STIX Cyber Observable - Webpage Object

"Record an extract of text from a webpage"

What is it?

When I've spoken with different dark web monitoring threat intelligence providers about STIX, and specifically about the release of STIX v2, inevitably the first or second question I get is 'Can we record our dark web web forum monitoring in STIX v2?". All I've spoken to wanted to know if they could share excerpts from their web forum posts using STIX v2 objects.

The answer today is **no**.

We have a way of recording a full binary copy of an HTTP request using the Network object with HTTP extension, but there is no way of easily recording an HTTP response as yet, or the webpage that is returned. This makes it difficult to describe:

- A Webpage containing a javascript link to a specific piece of javascript
- A series of text extracts from within a webpage
- A Webpage with a title of 'My ransomware page'
- Changes to a webpage over time

I believe we need a way of recording information about a webpage in a structured way, enabling intel providers to easily record multiple partial or full extracts from a webpage. There is a real need for it; MISP has a 'text' type to allow extracts of text to be recorded for situations just like this.

Why do we need it?

We need to give intel providers a way of recording information they have found on webpages in a structure that allows them to derive higher level intelligence, and enhances their ability to share either the complete page or extracts of that page information to their customers.

With an ability to share text extracts of webpages, we allow threat intelligence providers to:

- Send dark-web criminal forum extracts to their customers showing the threat actors discussing their upcoming plans.
- Send extracts of ransomware .onion landing pages, helping your customers detect ransomware installs within your businesses network.
- Record web-defacements.
- Record website changes over time with a series of webpage objects.

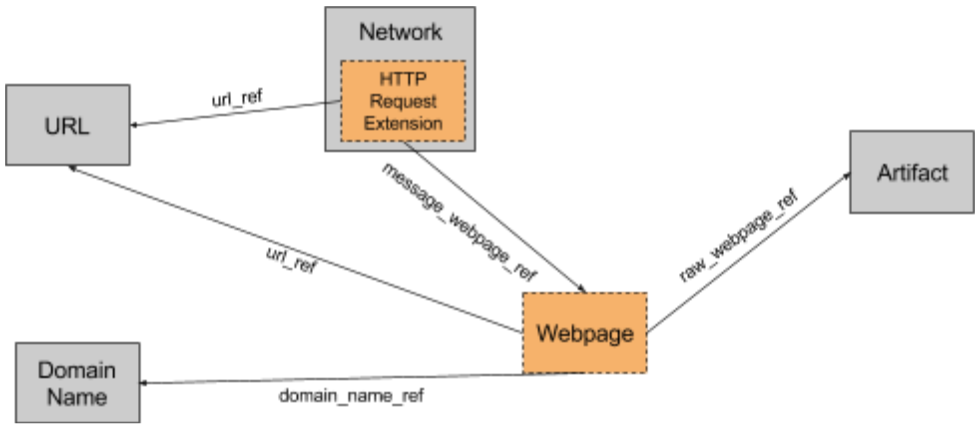
Even more importantly, having a structure for recording webpage content means that we can use that same structure for creating STIX Indicators that can look for webpage content… useful if you are a web proxy vendor and you want to enhance your use of STIX threat intelligence.

How would it work?

I propose that we add a new STIX Cyber Observable (SCO) object to STIX v2.1 - the **Webpage** object, and that we create an HTTP Response extension for the Network Object to allow us to use the Webpage object to describe HTTP responses containing the webpage as well as the Webpage itself.

The Webpage object would allow an (x)HTML webpage to be modelled within STIX, allowing its information to be recorded. The Webpage object could record some or all of the content of the webpage if so desired.

The Webpage (and HTTP Request extension object) could then be related to other existing SCOs like so:



How to model and capture extracted text from a criminal forum website

In order to do this we would need to either create a new `http-response-ext` for the Network object that allows a **message_webpage_ref** entry pointing to the Webpage object, or rename the `http-request-ext` object to `http-ext` and include an additional **message_webpage_ref** entry which would enable the HTTP object to reference the webpage object.

What benefits would it provide?

As mentioned earlier, the Webpage object gives intel providers a new way to share the sort of intel they want to share, and allows recipients to use that information to help detect maliciousness in their web traffic.

Creating a structure to recording webpage content allows web proxy vendors to use that information to detect badness.

By providing the ability to share text extracts from webpages, we allow threat intelligence providers to:

- Send dark-web criminal forum extracts to their customers showing the threat actors discussing their upcoming plans.
- Send extracts of ransomware .onion landing pages, helping your customers detect ransomware installs within your businesses network.
- Record web-defacements.
- Record website changes over time with a series of webpage objects.

STIX Cyber Observable object proposal

2.6.Webpage Object

Type Name: `webpage`

The Webpage Object represents an instance of a webpage, corresponding to the HTML W3C recommendations described at https://www.w3.org/TR/#tr_HTML.

All HTML needs to be escaped so that it can be represented within JSON as per RFC7159. A reminder that all quotation marks, reverse solidus, and the control characters (U+0000 through U+001F) must be escaped by preceding them with a reverse solidus (\) e.g. the HTML string `<link rel="stylesheet" href="css\mysite.css" type="text/css">` would be placed into the links list as `"<link rel=\"stylesheet\" href=\"css\mysite.css\" type=\"text/css\">"`.

Any characters in the encoded value which cannot be decoded into Unicode **SHOULD** be replaced with the 'REPLACEMENT CHARACTER' (U+FFFD). If it is necessary to capture the raw HTML as observed, this can be achieved by referencing an Artifact Object through the **raw_webpage_ref** property.

2.6.1. Properties

Common Properties		
type, description, extensions		
Webpage Object Specific Properties		
date, content_type, url_ref, domain_name_ref, base, links, metas, scripts, noscript, imgs, pictures, as, addresses, objects, articles, embeds, sections, audios, videos, iframe_refs, head, body_text, body, raw_webpage_ref		
Property Name	**Type**	**Description**
type (required)	`string`	The value of this property **MUST** be `webpage`.
date (optional)	`timestamp`	Specifies the date/time that the webpage was retrieved from the website.
content_type (optional)	`string`	Specifies the value of the “Content-Type” header of the webpage.
url_ref (optional)	`object-ref`	Specifies the location of the webpage using a URL object. The object referenced in this property **MUST** be of type `url`.
domain_name_ref (optional)	`object-ref`	Specifies the domain name of the webpage using a domain-name object.

		The object referenced in this property **MUST** be of type `domain-name`.
title(optional)	`string`	Specifies the **title** element as defined in the latest W3C HTML Recommendation extracted from the head of the webpage.
base (optional)	`string`	Specifies the **base** element as defined in the latest W3C HTML Recommendation extracted from the head of the webpage.
links (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **link** elements as defined in the latest W3C HTML Recommendation extracted from either the head or body of the webpage. The **link** elements included in the list MUST include the element start and end tags in the string e.g. "<link>....</link>". List values **MUST** appear in the same order as present in the webpage.
metas (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **meta** elements as defined in the latest W3C HTML Recommendation extracted from either the head or body of the webpage. The **meta** elements included in the list MUST include the element start and end tags in the string e.g. "<meta>....</meta>". List values **MUST** appear in the same order as present in the webpage.
scripts (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **script** elements as defined in the latest W3C HTML Recommendation extracted from either the head or body of the webpage.

		The **script** elements included in the list MUST include the element start and end tags in the string e.g. "<script>....</script>". List values **MUST** appear in the same order as present in the webpage.
noscripts (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **noscript** elements as defined in the latest W3C HTML Recommendation extracted from either the head or body of the webpage. The **noscript** elements included in the list MUST include the element start and end tags in the string e.g. "<noscript>....</noscript>". List values **MUST** appear in the same order as present in the webpage.
imgs (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **img** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **img** elements included in the list MUST include the element start and end tags in the string e.g. "". List values **MUST** appear in the same order as present in the webpage.
pictures (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **picture** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **picture** elements included in the list MUST include the element start and end tags in the string e.g. "<picture>....</picture>".

		List values **MUST** appear in the same order as present in the webpage.
as (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **a** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **a** elements included in the list MUST include the element start and end tags in the string e.g. "....". As a reminder, **a** elements represent the majority of clickable links on a webpage. List values **MUST** appear in the same order as present in the webpage.
addresses (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **address** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **address** elements included in the list MUST include the element start and end tags in the string e.g. "<address>....</address>". List values **MUST** appear in the same order as present in the webpage.
articles (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **article** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **article** elements included in the list MUST include the element start and end tags in the string e.g. "<article>....</article>". List values **MUST** appear in the same order as present in the webpage.

objects (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **object** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **object** elements included in the list MUST include the element start and end tags in the string e.g. "<object>....</object>". List values **MUST** appear in the same order as present in the webpage.
embeds (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **embed** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **embed** elements included in the list MUST include the element start and end tags in the string e.g. "<embed>....</embed>". List values **MUST** appear in the same order as present in the webpage.
sections (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **section** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **section** elements included in the list MUST include the element start and end tags in the string e.g. "<section>....</section>". List values **MUST** appear in the same order as present in the webpage.
videos (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **video** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage.

		The **video** elements included in the list MUST include the element start and end tags in the string e.g. "<video>....</video>". List values **MUST** appear in the same order as present in the webpage.
audios (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains **audio** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. The **audio** elements included in the list MUST include the element start and end tags in the string e.g. "<audio>....</audio>". List values **MUST** appear in the same order as present in the webpage.
iframe_refs (optional)	`list` of type `object-ref`	Specifies a `list` of type `object-ref` where each `object-ref` points to another webpage object representing the **iframe** elements as defined in the latest W3C HTML Recommendation extracted from the body of the webpage. List values **MUST** appear in the same order as present in the webpage.
head (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains HTML excerpts from the **head** element as defined in the latest W3C HTML Recommendation from the head of the webpage. List values **MUST** appear in the same order as present in the webpage. **Note: Items in this list MAY contain HTML markup.**

body_text (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains excerpts of text from the **body** element as defined in the latest W3C HTML Recommendation from the body of the webpage. List values **MUST** appear in the same order as present in the webpage. **Note: Items in this list MUST NOT contain HTML markup. The HTML markup is removed and only the text is shown.**
body (optional)	`list` of type `string`	Specifies a `list` of type `string` where each list item contains excerpts of HTML markup and text from the **body** element as defined in the latest W3C HTML Recommendation from the body of the webpage. List values **MUST** appear in the same order as present in the webpage. **Note: Items in this list MAY contain HTML markup.**
raw_webpage_ref (optional)	`object-ref`	Specifies the complete raw binary contents of the webpage, including both the headers and body, as a reference to an Artifact Object. Note: This does NOT include the HTTP headers. HTTP Headers are specified in a Network Object with an HTTP extension.. The object referenced in this field **MUST** be of type `artifact`.

2.6.3. Examples

Hacked Website redirecting to exploit site using Javascript

```
{
  "0": {
    "type": "url",
```

```
    "value": "https://mymainnews.com/news/index.html"
  },
  "1": {
    "type": "webpage",
    "url_ref" "0",
    "title": "Main News",
    "links": [
      "<link rel=\"stylesheet\" href=\"css\mysite.css\" type=\"text/css\">"
    ],
    "scripts": [
      "<script src=\"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.js\"
type=\"text/javascript\"></script>",
      "<script src=\"https://myhackedsite.com/files/uploads/d.js\" type=\"text/javascript\"></script>"
    ]
  }
}
```

Hacked Website housing images containing steganographic text

```
{
  "0": {
    "type": "url",
    "value": "https://wayneindustries.com/research/index.html"
  },
  "1": {
    "type": "webpage",
    "url_ref" "0",
    "title": "I've hacked the Internetz",
    "links": [
      "<link rel=\"stylesheet\" href=\"css\mysite.css\" type=\"text/css\">"
    ],
    "scripts": [
      "<script src=\"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.js\"
type=\"text/javascript\"></script>",
      "<script src=\"https://myhackedsite.com/files/uploads/d.js\" type=\"text/javascript\"></script>"
    ],
    "body_text": [
      "I hack3d the Internetz and no 1 will get me eva.",
      "I've managed to get into the CIA man!",
    ],
    "body": [
      "I hack3d the Internetz and no 1 will get me eva.",
      "I've managed to get into the CIA man!",
    ]
  }
}
```

Recording Forum Posts on Website

```
{
  "0": {
    "type": "domain-name",
    "value": "wayneindustries.com"
  },
  "1": {
    "type": "network-traffic",
    "dst_ref": "0",
    "protocols": [
      "tcp",
      "http"
    ],
```

```
    "extensions": {
      "http-request-ext": {
        "request_method": "get",
        "request_value": "/research/index.html",
        "request_version": "http/1.1",
        "request_header": {
          "Accept-Encoding": "gzip,deflate",
          "User-Agent": "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113",
          "Host": "wayneindustries.com"
        },
        "message_webpage_ref": "2"
      }
    }
  },
  "2": {
    "type": "webpage",
    "domain_name_ref": "0",
    "url_ref" "3",
    "title": "I've hacked the Internetz",
    "body_text": [
      "I hack3d the Internetz and no 1 will get me eva.",
      "I've managed to get into the CIA man!",
    ],
    "body": [
      "I hack3d the Internetz and no 1 will get me eva.",
      "I've managed to get into the CIA man!",
    ]
  },
  "3": {
    "type": "url",
    "value": "https://wayneindustries.com/research/index.html"
  }
}
```