

# 10.16. [redacted](10.16. Misuse of Access Token to Impersonate Resource Owner in Implicit Flow)

## 10.16. [redacted]

[redacted] [redacted] [redacted]

[redacted]response\_type=token[redacted]

[redacted]

## 10.16. Misuse of Access Token to Impersonate Resource Owner in Implicit Flow

For public clients using implicit flows, this specification does not provide any method for the client to determine what client an access token was issued to.

A resource owner may willingly delegate access to a resource by granting an access token to an attacker's malicious client. This may be due to phishing or some other pretext. An attacker may also steal

a token via some other mechanism. An attacker may then attempt to impersonate the resource owner by providing the access token to a legitimate public client.

In the implicit flow (`response_type=token`), the attacker can easily switch the token in the response from the authorization server, replacing the real access token with the one previously issued to the attacker.

Servers communicating with native applications that rely on being passed an access token in the back channel to identify the user of the client may be similarly compromised by an attacker creating a

compromised application that can inject arbitrary stolen access tokens.

Any public client that makes the assumption that only the resource owner can present it with a valid access token for the resource is vulnerable to this type of attack.

This type of attack may expose information about the resource owner at the legitimate client to the attacker (malicious client). This will also allow the attacker to perform operations at the legitimate client with the same permissions as the resource owner who originally granted the access token or authorization code.

Authenticating resource owners to clients is out of scope for this specification. Any specification that uses the authorization process as a form of delegated end-user authentication to the client (e.g., third-party sign-in service) MUST NOT use the implicit flow without

additional security mechanisms that would enable the client to determine if the access token was issued for its use (e.g., audience-restricting the access token).