Independent Submission

Internet-Draft

Intended status: Informational
Expires: August 11, 2020

M. Fabbrini

February 8, 2020

# A Web Model where Content Is Stored in a File's Source: The Unentangled Network

draft-fabbrini-web-model-unentangled-network-00

Abstract

This document describes an experimental model of web whose main characteristic is that the content is stored in a file's source and accessed through a common text browser from the command line under Linux Os.

This work also aims to evaluate the implications of such a network in relation, among other aspects, to security and tracking.

Status of This Memo

This Internet-Draft will expire on August 11, 2020.

## Copyright Notice

Table of Contents



## 1. Introduction

Nowadays getting text content is not a risk-free task and also involves some drawbacks that often discourage the intensive use of the internet by some users who are particularly sensitive to privacy issues.

Rendering a web page in a browser poses first of all security risks

deriving mostly from the execution of malicious script code.

Secondly, the tracking of every user action carried out by the advertising machine, which was once mainly represented by cookies, is now enriched by tools that are permanently installed on local

storages, such as for example service workers and IndexedDB.

The particular web model proposed in this document mitigates both the security risks and the intrusiveness of tracking tools.

## 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. The Core

The peculiar characteristic of this model is that the text content

is inserted in the source of a file.

The carrier document can be of two types:

i. a text file saved as an image or binary format.

ii. a file in which the textual content follows the correct hex

signature such as for instance "42 4D" relating to the BMP format.

## 3. System Requirements

The model was imagined for the Linux Os environment with Lynx as text browser. Lynx, one of the most popular web browsers for command-line interfaces, was originally designed to display plain ASCII text on simple terminals of UNIX, without including any multimedia content. Although Lynx is preferable for some specific features, any other text web browser can be used.

## 4. Content Retrieving

To start Lynx, at the command line prompt, enter 'lynx' followed by the '-source' option and append a carrier document's url.

Example:

'lynx -source https://example.com/.../foo.png'

The result of executing this command is that the unrendered source of the document is displayed.

## 5. Ineffectiveness of Malicious Scripts

Since the unrendered source is retrieved, no event can be triggered

by a script.

In particular, attacks launched via JavaScript will be impossible to perform.

## 6. Fingerprinting

Browser fingerprinting involves gathering information about an internet user's browser and associated software and hardware, such as the browser type, the operating system, various network request headers, cookies, extensions, screen resolution and so on.

These properties can be collected using JavaScript.

Since in this environment no script can be run, fingerprinting methodologies based on JavaScript are ineffective.

## 7. Ads

In a text-only environment, with scripting languages out of the game, the invasive banner and video ads that often make the content de facto impossible to read, will not find space.

## 8. Separation of Content and Form

This web model allows the separation of content and form.

In fact, it is up to the user to choose the font, size, color, acting on the terminal settings of the installed Linux version.

## 9. Enhanced Trust

As a consequence of what is discussed in paragraphs 5, 6 and 7, the information system resulting from the implementation of such a network model will be probably trusted by the users.

## 10. IANA Considerations

This memo includes no request to IANA.

## 11. Security Considerations

In addition to what is examined in paragraph 5, it is worth noting that although in any browser it is possible to disable JavaScript, in the Unentangled Network security is in a certain sense "by design". In fact, it is the model itself that prevent scripts from running and no user intervention is required.

## 12. Informative References

[I-D.wood-pearg-website-fingerprinting]
I. Goldberg, T. Wang, C. Wood, "Network-Based Website Fingerprinting", draft-wood-pearg-website-fingerprinting-00, (work in progress).

[SW] F. Copes "Service Workers explained" <https://flaviocopes.com/service-workers/>

[Lynx] Thomas E. Dickey <http://lynx.invisible-island.net/>

[JSTAttacks] Michael Schwarz, Florian Lackner, Daniel Gruss Graz University of Technology "JavaScript Template Attacks: Automatically Inferring Host Information for Targeted Exploits

<https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_01B-4_Schwarz_paper.pdf>

[JSFingerprinting] T.Claburn "JavaScript tells all, which turns out not to be so great for privacy: Side-channel leaks can be exploited to follow you around the interweb <https://www.theregister.co.uk/2019/06/11/javascript_fingerprinting/>

Author's Address

Michele Fabbrini

Email: unentangled.net@protonmail.com
335 Via Statale Abetone

I-56017 San Giuliano Terme

Pisa, Tuscany

Country: Italy