10.12. 跨站请求伪造(10.12. Cross-Site Request Forgery)

10.12. 跨站请求伪造

跨站请求伪造（CSRF）是一种漏洞利用，攻击者致使受害的最终用户按恶意URI（例如以误导的链接、图片或重定向提供给用户代理）到达受信任的服务器（通常由存在有效的会话Cookie而建立）。

针对客户端的重定向URI的CSRF攻击允许攻击者注入自己的授权码或访问令牌，这将导致在客户端中使用与攻击者的受保护资源关联的访问令牌而非受害者的（例如，保存受害者的银行账户信息到攻击者控制的受保护资源）。

客户端必须为它的重定向URI实现CSRF保护。这通常通过要求向重定向URI端点发送的任何请求包含该请求对用户代理身份认证状态的绑定值（例如，用于对用户代理进行身份验证的会话Cookie的哈希值）来实现。

一旦从最终用户获得授权，授权[illegible]节最终用户用户代理[illegible]请求包含[illegible]如[illegible]cookie[illegible]中[illegible]HTML5本地存储[illegible]通过该绑定值与用户代理的身份验证状态[illegible]的[illegible]确保[illegible]请求的有效性。用于CSRF

针对授权服务器的授权端点的CSRF攻击可能导致攻击者获得最终用户为恶意客户端的授权而不牵涉或警告最终用户。

授权服务器必须为它的授权端点实现CSRF保护并且确保在资源所有者未意识到且无显式同意时恶意客户端不能获得授权。

10.12. Cross-Site Request Forgery

```
   Cross-site request forgery (CSRF) is an exploit in which an attacker   causes the user-agent
of a victim end-user to follow a malicious URI   (e.g., provided to the user-agent as a
misleading link, image, or
   redirection) to a trusting server (usually established via the   presence of a valid session
cookie).
   A CSRF attack against the client's redirection URI allows an attacker   to inject its own
authorization code or access token, which can   result in the client using an access token
associated with the
   attacker's protected resources rather than the victim's (e.g., save   the victim's bank
account information to a protected resource
```

controlled by the attacker).

The client MUST implement CSRF protection for its redirection URI. This is typically accomplished by requiring any request sent to the redirection URI endpoint to include a value that binds the request to

the user-agent's authenticated state (e.g., a hash of the session cookie used to authenticate the user-agent). The client SHOULD utilize the "state" request parameter to deliver this value to the

authorization server when making an authorization request.

Once authorization has been obtained from the end-user, the authorization server redirects the end-user's user-agent back to the client with the required binding value contained in the "state"

parameter. The binding value enables the client to verify the validity of the request by matching the binding value to the user-agent's authenticated state. The binding value used for CSRF

protection MUST contain a non-guessable value (as described in Section 10.10), and the user-agent's authenticated state (e.g., session cookie, HTML5 local storage) MUST be kept in a location

accessible only to the client and the user-agent (i.e., protected by

same-origin policy).

A CSRF attack against the authorization server's authorization endpoint can result in an attacker obtaining end-user authorization for a malicious client without involving or alerting the end-user.

The authorization server MUST implement CSRF protection for its authorization endpoint and ensure that a malicious client cannot obtain authorization without the awareness and explicit consent of

the resource owner.

Revision #1
Created Wed, Mar 25, 2020 11:26 PM by □□□
Updated Wed, Mar 25, 2020 11:26 PM by □□□