

写在翻译稿前面

最近，笔者在研究一些与WordPress漏洞相关内容。Sucuri、RIPS、Fortinet等安全公司对WordPress有着一系列的深入研究，因此笔者计划陆续将一些有意思文章翻译出来。

这篇文章是来自Sucuri Labs，本文介绍了强大的 WordPress 统计分析插件 WP Statistics的一个xss漏洞以及与防火墙的组合利用，文中括号加粗的是我自己分析这个原稿的一些见解，大家可以在评论里一起讨论下。

下面翻译稿正文开始

风险评级：次要：可用于有针对性攻击，但需要特定配置后利用。

漏洞名称：存储型XSS漏洞

修复版本：12.6.7

WordPress插件[WP Statistics](#)具有50万用户的活动安装基础，在12.6.7之前的版本上存在未经身份验证的存储XSS漏洞。

此漏洞只能在某些配置下使用 - 默认设置不易受到攻击。

时间线

- 2019/06/26 - 初步与开发人员取得联系。
- 2019/06/27 - 开发人员回应，披露漏洞。
- 2019/06/30 - 拟议审核补丁。
- 2019/07/01 - 版本12.6.7发布，修复漏洞。

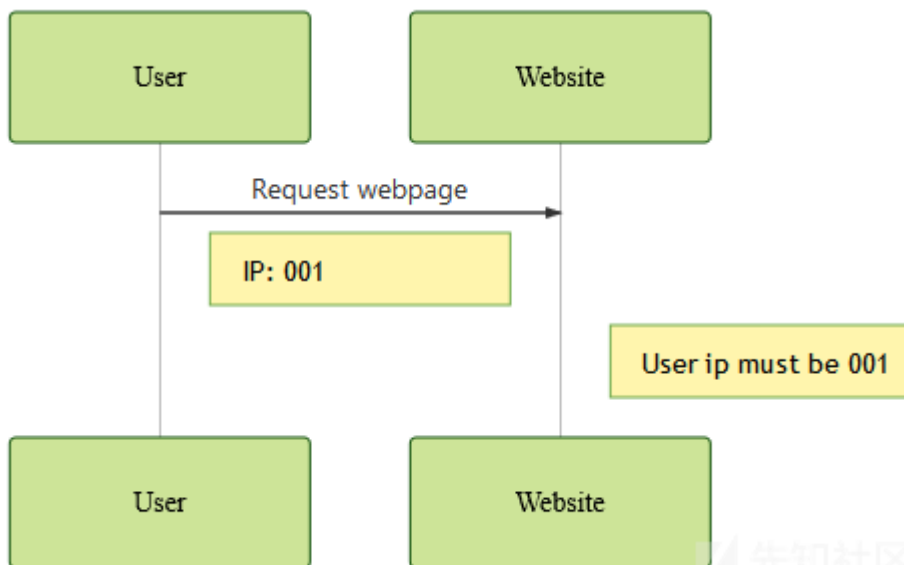
通过IP操作存储型XSS漏洞

在该插件某些配置情况下，网站可以使用header来查找访问者的IP。

在使用防火墙的环境中，经常会需要这样的方式(译者注：使用header来查找访问者的IP)，否则所有访问者都会拥代理的IP而不是自己的IP。

为什么使用防火墙需要使用Header

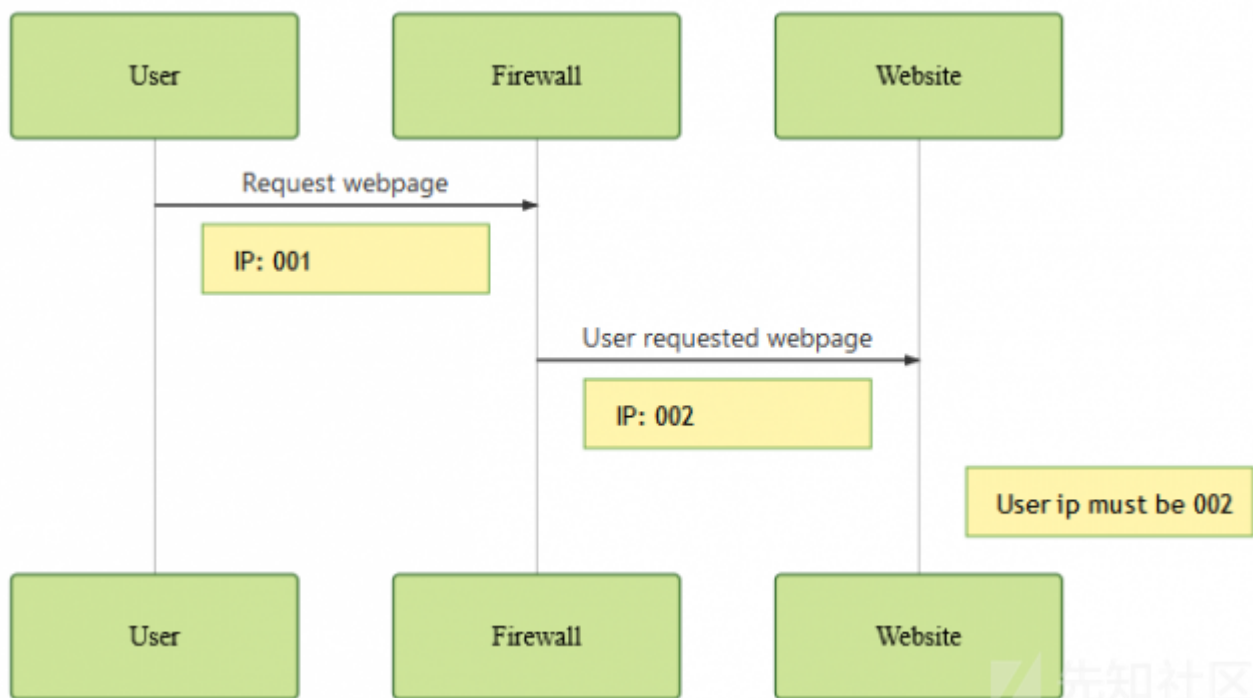
默认情况下，网站可以轻松找到访问用户的IP地址。并且找到的一定是发起请求User的IP，如下图所示：



请求访问没有防火墙的网站

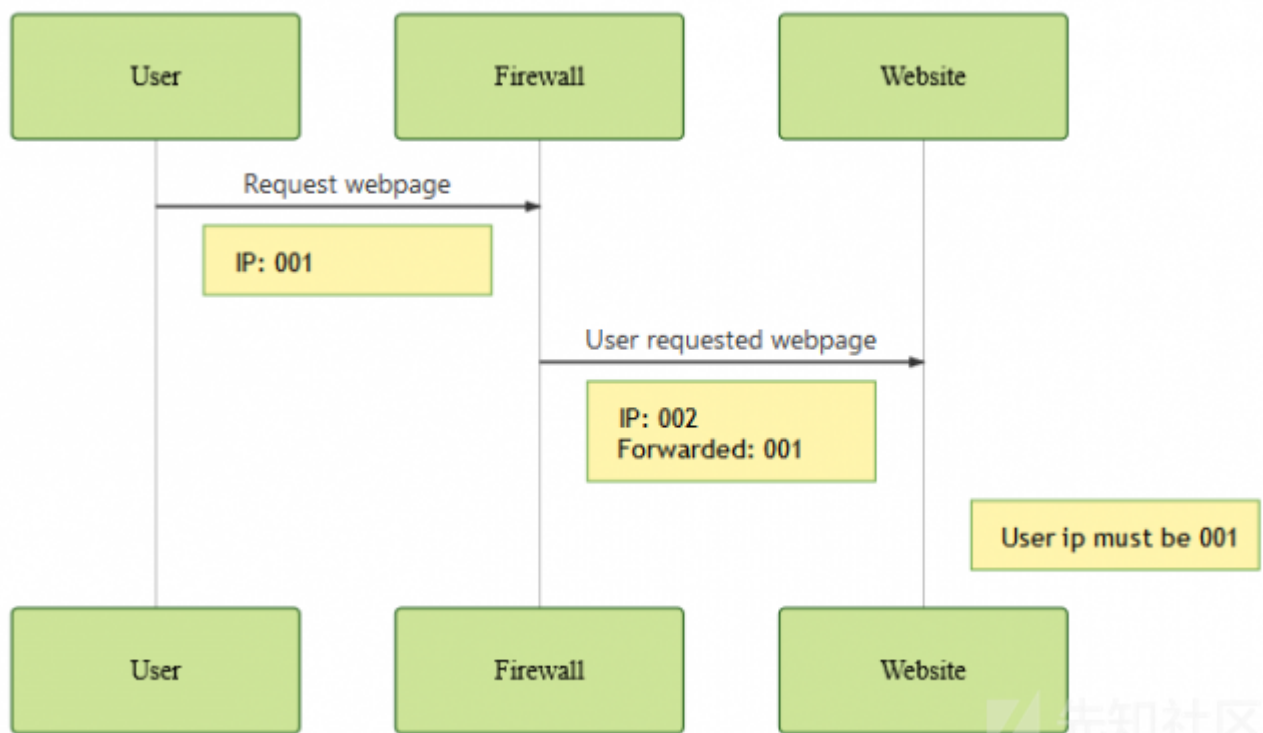
当使用网站防火墙时，事情就会变得有些棘手；要么优化我们的性能，要么保护我们自己免受攻击。

由于用户的请求在到达网站前，已经通过了防火墙，因此网站无法利用连接地址来找出实际发起请求的IP。



忽略原始用户IP，使用防火

为了解决这种情况，防火墙通常会在HTTP header中自动添加用户原始IP。



防火墙会转送请求访问的

这使得网站能够正确识别原始用户以及其相应的IP。

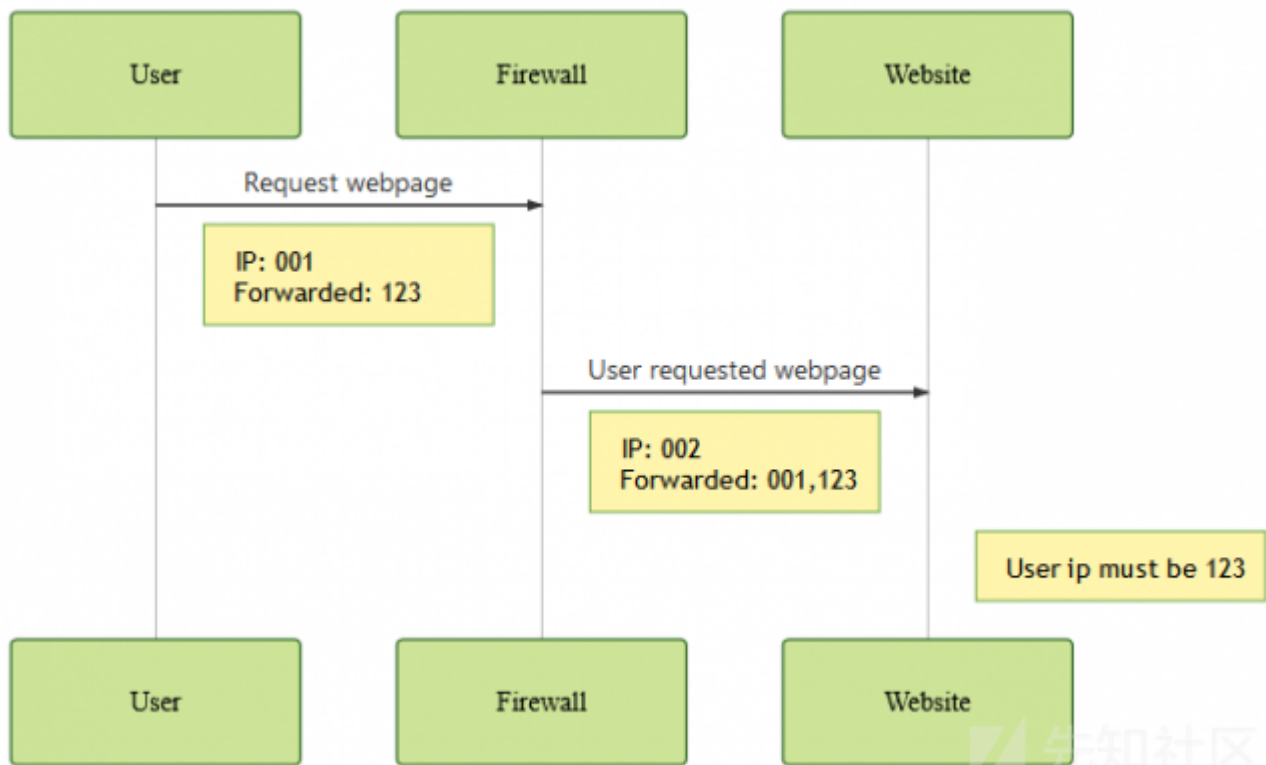
多层防火墙

当攻击者自己构造forwarded

IP（即使这个IP并不存在）时，可能会使服务器弄错原始IP。（译者注：这里的意思是，攻击者在自己发出的原始报文中，加入了一个forwarded IP，例如下图中，加入一个Forwarded:

123,当最终website读取用户IP时，会误认为123是原始IP，即使123可能不存在）使用多层防火墙时也会有这种情况出现，因为每一层都会在现有的基础上再添加一个IP地址。

这完全取决于防火墙的配置方式，以及它如何处理现有的转送数据。



IP，混淆服务器

大多数防火墙都会将用户的真实IP加入HTTP header自定义字段里，例如X_SUCURI_CLIENTIP（译者注：SUCURI是这篇文章的公司的名字，这个字段是他家防火墙header里的存放用户的真实IP的自定义字段）。

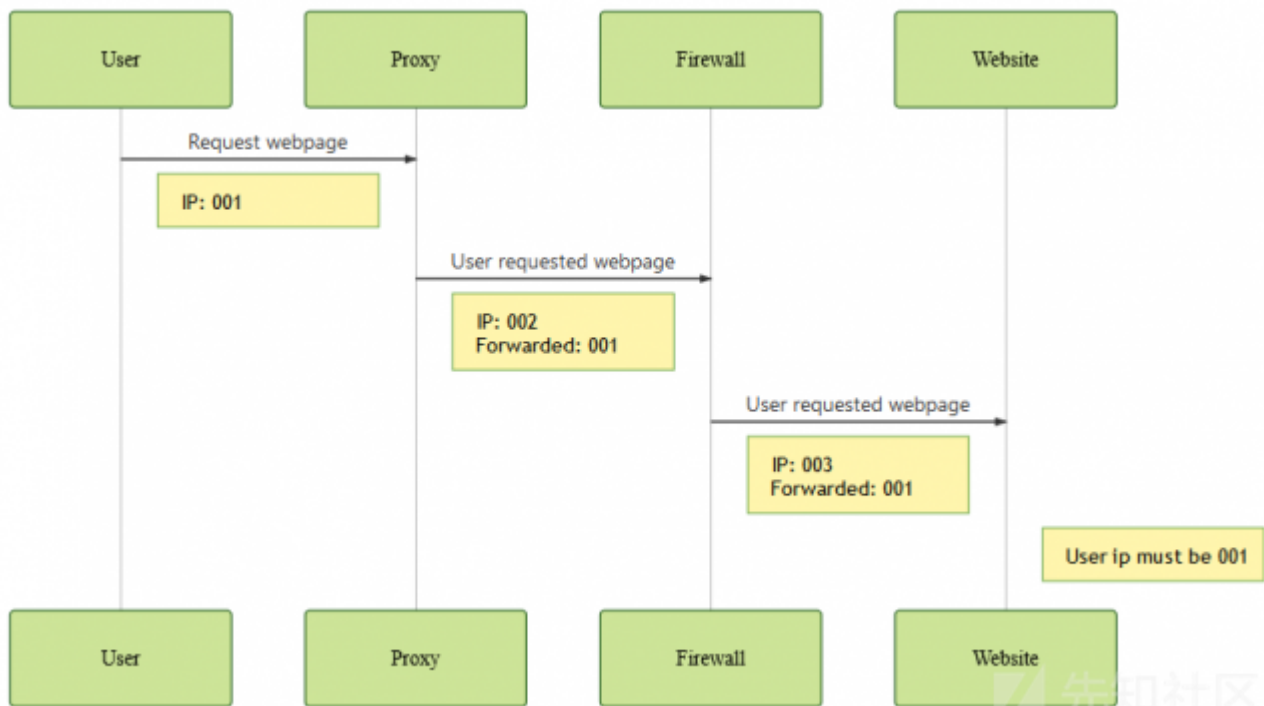
此HTTP header里始终含有用户原始IP信息，因为它应该永远不会存在于防火墙之外的环境里。因此，即使存在也会被覆盖（译者注：即使攻击者在自己的报文里构造了X_SUCURI_CLIENTIP，也会被覆盖）。

在您有多个代理的情况下，您可能希望获取用户真实IP，而不是最后进行连接的IP。在这种情况下，您可以使用可包含多个IP地址的X-forwarded-for。

如果X-forwarded-for在连接防火墙之前已具有IP值，则将执行以下某一项操作：

- 将用户的IP附加到任意现有IP列表中。
- 保持原样；不添加或替换值。
- 清除header；删除所有值。
- 使用用户的IP覆盖header。

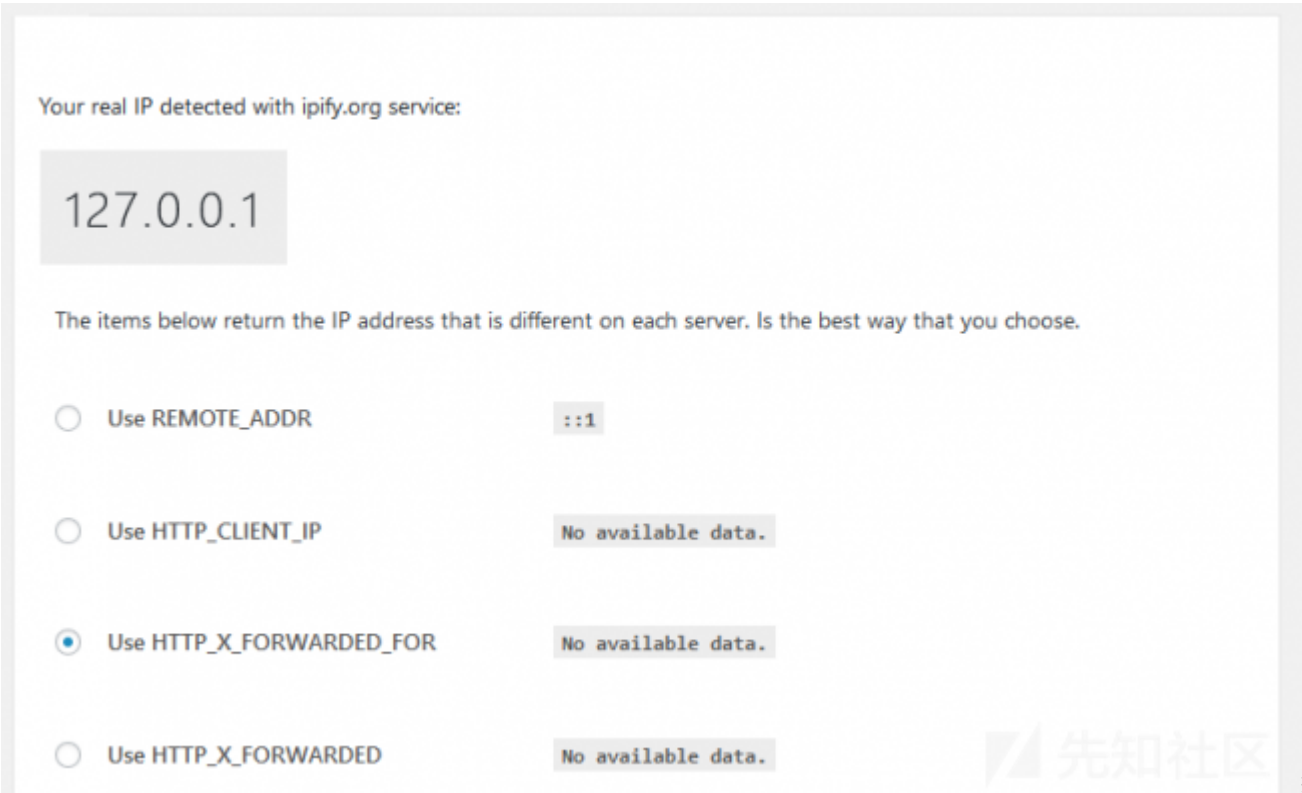
这些结果可能取决于所使用的防火墙和它的配置的不同而不同，最终保证多级防火墙来正确地发送所述用户信息给服务器。



漏洞分析

该插件的漏洞是由于不过滤或验证用户IP所产生的。

只有当插件使用header来识别访问者的IP地址（例如，不是REMOTE_ADDR）时，才能触发利用：



该漏洞利用触发还必须满足以下两个条件之一：

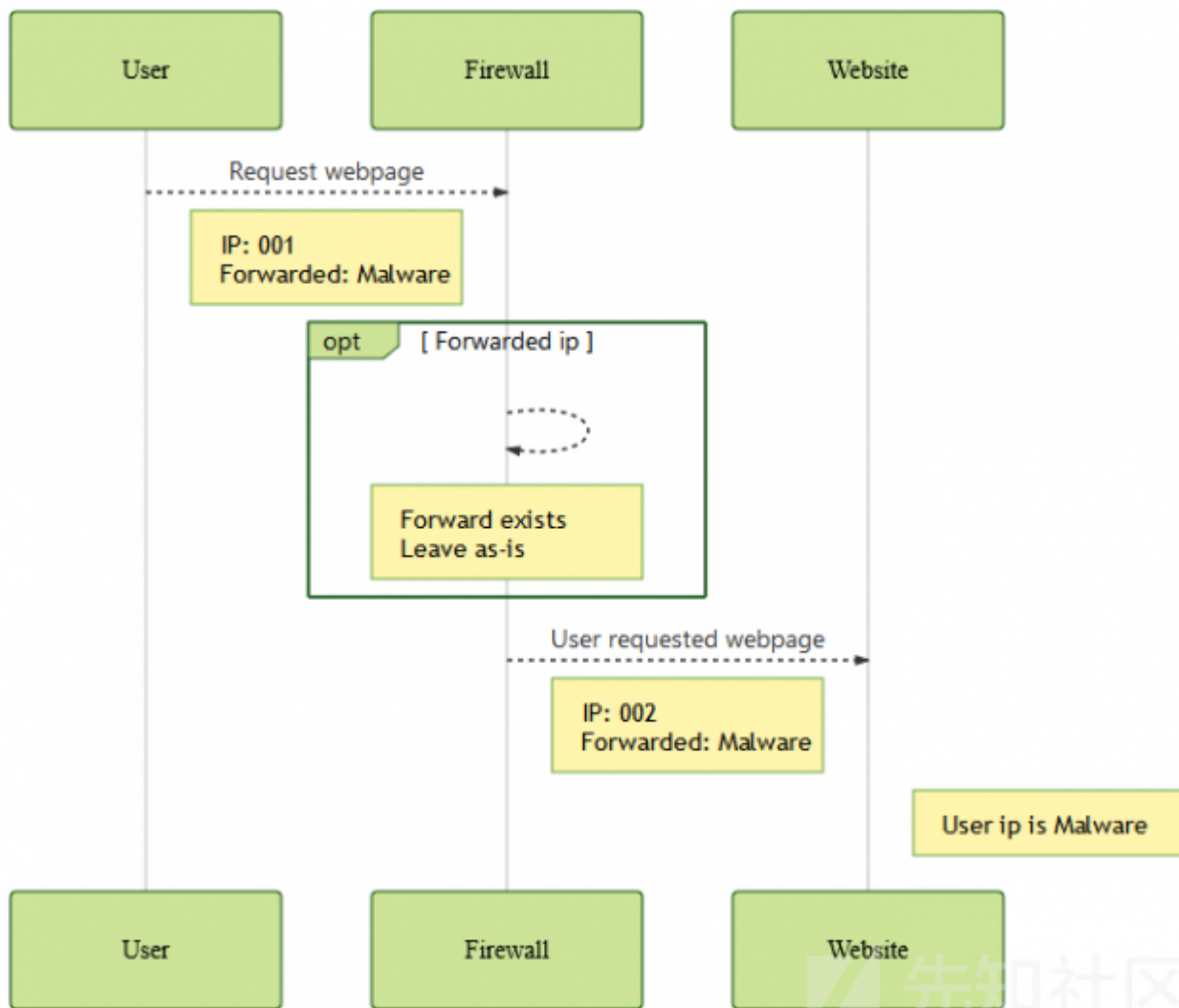
- 防火墙必须是可绕过的。

这意味着必须将网站配置为接受来自于所有人的连接，而不仅仅是接受使用防火墙端口转发的连接。

(译者注：这里作者“防火墙必须是可绕过”的意思是，攻击者的请求，可以不经过防火墙，直接发送至wordpress站点)

或者

- 防火墙必须原样保留header，如果其存在的话。



IP不变

以上两个利用条件的共同点是forwarded值完全由攻击者控制。

启用WAF使您不会容轻易遭受攻击（请[参阅此处以获取有关如何防止防火墙被绕过的说明](#)），除非攻击者可以绕过WAF；或者防火墙被配置为保持IP不变。

如果您以前使用过WAF，但是在未更新插件设置的情况下停止运行，则可能会受到攻击。

由于header可以包含多个IP地址，具体取决于防火墙的数量及其配置，因此插件将首先取出完整的header IP列表值，然后在提供多个地址的情况下遍历IP列表，依次向右边的寻找并将IP值替换为有效地址(译者注：原理可以看下面那个代码段的图)。

在这两种易受攻击的配置中，IP变量完全由攻击者控制。这使得他们可以将恶意JavaScript代码作为自己的IP注入，并将这些代码存储在管理页面上并执行。

技术细节

该插件使用class-wp-statistics.php文件中的get_IP方法。

```

// Get User Set $_SERVER HEADER
$ip_method = self::getIPMethod();

// Get User IP
if ( isset( $_SERVER[ $ip_method ] ) ) {
    $this->ip = $_SERVER[ $ip_method ];
}

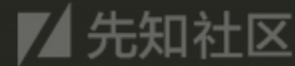
/**
 * This Filter Used For Custom $_SERVER String
 */
$user_ip = apply_filters( 'wp_statistics_sanitize_user_ip', $this->ip );

// Check If X_FORWARDED_FOR
foreach ( explode( ',', $user_ip ) as $ip ) {
    $ip = trim( $ip );
    if ( filter_var( $ip, FILTER_VALIDATE_IP, FILTER_FLAG_NO_PRIV_RANGE | FILTER_FLAG_NO_RES_RANGE ) !== false ) {
        $user_ip = $ip;
    }
}

// If no valid ip address has been found, use 127.0.0.1 (aka localhost).
if ( false === $user_ip ) {
    $this->ip = '127.0.0.1';
} else {
    $this->ip = $user_ip;
}

return $this->ip;

```



IP变量的默认值为设置中提供的header，默认情况下为REMOTE_ADDR（译者注：默认配置情况下不存在漏洞，具体配置见上文插件的设置页面）。如果有多个以逗号分隔

由于IP地址的默认值是header的值，并且未使用FILTER_VALIDATE_IP方法进行清理或验证，因此当header中没有多个IP地址时，它将按原样存储。

在top visitor，online users与最近访问者等模块中，访问者IP将作为页面的一部分输出，它是插件渲染页面的一部分。

（译者注：我对于上述的理解是，插件设置页面提供了一个用户自己选择获取ip的方式“the items below return the ip address that is different on each server. Is the best way that you choose”，如果用户选的是X_Forwarded_For方式，\$user_ip里获得的是X_Forwarded_For中的值，攻击者在X_Forwarded_For中传入xss payload，由于不是合法的ip格式，在foreach里的filter_var结果为false，因此不会进入if中执行\$user_ip=\$ip语句，\$user_ip中值仍然是xss payload，最终，在代码最后一个else中赋值给\$this->ip并return）

结论

某种类型的信息看起来可能是安全的（像是访问者的IP地址），但实际上与预期相悖。由于开发人员的某些臆断，使得攻击者可以在在管理页面上注入恶意代码，从而导致整

为了不受此漏洞影响，我们强烈建议用户尽快将插件更新到12.6.7版。

原文地址：<https://blog.sucuri.net/2019/07/wordpress-plugin-wp-statistics-unauthenticated-stored-xss-under-certain-configurations.html>

写在翻译稿后面

我跟踪了下wp-statistics的修复

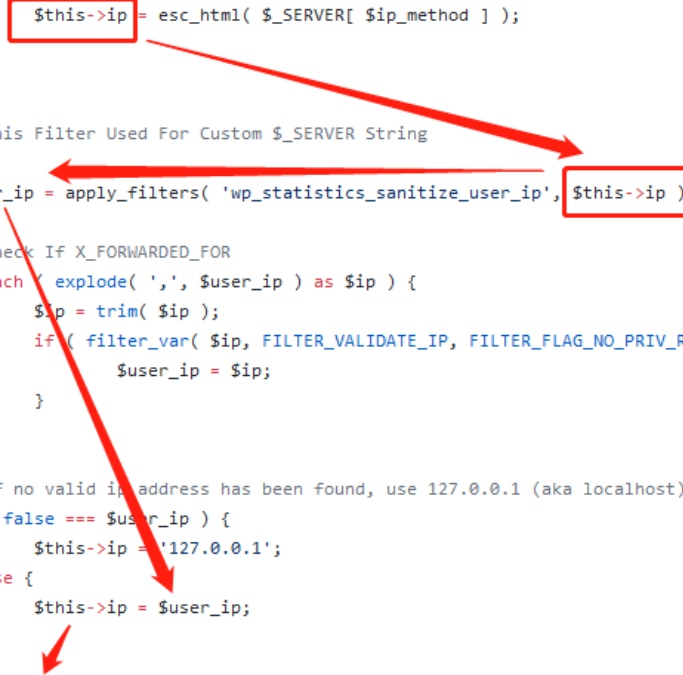
814	@@ -814,7 +817,7 @@ public function get_IP() {	817	
815	// Get User IP	818	// Get User IP
816	if (isset(\$_SERVER[\$ip_method])) {	819	if (isset(\$_SERVER[\$ip_method])) {
817	- \$this->ip = sanitize_text_field(\$_SERVER[\$ip_method]);	820	+ \$this->ip = esc_html(\$_SERVER[\$ip_method]);
818	}	821	}
819		822	
820	/**	823	/**



使用esc_html对获取到的IP进行过滤

虽然现在仍然可以在header里传payload，但是由于esc_html的转义，到页面渲染的时候，payload已经失效，流程见下图

```
817     if ( isset( $_SERVER[ $ip_method ] ) ) {
818         $this->ip = esc_html( $_SERVER[ $ip_method ] );
819     }
820
821     /**
822      * This Filter Used For Custom $_SERVER String
823      */
824     $user_ip = apply_filters( 'wp_statistics_sanitize_user_ip', $this->ip );
825
826     // Check If X_FORWARDED_FOR
827     foreach ( explode( ',', $user_ip ) as $ip ) {
828         $ip = trim( $ip );
829         if ( filter_var( $ip, FILTER_VALIDATE_IP, FILTER_FLAG_NO_PRIV_RANGE | FILTER_FLAG_NO_RES_RANGE ) !== false
830             $user_ip = $ip;
831         }
832     }
833
834     // If no valid ip address has been found, use 127.0.0.1 (aka localhost).
835     if ( false === $user_ip ) {
836         $this->ip = '127.0.0.1';
837     } else {
838         $this->ip = $user_ip;
839     }
840
841     return $this->ip;
842 }
```



点击收藏 | 0 关注 | 1

[上一篇：MySQL 客户端攻击（抓包分析，...）](#) [下一篇：Badusb初识](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)