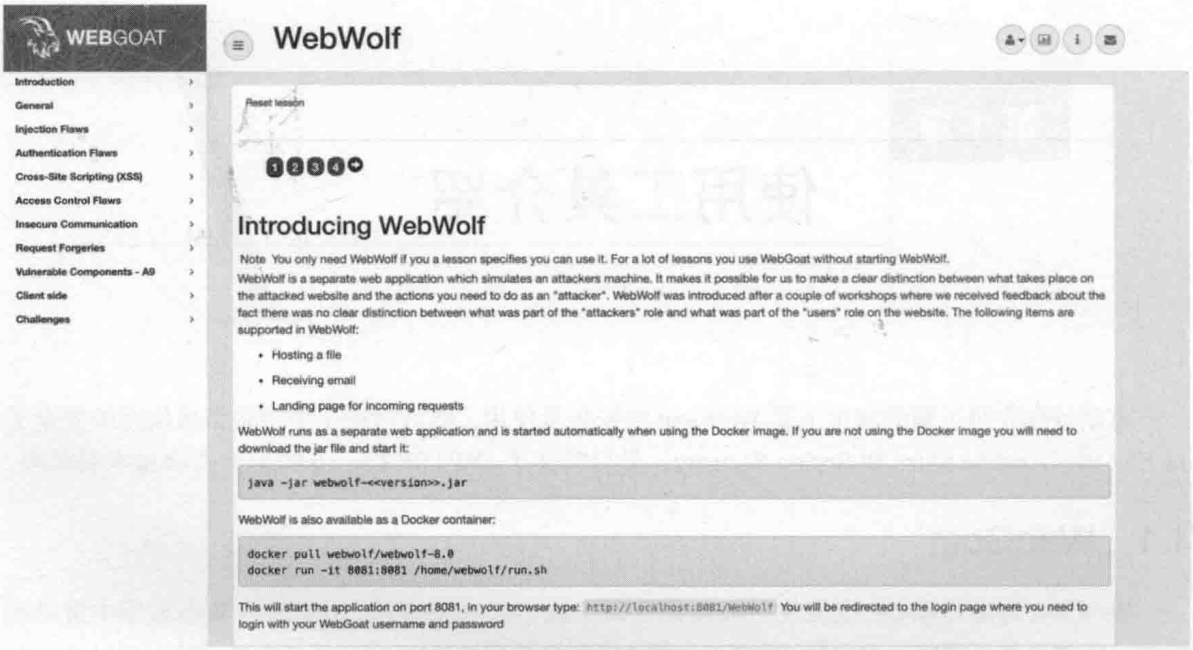


图 1-1　WebGoat 运行截图

(3) 使用命令 java -jar webwolf.jar 运行 WebWolf，访问 http://localhost:8081/login 进行注册并使用已注册的用户名及密码登录，即可使用 WebWolf 相关功能。WebWolf 的运行截图如图 1-2 所示。

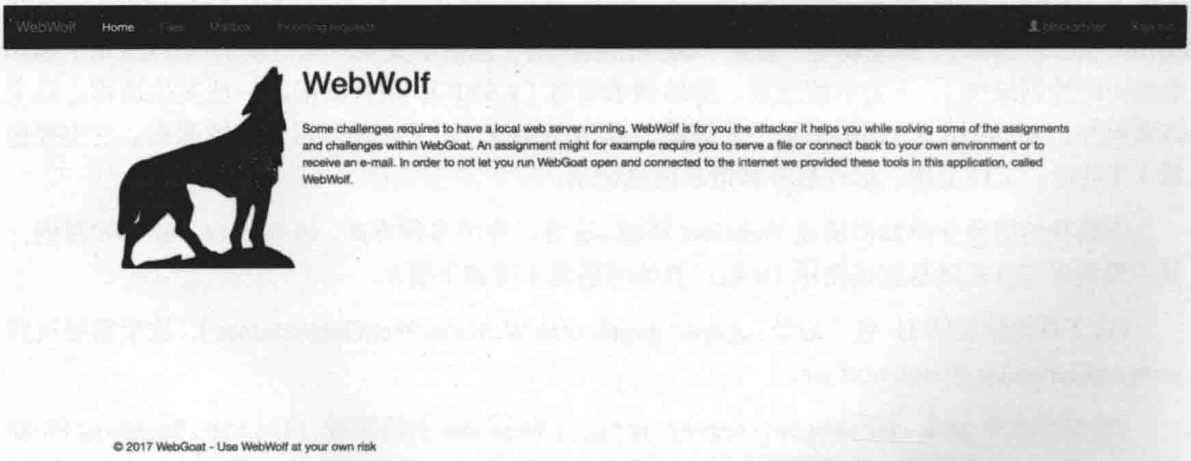


图 1-2　WebWolf 运行截图

WebWolf共包含3个功能，第一个功能为文件上传。WebWolf可以作为一个文件服务器使用，通过所上传文件最右侧的link超链接能够获取文件的存储地址，如图1-3所示。

图1-3 文件上传

第二个功能为模拟邮件服务器。WebGoat能够向名为user@random的邮箱发送邮件，该邮件服务器会收到发送的相关邮件，并在页面中展示，如图1-4所示。

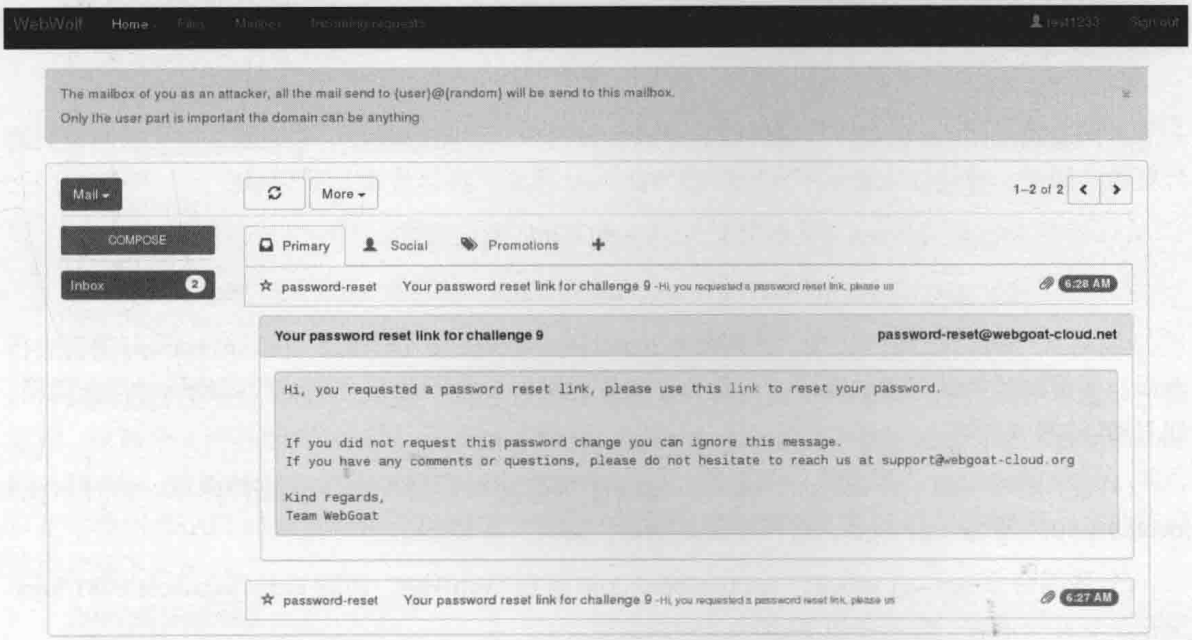


图1-4 邮件服务器

第三个功能为请求记录。向 URL 地址 http://127.0.0.1:8081/WebWolf/landing/*发起请求，该页面将会记录请求与响应消息，并在页面下方展示，如图 1-5 所示。

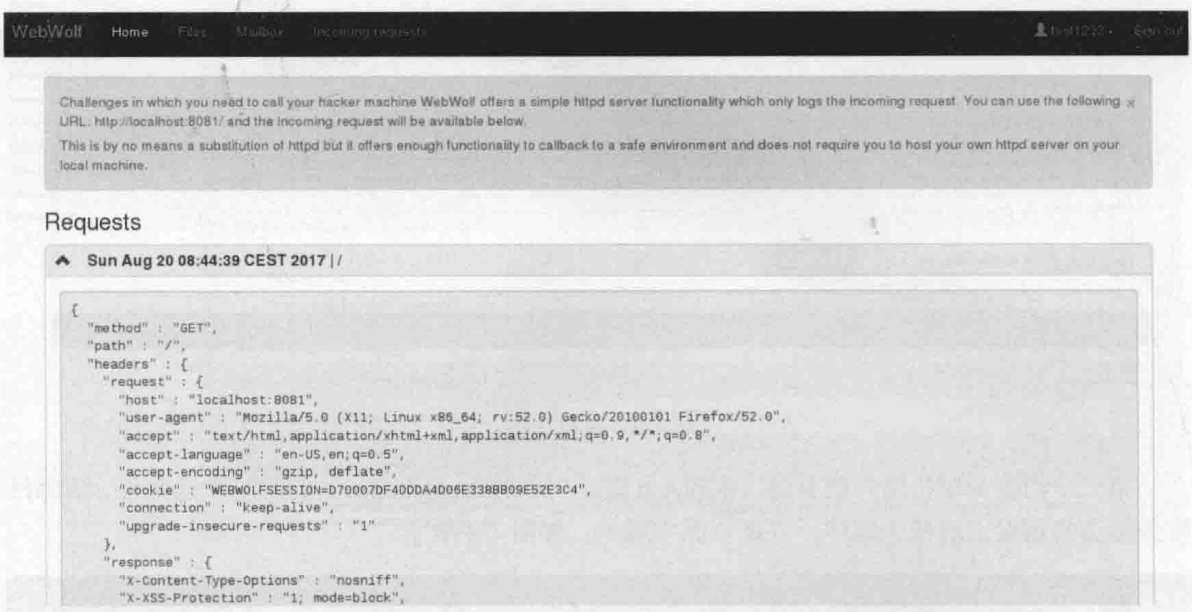


图 1-5　请求记录

虽然直接使用 Jar 包进行环境搭建的方式比较简单，但是也存在一些缺陷，如无法对源码进行更改。因此，我建议大家使用源码搭建 WebGoat 环境，具体步骤如下所示。

(1) 确保 Java 8、Maven（版本大于 3.2.1）和 Git 环境已安装。

(2) 使用命令 `git clone https://github.com/WebGoat/WebGoat.git` 下载 WebGoat 源码。

(3) 进入 WebGoat 源码目录，使用命令 `mvn clean compile install -DskipTests` 对项目进行编译。尽量跳过测试代码的编译，一是为了节省时间，二是因为某些测试代码编译时存在错误，如 XXE 的测试代码。

(4) 使用命令 `mvn -pl webgoat-server spring-boot:run` 运行 WebGoat 主体课程。访问 http://localhost:8080/WebGoat/login，进行注册并登录，即可开始课程的学习。

(5) 使用命令 `mvn -pl webwolf spring-boot:run` 运行 WebWolf，访问 http://localhost:8081/login 进行注册并登录后，即可使用 WebWolf 相关功能。

注意　WebGoat 8.0 没有默认用户，我们需要注册自己的用户，然后完成登录。

1.2 ESAPI

ESAPI的全称为The OWASP Enterprise Security API，是一款免费、开源的Web应用程序安全控制包。它为常见的Web安全漏洞提供了防护方案，并为构建Web应用安全防御体系提供了重要的参考。ESAPI除了提供Java安全包的实现外，还提供了.NET、ASP、PHP、Python和JavaScript等安全包的实现，因此使用它可以解决多种语言的安全问题。

ESAPI的使用比较简单，只需要导入依赖的log4j等相应的Jar包就可以开始使用了。下面展示使用Maven完成相关依赖及ESAPI的导入示例，其中ESAPI是通过本地Jar包导入的，Jar包的地址通过<systemPath>标签来定义：

```
<dependency>
    <groupId>org.owasp.esapi</groupId>
    <artifactId>esapi</artifactId>
    <version>2.1.0</version>
    <scope>system</scope>
    <systemPath>${basedir}/lib/esapi-2.1.0.jar</systemPath>
</dependency>
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>slf4j-api</artifactId>
<version>1.7.25</version>
</dependency>
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>slf4j-log4j12</artifactId>
<version>1.7.12</version>
</dependency>
<dependency>
<groupId>log4j</groupId>
<artifactId>log4j</artifactId>
<version>1.2.17</version>
</dependency>
```

如果直接使用这个程序，会因为缺少配置文件报错，所以需要导入两个配置文件才能够正常运行。这两个文件分别为ESAPI.properties和validation.properties，下载地址为https://github.com/ESAPI/esapi-java-legacy/tree/develop/configuration/esapi。将这两个文件放入resources目录，就可以正常使用ESAPI的相关功能了，这两个文件中某些内容的含义将在第13章中说明，文件的具体内容如下所示：

```
ESAPI.properties
ESAPI.printProperties=true
ESAPI.AccessControl=org.owasp.esapi.reference.DefaultAccessController
ESAPI.Authenticator=org.owasp.esapi.reference.FileBasedAuthenticator
ESAPI.Encoder=org.owasp.esapi.reference.DefaultEncoder
ESAPI.Encryptor=org.owasp.esapi.reference.crypto.JavaEncryptor
ESAPI.Executor=org.owasp.esapi.reference.DefaultExecutor
ESAPI.HTTPUtilities=org.owasp.esapi.reference.DefaultHTTPUtilities
ESAPI.IntrusionDetector=org.owasp.esapi.reference.DefaultIntrusionDetector
```

```
ESAPI.Logger=org.owasp.esapi.reference.Log4JLogFactory
ESAPI.Randomizer=org.owasp.esapi.reference.DefaultRandomizer
ESAPI.Validator=org.owasp.esapi.reference.DefaultValidator

# 认证配置
Authenticator.AllowedLoginAttempts=3
Authenticator.MaxOldPasswordHashes=13
Authenticator.UsernameParameterName=username
Authenticator.PasswordParameterName=password
# 记住 token 的持续时间（以天为单位）
Authenticator.RememberTokenDuration=14
# 会话超时（以分钟为单位）
Authenticator.IdleTimeoutDuration=20
Authenticator.AbsoluteTimeoutDuration=120

# 编码器配置
Encoder.AllowMultipleEncoding=false
Encoder.AllowMixedEncoding=false
Encoder.DefaultCodecList=HTMLEntityCodec,PercentCodec,JavaScriptCodec

# 加解密配置
Encryptor.MasterKey=tzfztf56ftv
Encryptor.MasterSalt=123456ztrewq
Encryptor.PreferredJCEProvider=
Encryptor.EncryptionAlgorithm=AES
Encryptor.CipherTransformation=AES/CBC/PKCS5Padding
Encryptor.cipher_modes.combined_modes=GCM,CCM,IAPM,EAX,OCB,CWC
Encryptor.cipher_modes.additional_allowed=CBC
Encryptor.EncryptionKeyLength=128
Encryptor.ChooseIVMethod=random
Encryptor.fixedIV=0x000102030405060708090a0b0c0d0e0f
Encryptor.CipherText.useMAC=true
Encryptor.PlainText.overwrite=true
Encryptor.HashAlgorithm=SHA-512
Encryptor.HashIterations=1024
Encryptor.DigitalSignatureAlgorithm=SHA1withDSA
Encryptor.DigitalSignatureKeyLength=1024
Encryptor.RandomAlgorithm=SHA1PRNG
Encryptor.CharacterEncoding=UTF-8
Encryptor.KDF.PRF=HmacSHA256

# HTTP 工具配置
HttpUtilities.UploadDir=C:\\ESAPI\\testUpload
HttpUtilities.UploadTempDir=C:\\temp
# 设置 Cookie 的属性
HttpUtilities.ForceHttpOnlySession=false
HttpUtilities.ForceSecureSession=false
HttpUtilities.ForceHttpOnlyCookies=true
HttpUtilities.ForceSecureCookies=true
# HTTP 头的最大值
HttpUtilities.MaxHeaderSize=4096

# 文件上传配置
HttpUtilities.ApprovedUploadExtensions=.zip,.pdf,.doc,.docx,.ppt,.pptx,.tar,.gz,.tgz,.rar,.war,
```