MIME SNIFFING ISSUES

Larry Masinter

IETF 82 Taipei

November 16, 2011

Using Tracker for issues

- Open issues:
  - http://trac.tools.ietf.org/wg/websec/trac/query?status=new&status=assigned&status=reopened&component=mime-sniff
- This talk a brief review of some of issues
- Please use tracker for new issues
  and to summarize results of discussions

#15: Scope of document

- Introductory rationale lists:
  - ***Web sites where HTTP content-type label doesn't seem to match author's intent***
- Document covers many other use cases:
  - ***Content delivered by other means than HTTP (ftp:, file: URIs)***
  - ***No HTTP content-type is supplied at all***
- In practice, sniffing is used also for other situations
  - ***email clients***
  - ***W3C Web Application packaging***

Algorithm inadequate for all use cases?

- *ftp uses file extension*
- *Sniffing of content-type for new MIME types*

#17 Use magic numbers in MIME registry

- Scope covers “no content-type supplied” cases
  - Need to be able to sniff new types
- Use “magic numbers” in IANA MIME registry?
  - Current registry content is haphazard
  - Would need to update registry or create a new one

#18 using file extensions

- File extensions are not used for HTTP
- BUT scope covers ftp, file:, zip packaging
  - Those use cases *do* use file extensions
- File extensions are also part of IANA MIME registry
  - Again, MIME registry content is haphazard

#19: Do not sniff PDF

- Adobe Acrobat, Reader are popular interpreters for ISO 32000 format (aka application/pdf)
- Some browsers (Google Chrome, Apple Safari) have independent implementations
- Adobe developers request that no mislabeled content be sent to their software
  - Even if there are some sites with mislabeled content
  - Do Chrome and Safari development groups prefer sniffing?
  - Is sniffing “maximum allowed”?
  - Does content receiving software get to “opt out”?

#20 Opt-in on case-by-case basis

- ***If goal is to reduce amount of sniffing over time***

- As written, two conforming kinds of receivers:
  - NO sniffing at all
  - EXACTLY follow algorithm as specified
  - (except ‘algorithm’ has options for waiting or not waiting)
- Consider user with two browsers
  - One sniffs, others doesn’t
  - Based on return value, chooses one or the other to display
  - SHOULD be conforming, but isn’t
- Expressing this is hard

#21 “Polyglot” use cases

- Content which is legitimately interpretable as more than one MIME type
  - Text/html vs. application/xhtml+xml
  - Application/anything+xml vs. application/xml
  - Image/tiff vs. image/dng
  - Zip vs. zip-based MIME type

- Which to pick? How to resolve? False negatives?

#16: Lack of explanatory text and justification

- Some justification in Barth et all cited research paper
- Test suite for validation against algorithm
  - Can we find at least one real , deployed, useful site (not made up for testing) which needs sniffing
- May need browser help to validate algorithm
  - Otherwise Hard to extract which MIME type is actually used
- Test suite should also cover email, ftp, file content
- Need help
  - Hosting, maintenance, running tests

#22 Charset sniffing

- Part of overall “sniffing” process
  - Receiver needs to know not just MIME type, but entire content-type
  - Sniffing here is just first part of whole algorithm
  - If not part of this document
    charset sniffing still needs to be standards track
  - Sniffing charset currently HTML5 document

- If scope includes unlabeled content
  - *SOME* text types and application/something+xml
    types may need to know charset before proceeding
  - At least determining whether UTF16 or ASCII-compatible for XML declaration

More issues

- All sniffing is potentially “privilege escalation”
  - E.g., text/plain;charset=utf8 with buggy utf8 interpreter
  - Is privilege escalation the right concept?
- Sniffing for different purposes needs different algorithms?
  - presentation to end user
  - scanning for viruses, copyrighted material, “unwanted” content
- Standards track, BCP, Informational?