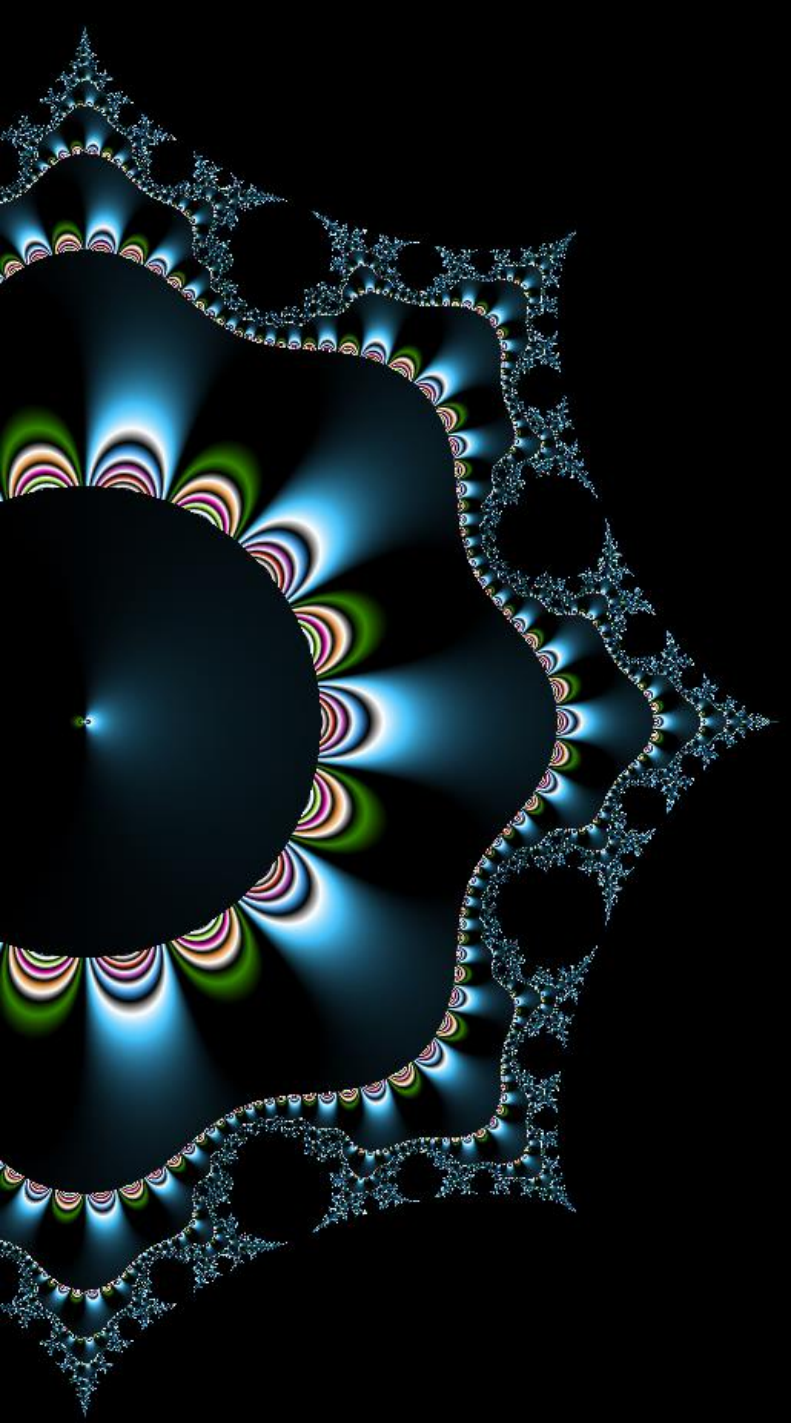


18 JS & XSS

Protections

Thibaut HENIN

www.arsouyes.org

Rappel du problème

Un attaquant inclus du contenu HTML/JS

Injection de contenu malveillant

Côté code

Comment bien coder ?

En PHP

Échapper / supprimer les *tags*

`Htmlspecialchars, Htmlentities, strip_tags`

Encoder les attributs

`Urlencode`

En JS

Échapper / supprimer les *tags*

`Dépend des frameworks`

Utiliser <template>

`textContent remplace innerHtml`

Cookies

Quelques attributs pour protéger les données

Ne pas utiliser les cookies

Ou alors…

Expires

(durée de validité)

Domain

(domaine de validité)

Path

(chemin de validité)

Secure

(ne transmettre que si TLS)

HttpOnly

(vers serveur uniquement)

SameSite

(depuis même site uniquement)

SOP

Same-Origin-Policy

Same Origin

Deux ressources ont la même origine si…

Même protocole

(http, https, ftp, …)

Même nom de domaine

Même port

(80, 443, 8080, 8443, …)

Politique pour les autres origines
Principalement sur XmlHttpRequest()

Ne peut pas accéder aux autres contenus

Mais ils peuvent être embarqués dans le HTML

(scripts, img, video, forms, …)

On peut quand même faire des requêtes

(GET et POST)

CORS

Cross Origin Ressource Sharing

Principe

Gérer plus finement les demandes extérieures

Nouvelles en-têtes HTTP

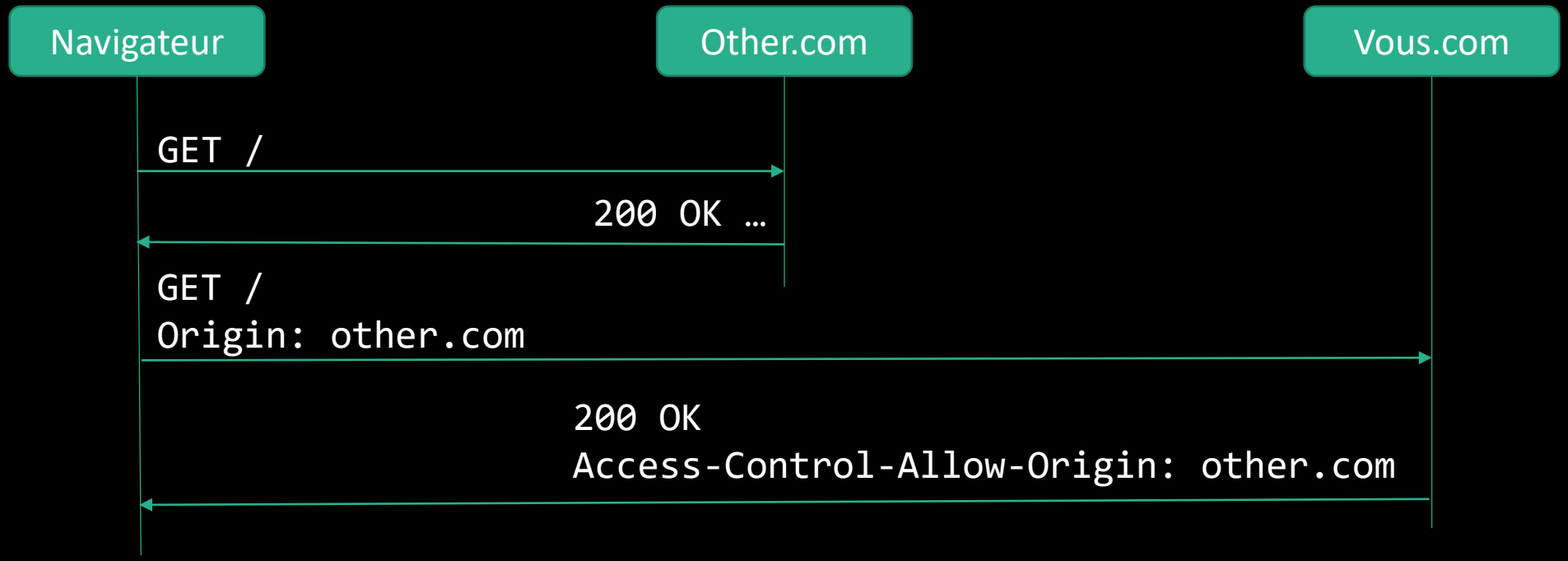
Navigateur demande les droits

`(Origin, Access-Control-Request-Method)`

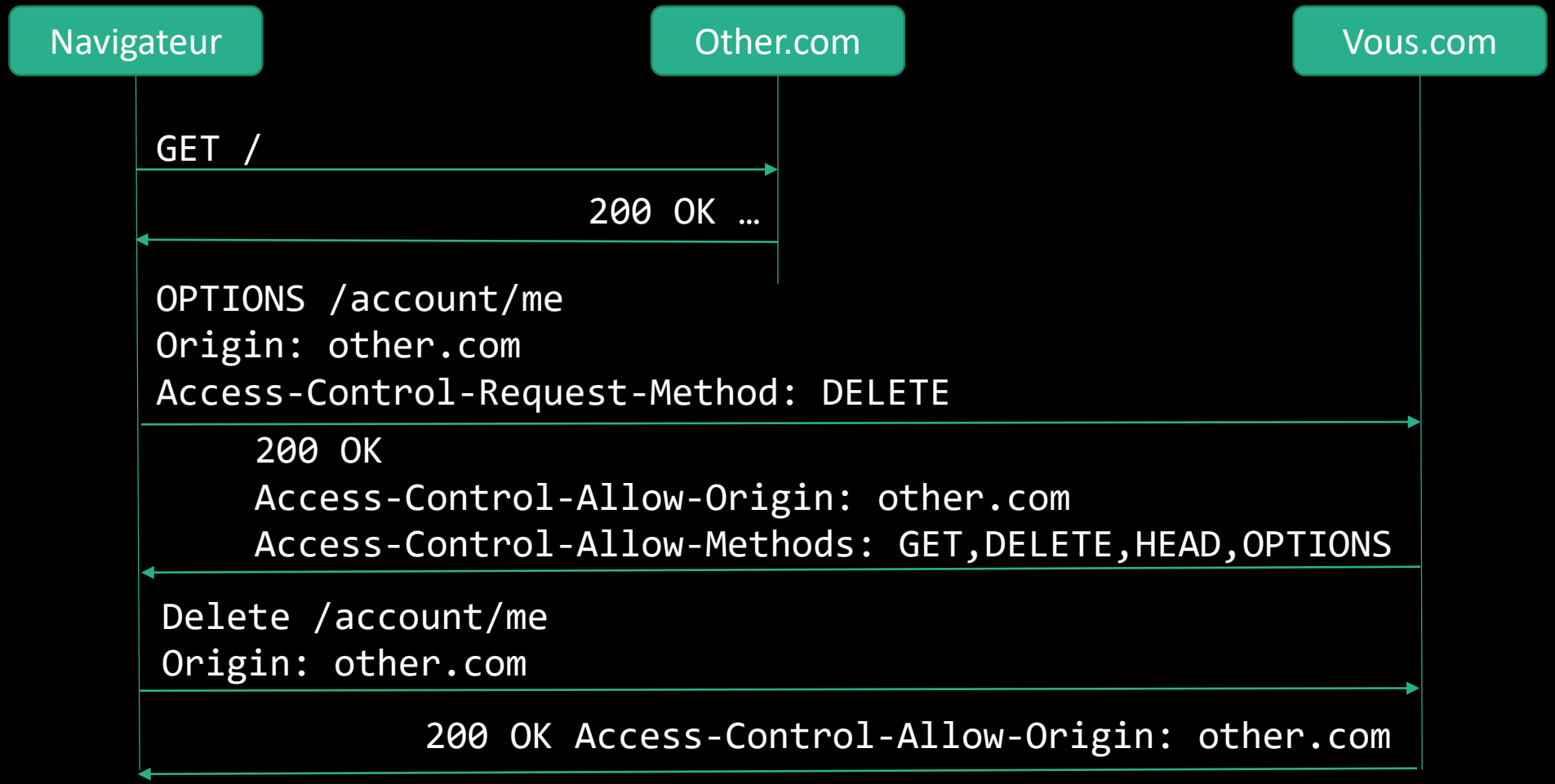
Serveur vérifie / spécifie les droits

`(Access-Control-Allow-Origin, Access-Control-Allow-Methods)`

Requête simple
(GET, POST, HEAD + content type)



Requête « preflight »
(le reste)



CSP

Content Security Policy

Principe : entête

Le serveur explicite la politique

En-tête HTTP

(Content-Security-Policy)

En-tête HTML

(meta, Content-Security-Policy)

Principe : règles

Restriction des origines utilisables

Catégorie de contenus

(default-src, script-src, style-src, …)

Origines autorisées

(‘self’, domaine, protocole+domaine)

Principe : rapports

Restriction des origines utilisables

Une URL

(pour recevoir les rapports JSON des navigateurs)

Un mode « rapport uniquement »

(Pour tester avant mise en production définitive)

Anti CSRF

Techniques disponibles

CSRF Token

Serveur génère une valeur aléatoire

(unique pour chaque session)

Placée dans le formulaire

`<input type=hidden>`

Vérifiée lors du submit

Double submit

Idem mais…

Cookie remplace la session

Variantes

(cookie chiffré ou signature)

Re-authentification

Demande du mot de passe

(pour les requêtes importantes uniquement)

Captcha

Test de Turing

(pénible pour les humains)