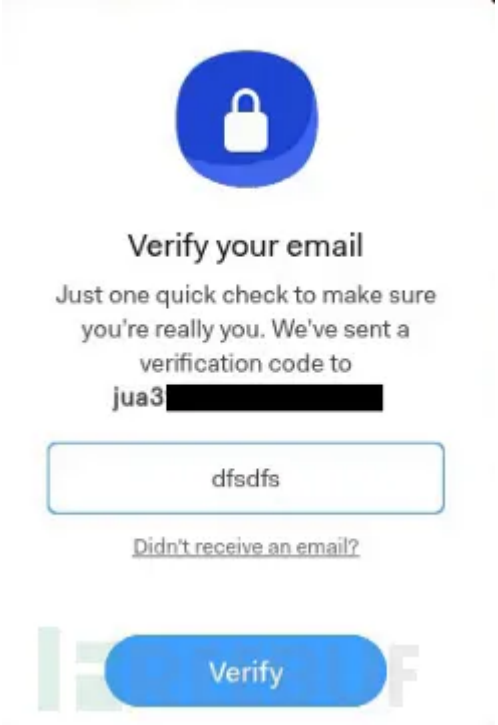
一次性验证密码（OTP）的简单绕过

clouds FreeBuf 今天

今天分享的是作者在众测过程中实现的一次性验证密码（OTP）绕过技巧，通过拦截修改响应中的内容即可有效绕过OTP，姿势非常简单，但也值得学习借鉴，一起来看看。

漏洞发现

假设目标网站为example.com，当我在其中创建了用户账号之后，我的注册邮箱中就收到了一个一次性验证密码（OTP），该OTP目的是通过验证邮箱来确认我的身份。



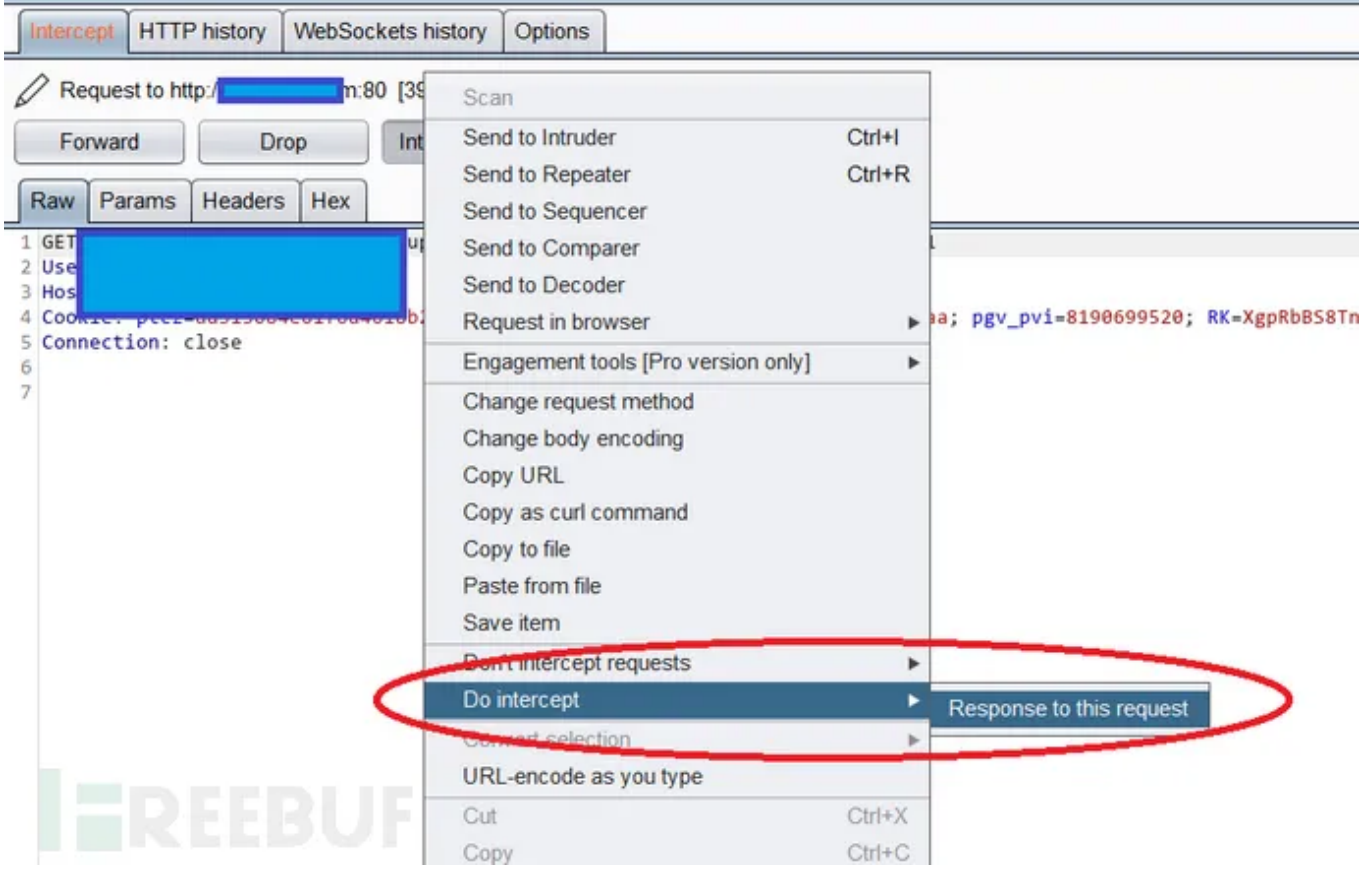
开启Burp抓包后，我输入了正确的OTP后，请求的响应简洁明了，其中包含一个简单的消息头’HTTP/1.1 200 Created’和一个大括号{} 的消息体。此时我想到了来尝试绕过这种OTP机制。

漏洞复现

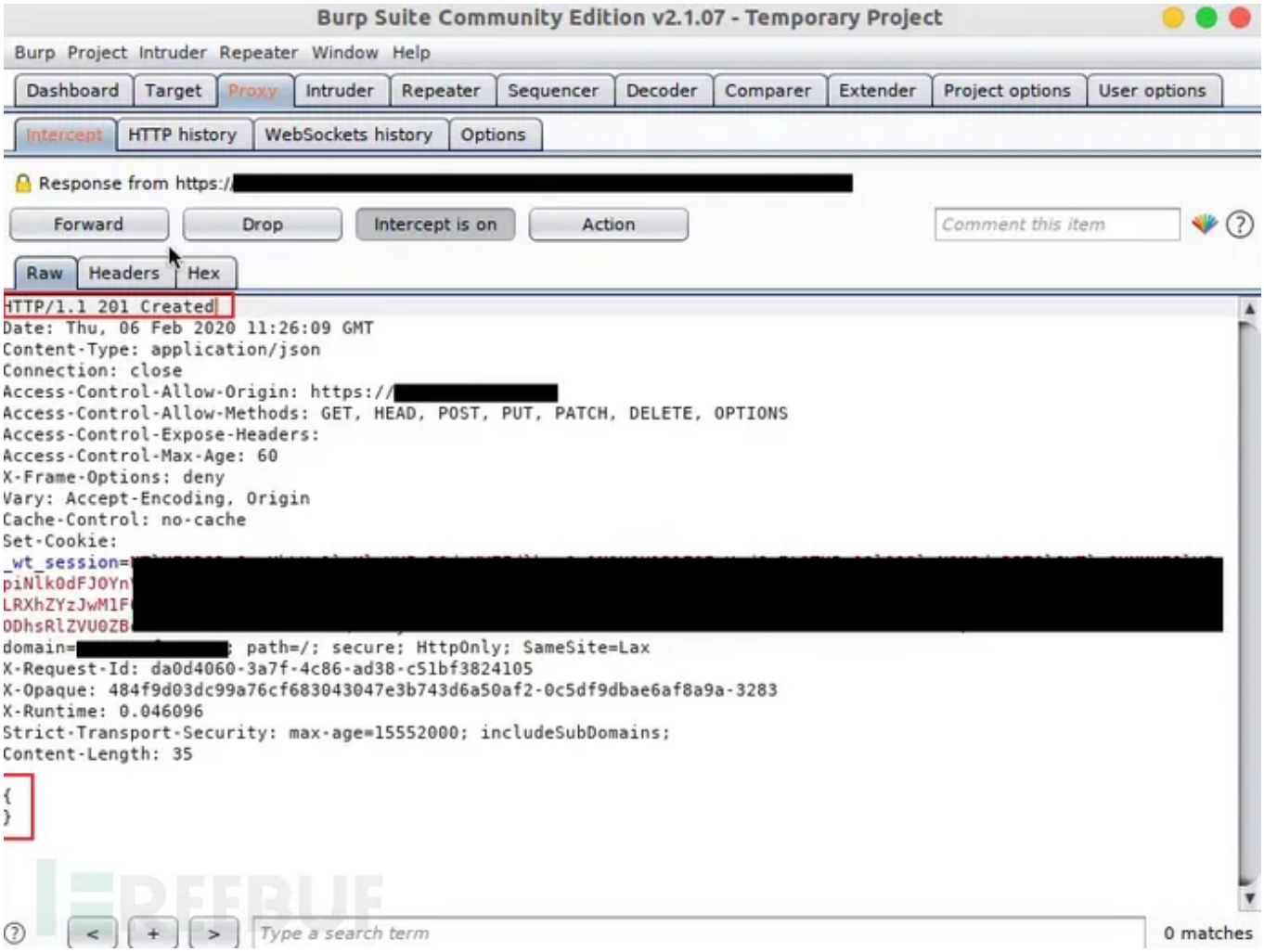
1、使用邮箱abc123@gmail.com创建账户；

2、之后，邮箱abc123@gmail.com会收到一个OTP验证密码；

3、把该OTP复制到验证区域，对用户身份进行验证。此时开启Burp抓包，在当前请求场景下，通过右键-Do Intercept >Response To This Request设置拦截响应包：



然后，我们拦截获得了正确OTP验证的响应包如下：



4、完成一次正确的OTP验证操作；

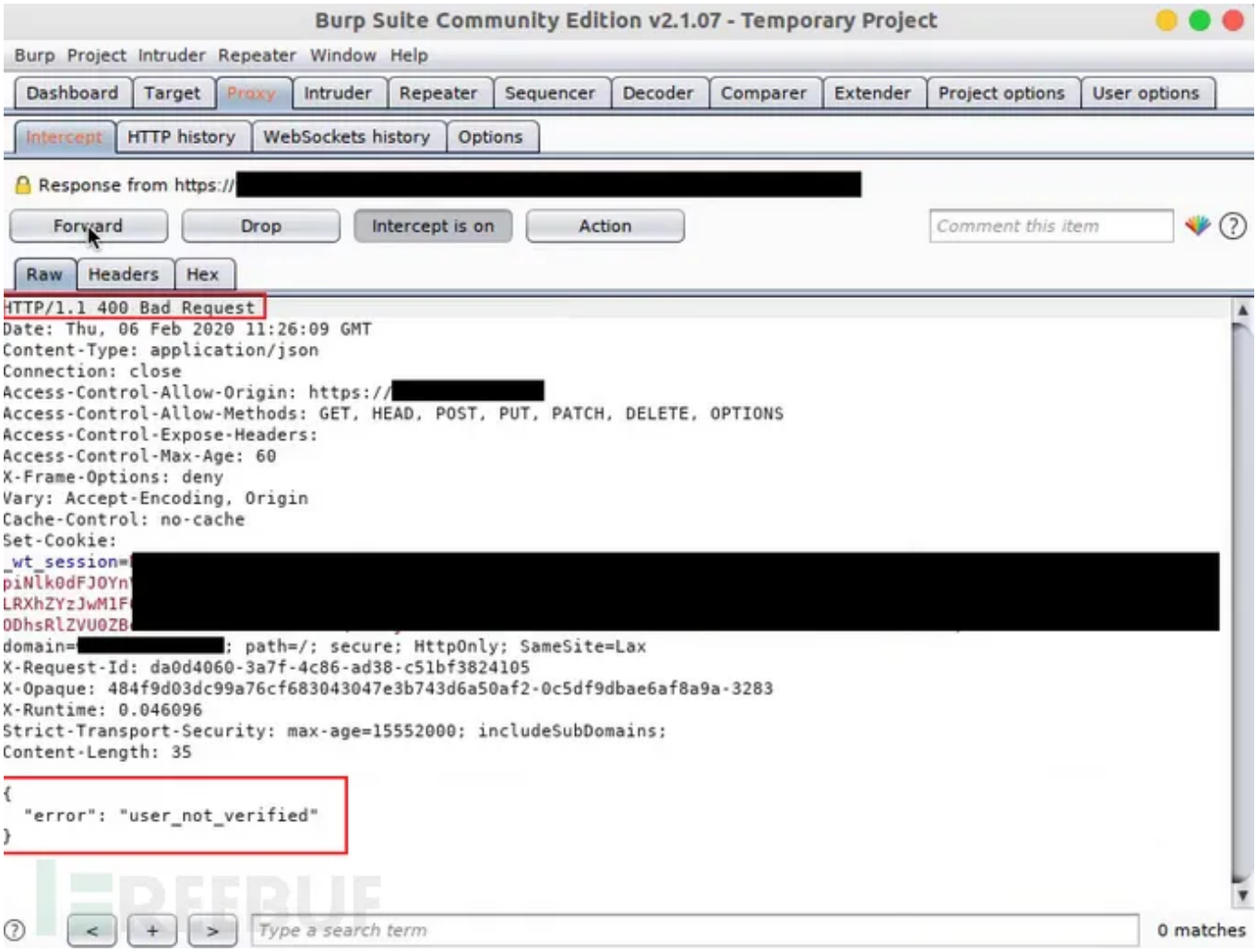
5、现在，用受害者邮箱victim123@gmail.com进行账户创建；

6、现在，可以肯定，目标网站会向受害者邮箱victim123@gmail.com发送了一个OTP验证码；

7、但是，因为我没有受害者邮箱victim123@gmail.com的登录权限，就只有尝试绕过了；

8、我们在目标网站的OTP验证区域随意输入一串OTP验证码；

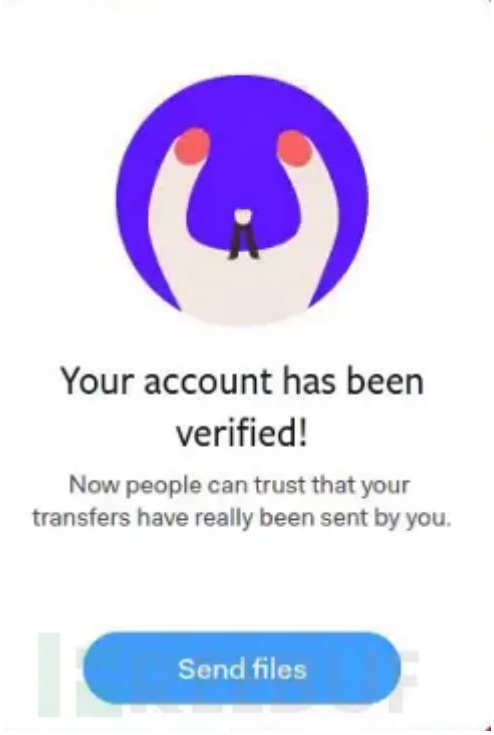
9、从Burp的抓包中，我们获得了上个步骤随意输入OTP验证的请求，然后同样按照Do Intercept >Response To This Request设置拦截获取响应包，如下：



10、可以看到，响应包提示验证失败了，其中的消息头和消息体为’HTTP/1.1 400 Bad Request’ 和{“error”: “user_not_verified”}；

11、现在，我们把响应包中的消息头和消息体分别替换为：’HTTP/1.1 200 Created’ 和{}，然后点击响应转发“Forward”；

12、接下来，奇迹出现了，目标网站的OTP验证区域提示“账户身份验证成功”！



OTP就这样被绕过了！

漏洞上报和处理进程

> 2020.2.5 漏洞初报
>
> 2020.2.6 漏洞分类
>
> 漏洞奖励：€ xxx

*参考来源：medium，clouds 编译整理，转载请注明来自 FreeBuf.COM

小程序

精彩推荐

阅读原文