# （CVE-2020-12462）WordPress Plugin - Ninja Forms CSRF to XSS

## 一、漏洞简介

Ninja Forms 3.4.24.2之前的所有版本中存在一个严重的CSRF to XSS漏洞。成功利用此漏洞可以使得攻击者将WordPress网站中启用的Ninja Forms表单替换为包含恶意JavaScript的表单。当用户使用这些表单时触发xss漏洞。

## 二、漏洞影响

Ninja Forms < 3.4.24.2

## 三、复现过程

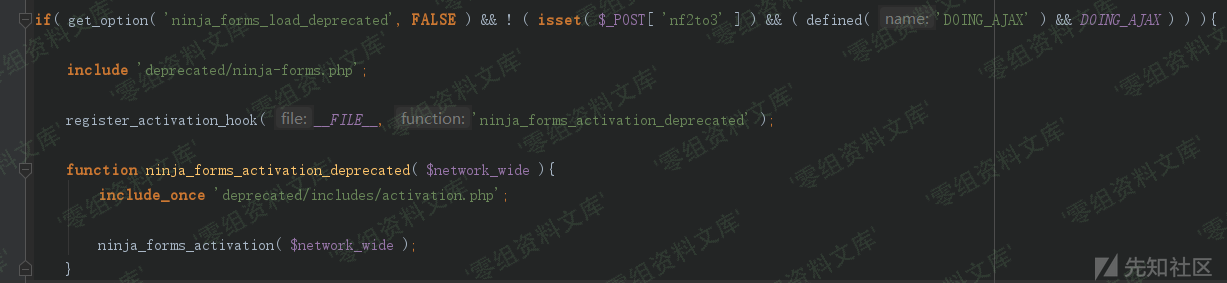
根据漏洞披露来看，本次漏洞存在于ninja-forms\lib\NF\_Upgrade.php文件 ninja\_forms\_ajax\_import\_form函数

在分析漏洞之前，我们先来了解下这个存在漏洞的文件是做什么用的。存在漏洞的文件名为NF\_Upgrade.php，从字面意义上来看，是升级的意思，但这个文件的功能并不是如同其命名（Upgrade）那样用来升级ninja-forms插件版本，而是涉及到ninja-forms的一个特殊功能——“降级”  
Ninja Forms插件中存在着一个名为 “降级”的功能。使用该功能用户可以将其表单样式和功能恢复为该插件2.9.x版本

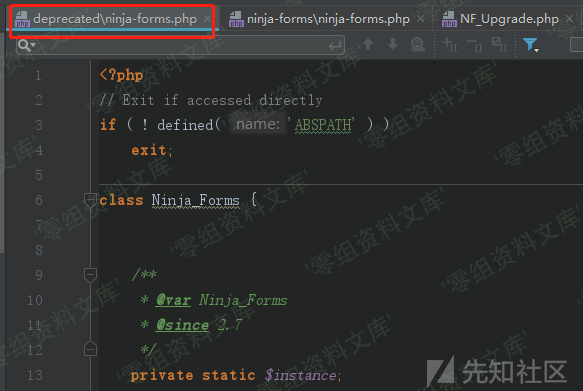


使用降级后，将删除目前安装的3.0版本的所有表单数据。因此该功能提示用户在使用该功能前导出表单数据。

从后台代码来看，点击降级按钮后，程序启用位于deprecated路径的旧版ninja-forms入口文件



下图是位于deprecated路径的2.x版本ninja-forms入口文件

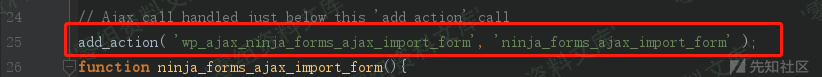


作为降级功能的一部分，ninja-forms编写了NF\_Upgrade.php文件，文件中AJAX函数旨在在使用“降级”模式时可以导入正常模式中导出的表单

在了解了NF\_Upgrade.php文件存在的意义之后，接下来分析下NF\_Upgrade.php文件中存在漏洞的ninja\_forms\_ajax\_import\_form函数

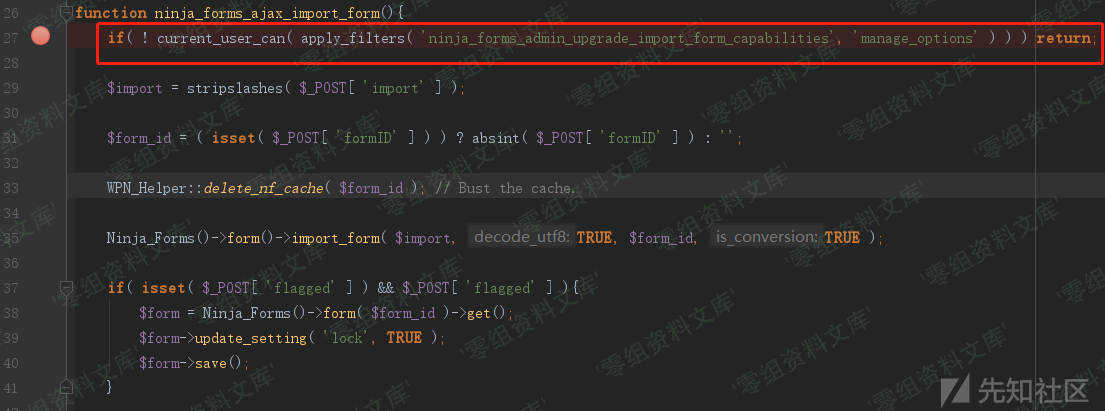


NF\_Upgrade.php文件在25行处通过add\_action注册了一个连接到ninja\_forms\_ajax\_import\_form函数的hook

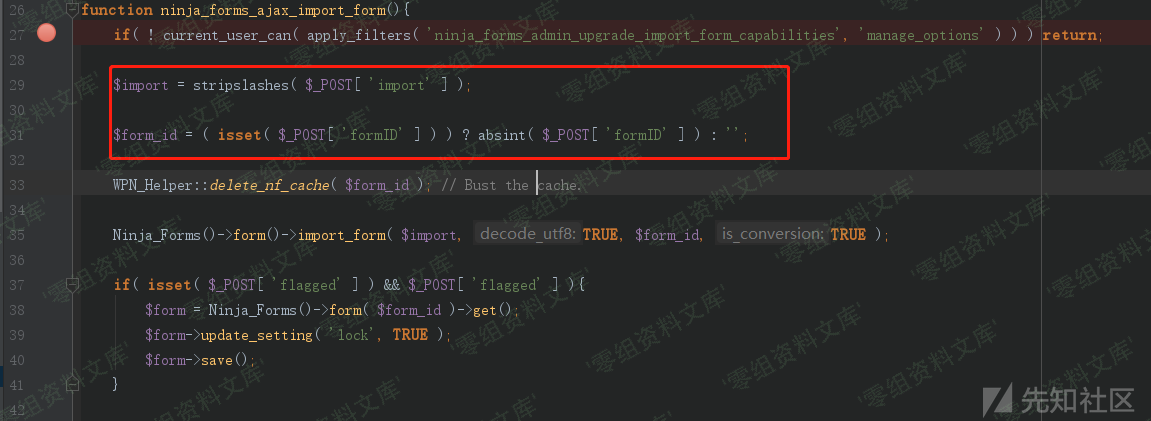


因此可以通过访问如下链接来触发ninja\_forms\_ajax\_import\_form函数

  
在了解了ninja\_forms\_ajax\_import\_form函数如何通过请求调用后，继续分析该函数



ninja\_forms\_ajax\_import\_form函数在27行处检验了用户权限



在29行处获取POST请求中import参数，在31行处获取POST请求中formID参数



程序在35行处将POST中提交的import参数与formID参数传入import\_form函数中进行导入处理

ninja\_forms\_ajax\_import\_form函数的作用，从对上文代码的理解，应该是用来给管理员提供导入表单功能用的。

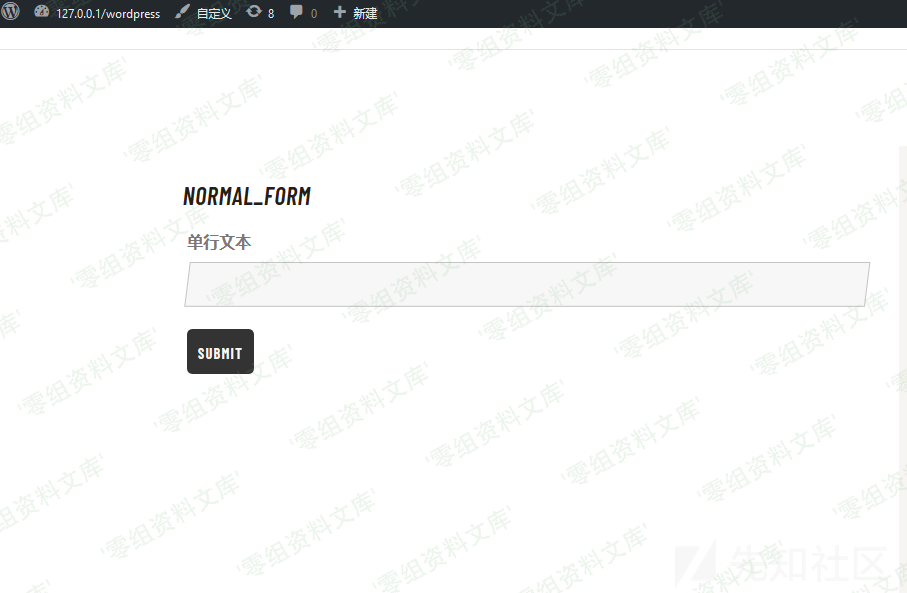
但是ninja\_forms\_ajax\_import\_form函数在安全上仅仅校验了使用者的权限，这点确实可以防止未授权访问的发生，但并未校验提交表单中的csrf token进行校验，并不能防范csrf漏洞，因此攻击者可以构造一个恶意页面链接诱骗管理员点击，从而向Ninja Forms插件提交一个导入恶意表单的请求。

ninja\_forms\_ajax\_import\_form函数接收POST传入的两个参数：import参数与formID参数，import参数中的内容是导入表单的数据，formID参数值是对应的表单id。经过实际测试发现，如果formID参数设置为一个已经存在的表单，则导入的表单内容将会覆盖原有表单。

在测试环境中有如下表单normal\_form。攻击者可以通过查看使用该表单的页面源码来获得formID，这里normal\_form的formID为2



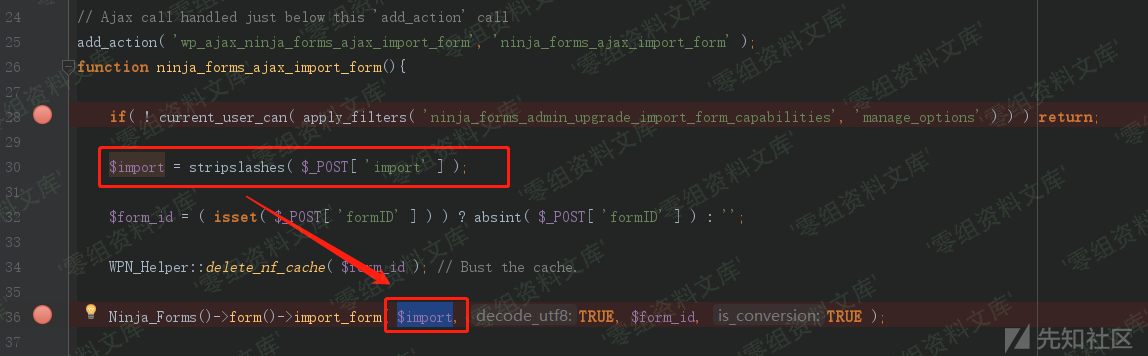
normal\_form是一个简单的单行文本表单



如果攻击者向在这个表单中插入一些恶意的js脚本，则可以在攻击时指定POST中formID为2

要覆盖的目标有了，但是导入的数据信息到底怎么构造呢？

回头看下ninja\_forms\_ajax\_import\_form函数



ninja\_forms\_ajax\_import\_form函数从POST请求中获取import参数，这个值就是导入模板的信息。接着程序将其传入import\_form方法中进行导入处理。

$\_POST[‘import’]值应该是什么样子的呢？只有知道了这个值的格式，我们才好构造payload

经过一番搜索，位于wp-content\plugins\ninja-forms\ninja-forms.php文件中  
13.png

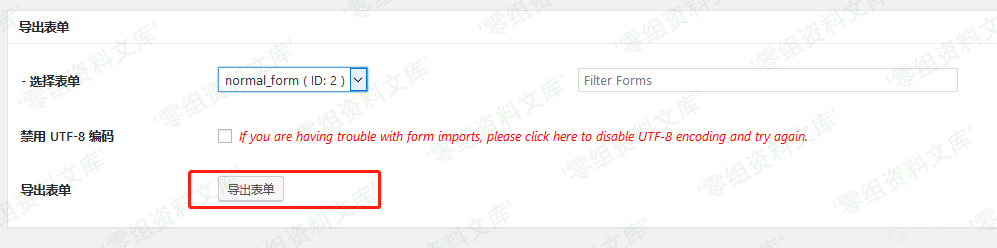
我们可以看到如上的代码

可见程序通过template方法读取位于wp-content\plugins\ninja-forms\includes\Templates  
文件夹的formtemplate-contactform.nff文件内容，并传入import\_form接口

formtemplate-contactform.nff文件是这个样子的



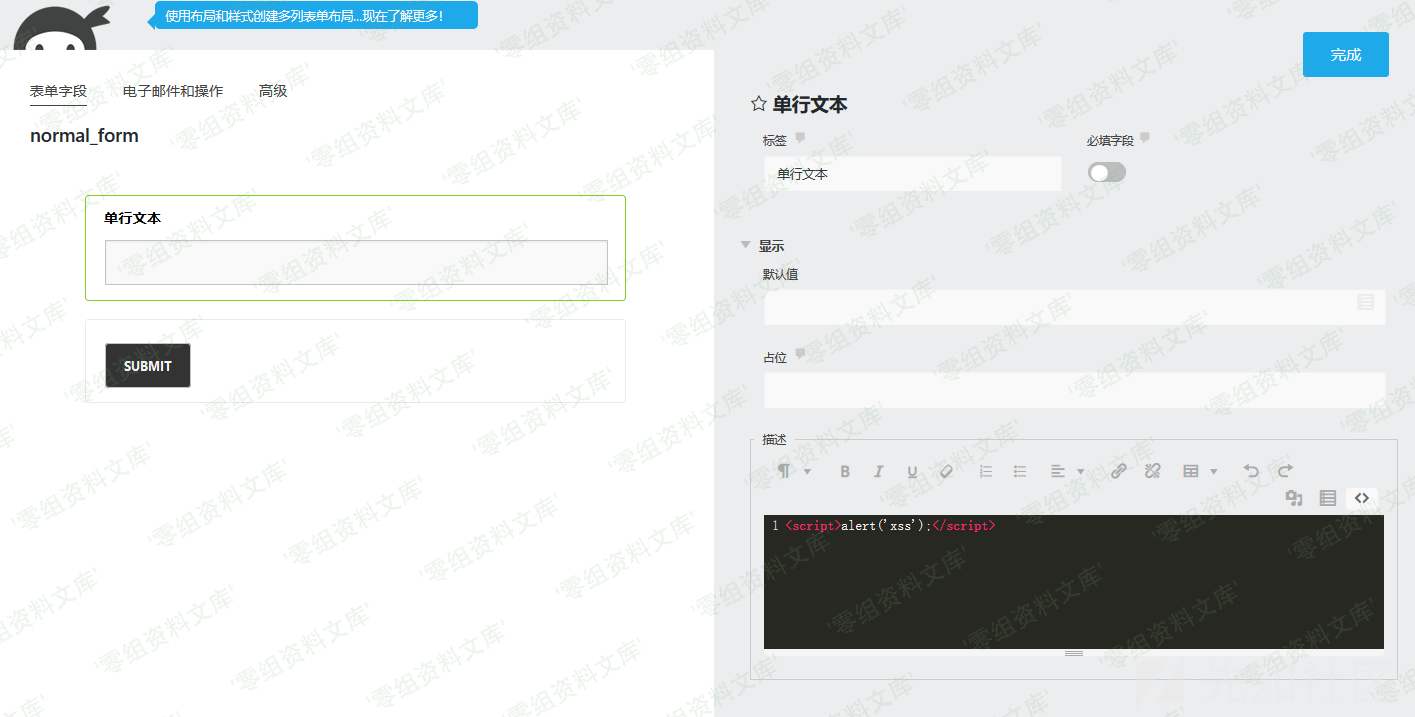
到这里就很好办了，.nff文件正是ninja forms插件导出表单的默认格式。



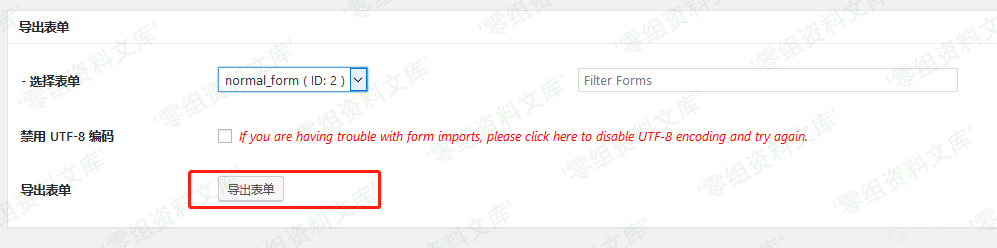
因此利用思路比较清晰了

1. 攻击者在本地模仿目标表单模板构造一个含有恶意代码的表单模板
2. 通过ninja forms插件将其导出为.nff文件
3. 构造csrf页面，该页面会在管理员访问时发起csrf攻击，使用恶意的表单模板替换原有模板
4. 诱使管理员访问
5. 当用户与管理员使用这个恶意表单的页面时，xss将会被触发

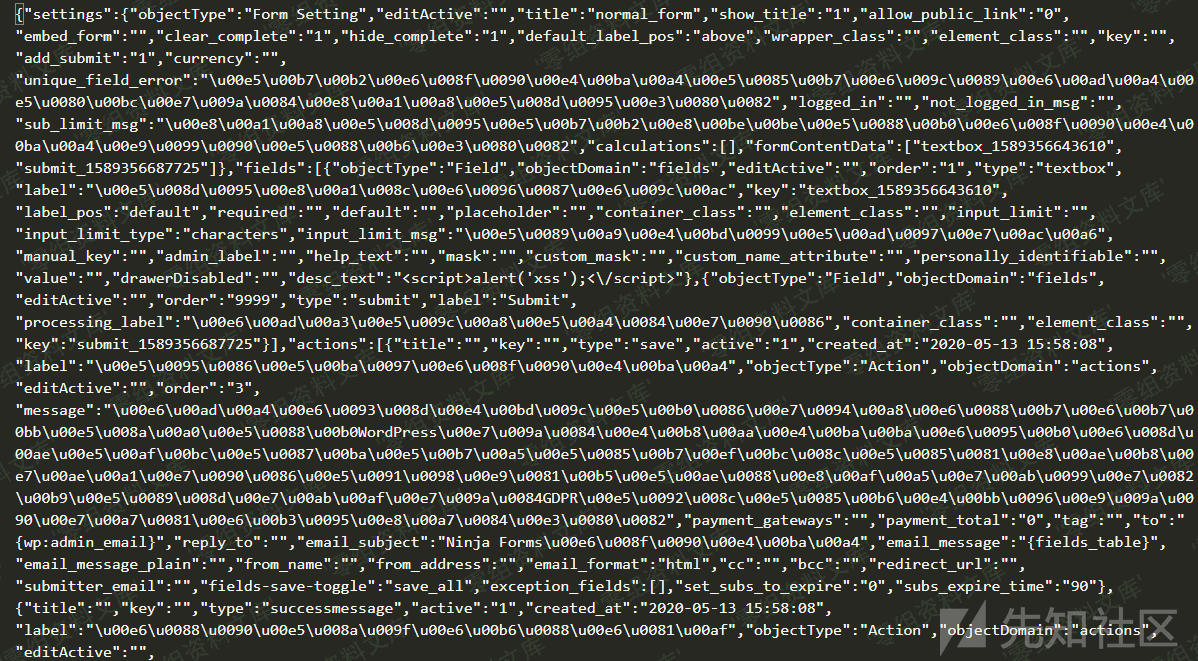
现在已经理清利用思路，接下来构造一个恶意的.nff文件。模仿上文的normal\_form表单，攻击者可以构造一个与之相似的恶意表单，与之不同的是，攻击者在描述处加入一行js脚本



攻击者将这个恶意的表单模板导出



导出的恶意表单部分内容如下



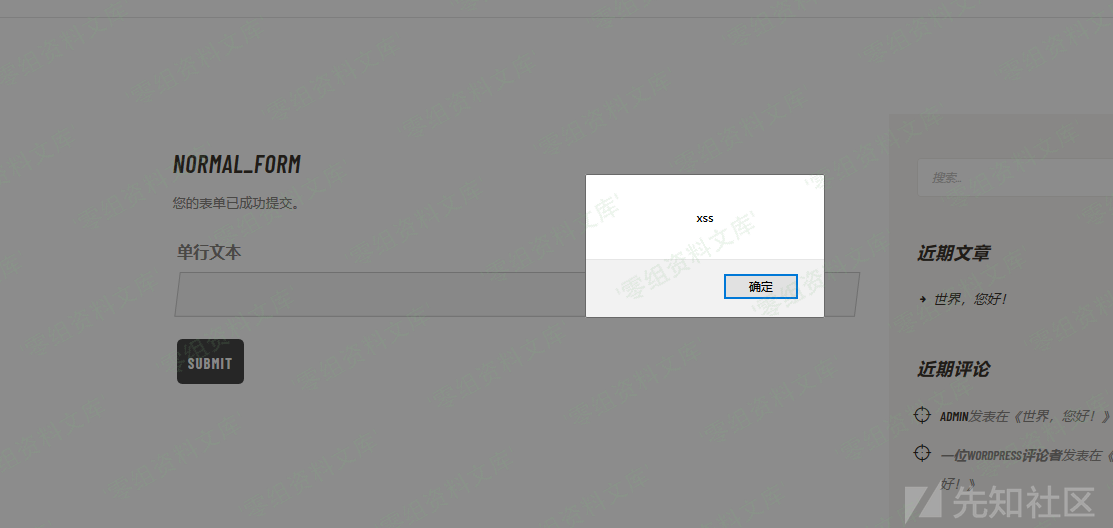
攻击者构造一个表单提交页面，表单提交的地址为http://www.0-sec.org/wordpress/wp-admin/admin-ajax.php?action=ninja\_forms\_ajax\_import\_form  
提交内容中import值为恶意文件内容、formID值为要覆盖的原normal\_form的id，并诱使管理员点击页面链接

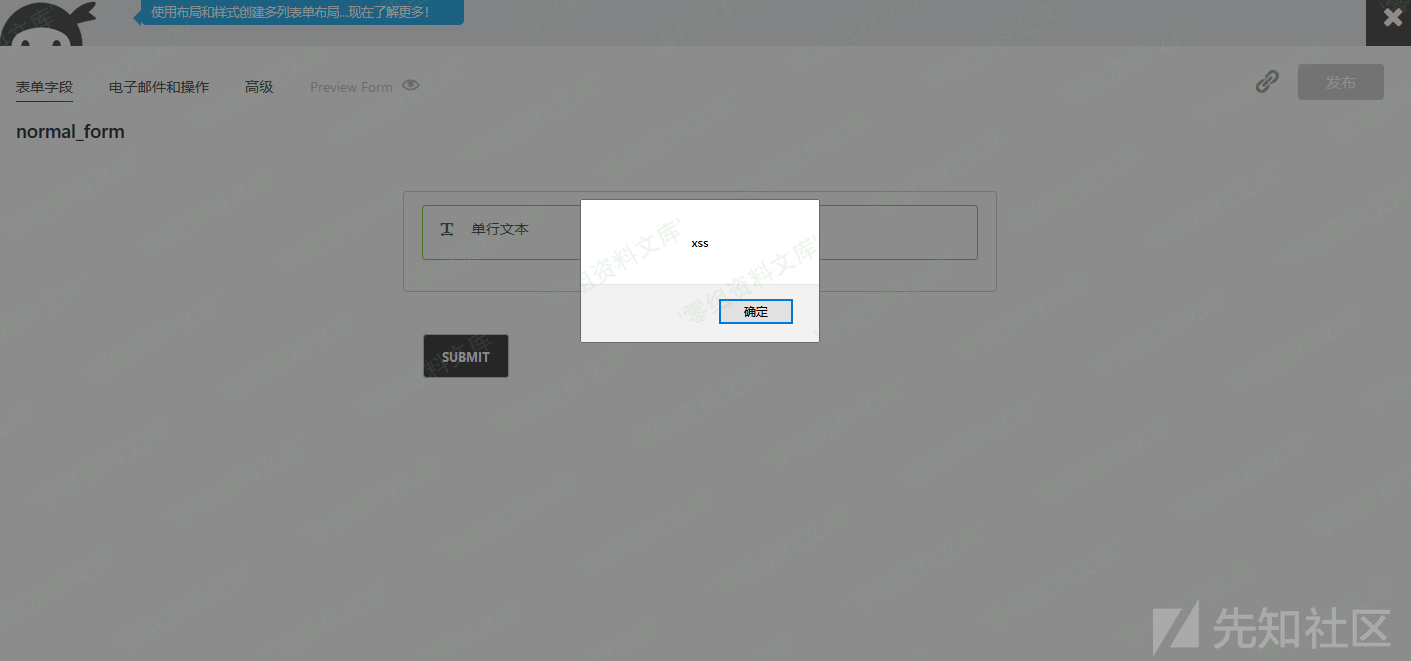
当攻击成功后，normal\_form表单将会被恶意的表单覆盖

当用户使用该表单进行提交时



点击submit时xss被触发

  
不仅如此，当管理员在后台管理编辑该表单时，当进入编辑页面，xss仍然可以被触发



## 参考链接

https://xz.aliyun.com/t/7768#toc-0