Building the OAuth Client

Justin Boyer
OWNER, GREEN MACHINE SECURITY

https://justinboyerwriter.com

The Client's Role

Ask for Permission

Redirect the resource owner to the authorization server to ask permission to access resources

Access the Protected Resource

Use the token from the authorization server to gain access to the protected resource

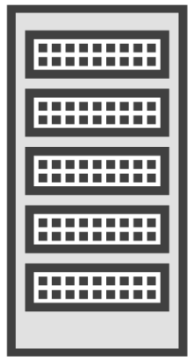
Where the Client Fits

Globomantics employee (resource owner)

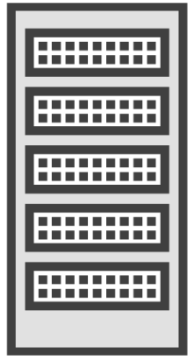
Globomantics client

Employee authenticates to Carved Rock Fitness and authorizes Globomantics to receive their data

Globomantics client app redirects user to Carved Rock Fitness authorization server

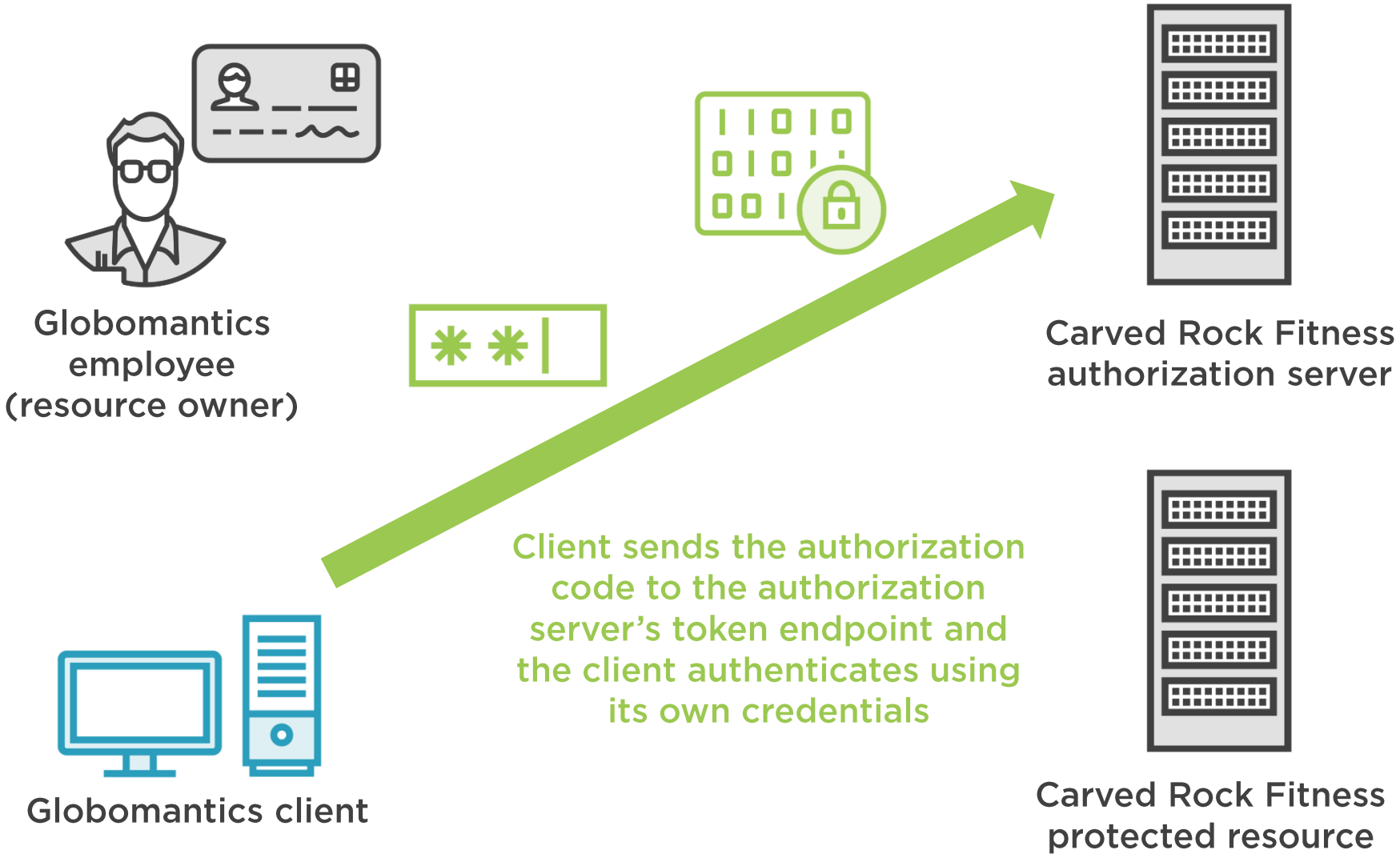


Carved Rock Fitness authorization server



Carved Rock Fitness protected resource

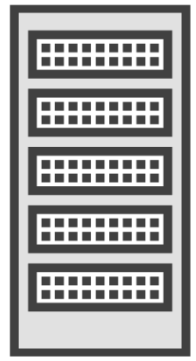
Where the Client Fits



Where the Client Fits

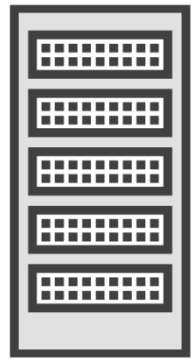
Globomantics employee (resource owner)

Globomantics' HR system can now access the employees' workout data from the Carved Rock Fitness database



Carved Rock Fitness authorization server

Globomantics client



Carved Rock Fitness protected resource

Common Client Vulnerabilities

Cross-site Request Forgery (CSRF)

Malicious application forges a request to a website where the user is authenticated

Attacker receives or forges an authorization code and tricks victim's browser into sending it to the client's callback URI

Fix: Use state parameter to add random value to every OAuth transaction

Theft of Client Credentials

When using static registration, poorly protected credentials can be stolen by an attacker

Fix: Keep the credentials secure with a credential management system (ex. Vault)

Fix: Use dynamic client registration

Incorrect Registration of the Redirect URI

Redirect URI must be as specific as possible

Using only the domain and matching on subdomains opens up your client for an attack

- Stealing authorization code from referrer header
- Stealing the token through an open redirect

Fix: Use exact match only to validate redirect uris

Bad example: https://youroauthclient.com

Good example: https://youroauthclient.com/oauth/oauthprovider/callback

Place the Token in a URL Parameter

OAuth allows tokens to be placed in the URL parameter

URL parameter is an insecure place for the token

- Logged in server logs
- Exposed in the referrer header
- Could be copy/pasted into a forum

Fix: Use the authorization header to pass a bearer token

What Have We Learned?

The client asks for permission and accesses a resource on the user's behalf

Implementing a client in Node.js

Common Client Vulnerabilities

- CSRF
- Theft of client credentials
- Incorrect redirect URI registration
- Token in a URL parameter

You have the tools, now build your client