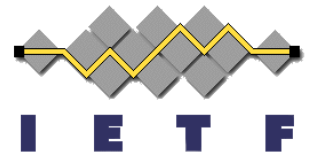


Frame-Options (FO) in IETF websec or move to W3C WebAppSec?



(draft-ietf-websec-frame-options-00)

Tobias Gondrom
November 2012

Intro

- We still have an open discussion on where to do FO?
 - In light of this the editors did not update the draft...
- (side-note: XFO is in WGLC but will still need some polishing)
- FO is easy, it basically specifies out some evolutionary improvements to XFO, mostly
 1. Allow-From Option (already partially in XFO)
 2. Consistent use of Origin determining sources

Frame-Options

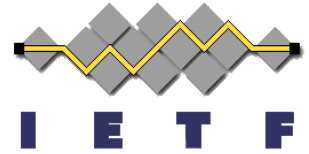
- Frame-Options

- In EBNF:

```
Frame-Options = "Frame-Options" ":" "DENY"/  
"SAMEORIGIN" / ("ALLOW-FROM" ":"URI)
```

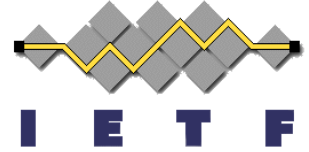
- **DENY**: The page cannot be displayed in a frame, regardless of the site attempting to do so.
- **SAMEORIGIN**: can only be displayed in a frame on the same origin as the page itself.
- **ALLOW-FROM**: can only be displayed in a frame on the specified origin

Reasons I heard to move FO to WebAppSec



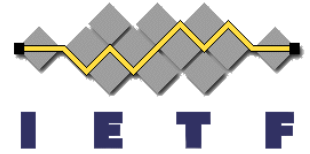
- Resources are available in webappsec
 - implementer types are in WebAppSec
 - making test cases
 - People in webappsec are paying attention to browser rendering engines not “protocol stuff”
- Synergy with CSP
 - having all this rendering policy stuff in one place spec-wise and wg wise is a benefit to everyone
- Chartered scope appropriateness
 - “FO is about presentation layer not protocol”
- Avoid “header bloat” if we include it in CSP

But....



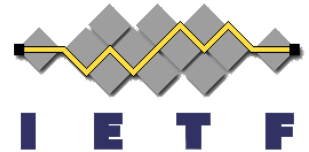
- Done some research on implications of FO as directive in CSP header and there is a big problem, because:
 - Allow-From SHOULD NOT list all URIs that are allowed to frame the resource (privacy and potentially very long URI lists)
- FO header generated dynamically per request
- No problem with one single FO http header, but probably conflicting with some CSP use cases:
 - caching
 - CSP using URI pointers for static CSP files
 - large CSP files generated dynamically

Frame-Options – Why keep it in WebSec?

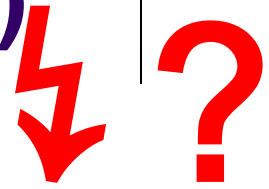


- FO is easy and probably close to done (?)
- Websec has access to resources we need to finish the draft, incl. browser people
- Synergy with other mechanisms is unclear?
- (on a side-note: FO without Allow-From mechanism would reduce it to XFO)

Options & Suggestions (am open to work either way)



1. Roll it into CSP as directive?



- We should solve the dynamic CSP question first
- OR decide the Allow-From is not dynamic per request

2. Roll it into a new CSP-safetyUI header?



- Better. Can we then reap the synergy?
- together with what? Does the other stuff fit into CSP?

3. Just review and finish it as stand-alone http header



- potentially add a report-only option (if needed?)
- do it in websec
- do it in WebAppSec (why move in that case?)