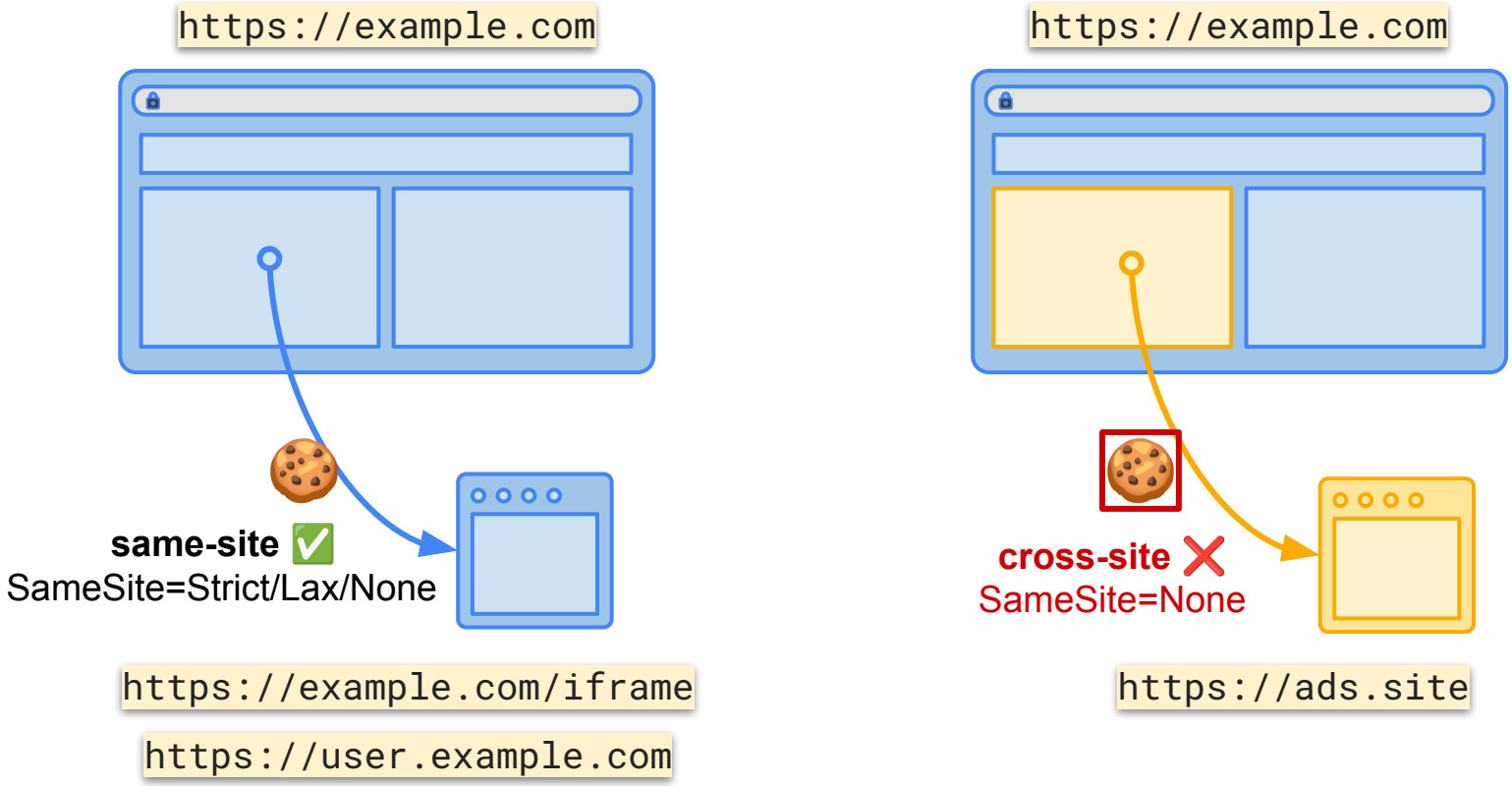
Allow `SameSite=None` Cookies in Sandboxed Contexts

Anusha Muley, Dylan Cutler

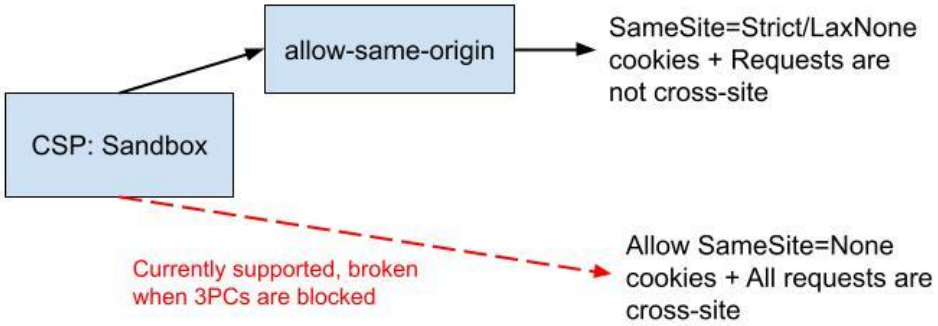
same-site Cookies

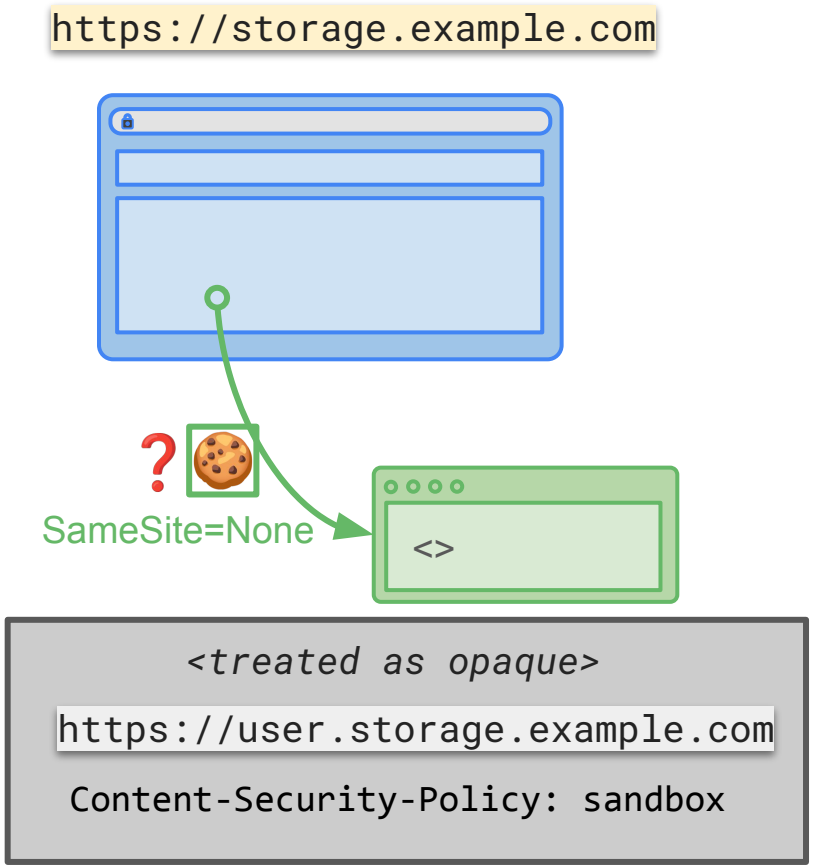


Sandboxing

```
Content-Security-Policy: sandbox
<iframe src="." sandbox></iframe>
```

- Document uses opaque origin → restricts access to `SameSite` cookie storage
- Sites use `SameSite=None` cookies for session/access control





With 3PCs Blocked

✅ allowed by default

❌ blocked

Example Scenario

storage.example.com hosts untrusted user uploaded content (ie code)

- `Content-Security-Policy: sandbox` → restrict scripts
- `SameSite=None` cookies to restrict content to only be visible to uploader

If 3PCs are blocked

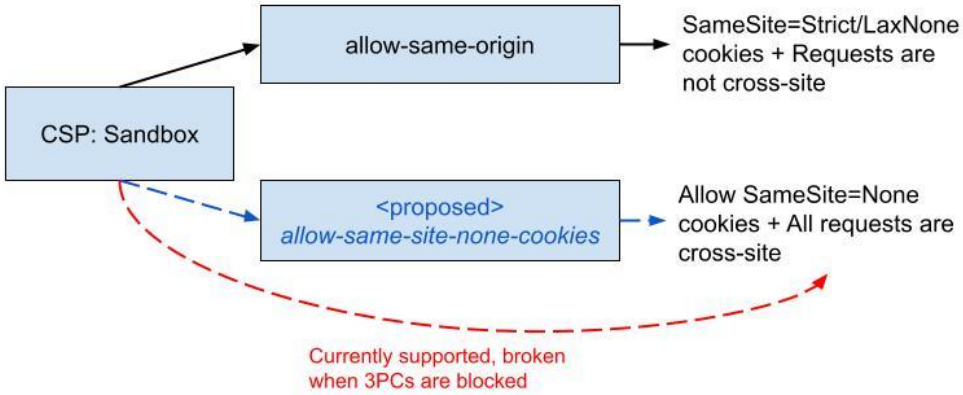
- Result: site cannot access `SameSite=None` cookies in sandboxed subresource pages

Proposal

```
Content-Security-Policy: sandbox allow-same-site-none-cookies;
```

```
<iframe src="." sandbox allow-same-site-none-cookies></iframe>
```

- Server opt-in to allow `SameSite=None` cookies in their sandboxed documents
- HTTP header → easy for developers to opt-in + accessible for non-scripts sandboxed contexts



Privacy and Security Considerations

- Opt-in behavior, servers can determine if vulnerability to re-enable
- Only allowing first party cookies:
  - Verify that frames are same-site with the sandboxed document
- Requests still treated as cross-site
  - CORS + `SameSite=Strict/Lax` filtering intact

Alternatives and Discussion

- SAA/another API based solution?
  - No allow-scripts (+ allow-storage-access-by-user-activation, allow-same-origin)
- 3PC exception for this case?
  - temporary fix, want opt-in functionality
- If 3PCs are not blocked, what would this directive do?
  - No impact if 3PCs aren’t blocked, frame + site specific so binary cases