# 漏洞分析报告

Online Enrollment Management System In PHP And PayPal 1.0 Cross Site Scripting(CVE-2021-40577)

Vulnerability Report – NS20211207-01

Revision 1.0

## 基本信息

|  |  |
| --- | --- |
| Name\_zh | PHP和PayPal 1.0跨站点脚本(xss)在线注册管理系统（CVE-2021-40577） |
| Name\_en | Online Enrollment Management System In PHP And PayPal 1.0 Cross Site Scripting(CVE-2021-40577) |
| CVE | CVE-2021-39316 |
| CVSS评分 |  |
| 威胁等级 |  |
| CNNVD |  |
| 其他id |  |
| 威胁等级 |  |
| 受影响软件 | onlineenrolmentsystem |
| POC/EXP链接 |  |

**漏洞简介\_zh:**

存储交叉站点脚本（XSS）：

存储攻击是将注入的脚本永久存储在

目标服务器，

例如在数据库、消息论坛、访问者日志、评论字段等中。

当受害者

请求存储的信息时，受害者然后从服务器检索恶意脚本。

存储型 XSS 有时也称为持久型 XSS。

PHP和Paypal支付系统中的在线注册管理系统中发现当我们使用管理员登录添加用户时，Add-Users页面给出了一个 POST 请求，其中包含Name 字段框，其中 有一个输入字段，Name是易受Stored-XSS攻击的参数。

**漏洞简介\_en:**  
Stored Cross-site scripting(XSS):

Stored attacks are those where the injected script is permanently stored on

the target servers,

such as in a database, in a message forum, visitor log, comment field, etc.

The victim then retrieves the malicious script from the server when it

requests the stored information.

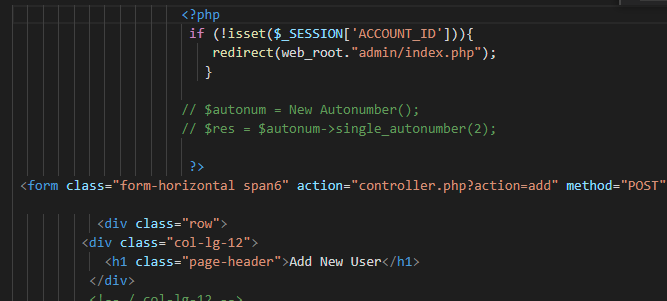
Stored XSS is also sometimes referred to as Persistent XSS..

Detailed description: It was found that when we Add User using the admin login, the Add-Users page is given a POST request containing the Name field box which has an input field And Name is the parameter that is vulnerable to Stored-XSS.

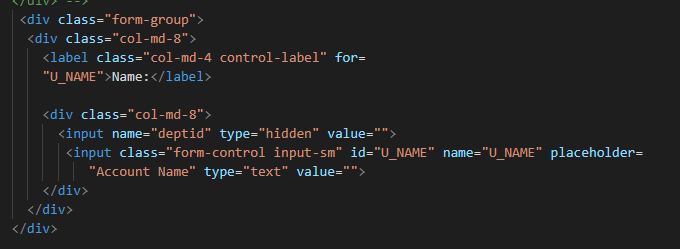
## 漏洞成因

后台的代码是在admin/user/index.php这里

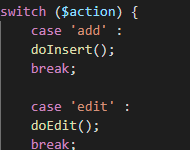
我们是在添加用户处触发了漏洞,所以跟进到add.php



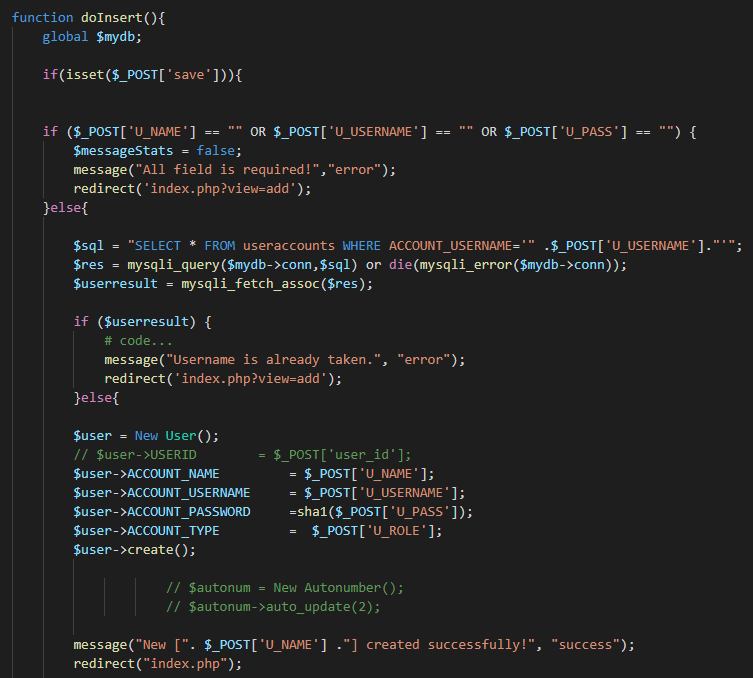
我们在这里可以看到,提交事件是交给了controller.php来处理



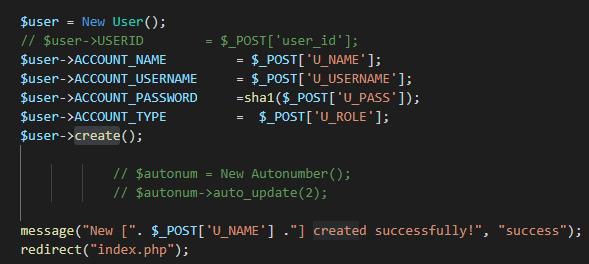
跟进到controller.php页面



在这里我们做的的是add操作,所以跟进到doInsert()函数中



按照正常的流程下去我们会执行到这里



这里就可以看出来了,他对U\_NAME是没有做过滤等操作的

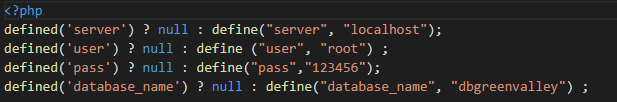
然后通过message("New [". $\_POST['U\_NAME'] ."] created successfully!", "success");这里,调用了U\_NAME

从而触发了我们的漏洞

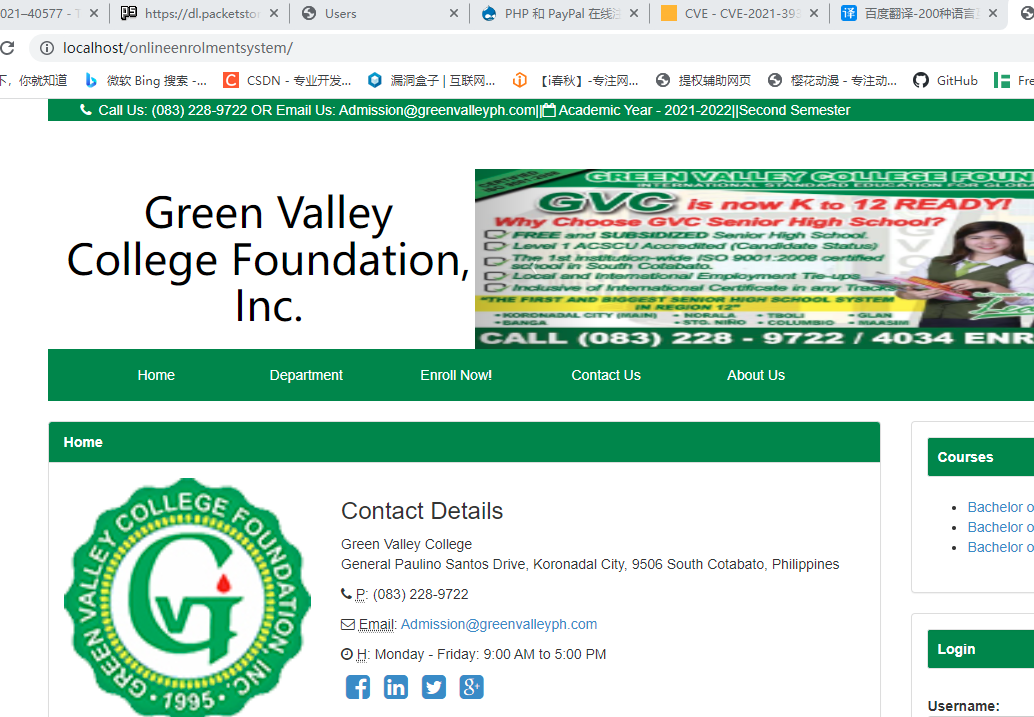
## 漏洞环境

https://www.sourcecodester.com/ 在该网站上可下载在线注册管理系统的源码.我们将其搭建在我们自己的环境中

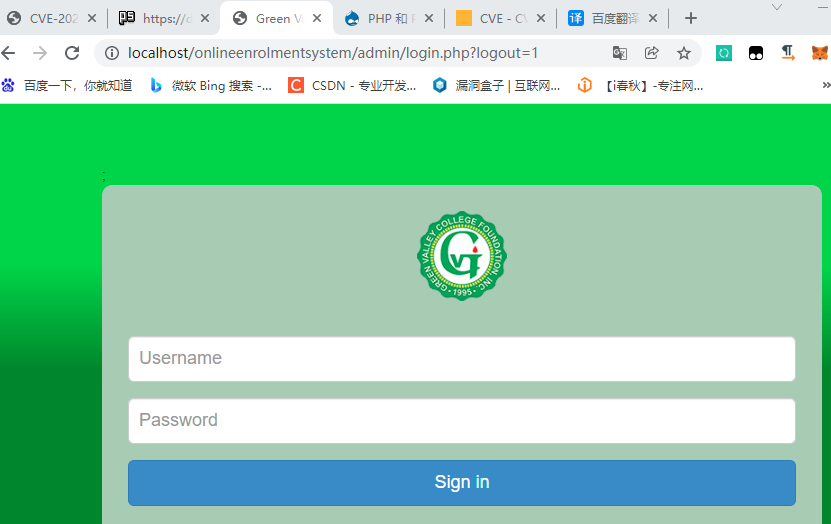
注意！该源码设置的mysql密码为空,如果我们的密码是设置好的话,需要到 index下的config中修改我们的sql语句的user和pass



在设置好之后,我们可以直接访问http://localhost/onlineenrolmentsystem/来到主界面

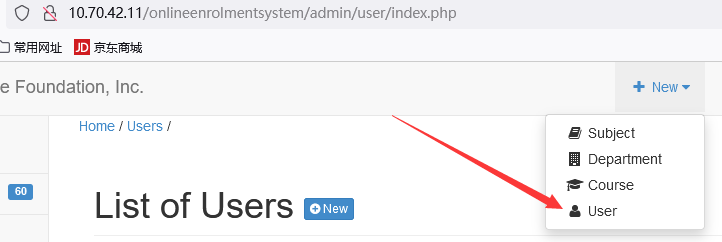


访问http://localhost/onlineenrolmentsystem/admin/login.php 可登录后台账户



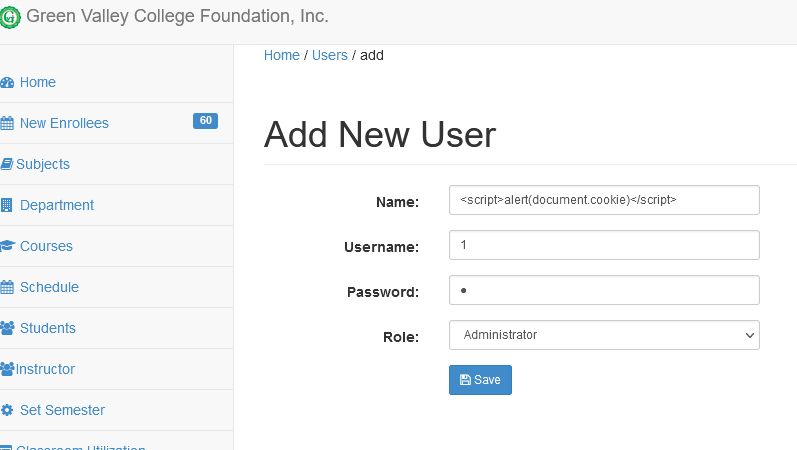
## 漏洞触发

在后台的New处选择User

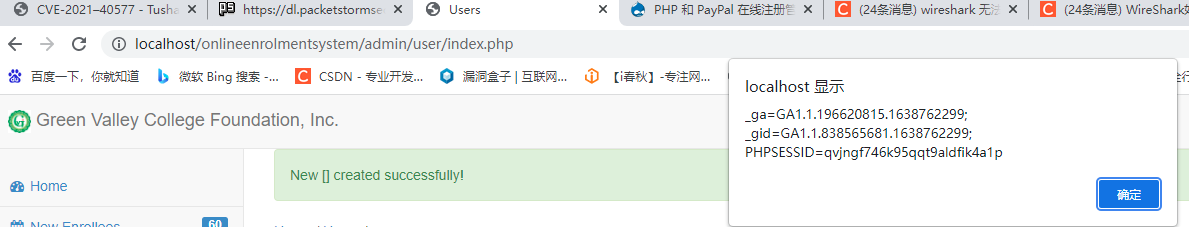


在Name处写上我们的XSS语句<script>(document.cookie)</script>

填写其他详细信息，然后单击“保存”按钮。



提交过后触发我们的XSS漏洞,可以拿到cookie



## 攻击检测

**Content Security Policy**

严格的 CSP 在 XSS 的防范中可以起到以下的作用：

禁止加载外域代码，防止复杂的攻击逻辑。

禁止外域提交，网站被攻击后，用户的数据不会泄露到外域。

禁止内联脚本执行（规则较严格，目前发现 GitHub 使用）。

禁止未授权的脚本执行（新特性，Google Map 移动版在使用）。

合理使用上报可以及时发现 XSS，利于尽快修复问题。

**输入内容长度控制**

对于不受信任的输入，都应该限定一个合理的长度。虽然无法完全防止 XSS 发生，但可以增加 XSS 攻击的难度。

**其他安全措施**

HTTP-only Cookie: 禁止 JavaScript 读取某些敏感 Cookie，攻击者完成 XSS 注入后也无法窃取此 Cookie。

验证码：防止脚本冒充用户提交危险操作。

## 攻击代码

<script>alert(document.cookie)</script>,简单的使用JavaScript把我们的cookie用弹窗的方式展现出来

## 解决方案

**利用模板引擎**

开启模板引擎自带的 HTML 转义功能。例如：

在 ejs 中，尽量使用 <%= data %> 而不是 <%- data %>；

在 doT.js 中，尽量使用 {{! data } 而不是 {{= data }；

在 FreeMarker 中，确保引擎版本高于 2.3.24，并且选择正确的 freemarker.core.OutputFormat。

**避免内联事件**

尽量不要使用 onLoad="onload('{{data}}')"、onClick="go('{{action}}')" 这种拼接内联事件的写法。在 JavaScript 中通过 .addEventlistener() 事件绑定会更安全。

**避免拼接 HTML**

前端采用拼接 HTML 的方法比较危险，如果框架允许，使用 createElement、setAttribute 之类的方法实现。或者采用比较成熟的渲染框架，如 Vue/React 等。

**时刻保持警惕**

在插入位置为 DOM 属性、链接等位置时，要打起精神，严加防范。

**增加攻击难度，降低攻击后果**

通过 CSP、输入长度配置、接口安全措施等方法，增加攻击的难度，降低攻击的后果。

**主动检测和发现**

可使用 XSS 攻击字符串和自动扫描工具寻找潜在的 XSS 漏洞。

## 参考资料

https://dl.packetstormsecurity.net/2112-exploits/oemspp10-xss.txt

https://medium.com/@VAPT01/cve-2021-40577-ec96a831ba71