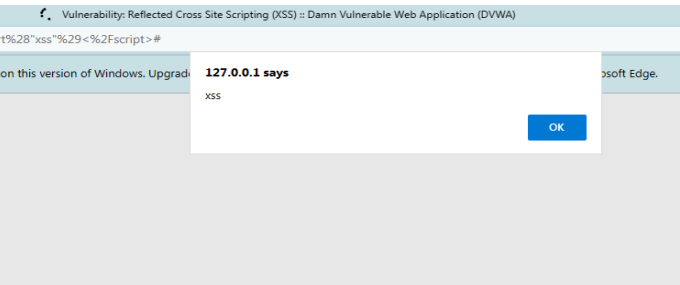
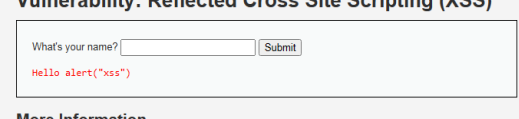
Low:

直接插入script脚本



执行成功，代码未采取任何过滤措施

Medium：

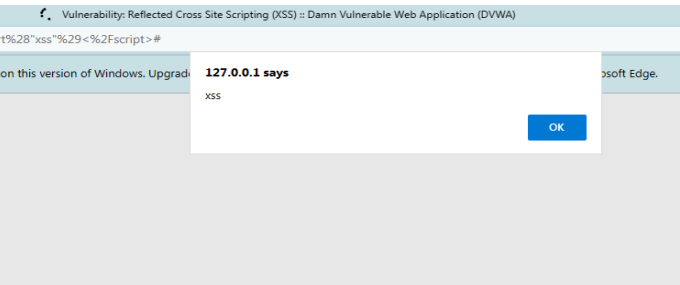


直接插入脚本后，脚本为执行。

猜想：<script> 和 </script> 被过滤掉

解决方案：利用html对大小写不敏感

插入脚本<Script>alert(“XSS”)</Script>



执行成功

尝试双写绕过

<scr<script>ipt> 这种方式源于str\_replace() 函数过滤后不会进行二次检查

原码解析：



原码将<script> 以空字符取代，实现过滤

High：

大小写和双写绕过均注入失败

可能使用了正则表达式，将script完全过滤。

考虑使用HTML标签

<img src = x onerror = alert(“XSS”)></img>

Onerror 事件是当图片加载发生错误的时候触发

原码解析：



Impossible：



Token是什么？

作为计算机术语时，token是令牌的意思。

通俗一点讲token就是暗号的意思，不同的token被授予不同的数据操作

Token 和 sessions 本质上没有区别，都是用来辨认用户的身份，只是识别机制有所不同

sessions：

用户首次登录后，服务端给用户发送sessions。下一次登录时用户通过cookie带上session便可以识别对象。此时serve端也保存着相应的session，由于serve不只一个，所以通过一个中间层保存session。

Token 用户信息只保存在客户端，服务端通过解析token进行身份验证。

两种方式各有利弊。