alessandro de vito *(@_cube0x8)*

truelit

how i met your browser: going incognito doesn’t hide your browsing from ragamuffin

bsides zurich 2017 – zurich (switzerland)

agenda

- Introduction
  - About me
  - What is Chrome Ragamuffin
  - What Chrome Ragamuffin is not
  - Why Chrome Ragamuffin should be useful
- Forensic Overview
  - Chrom(e|ium)
  - Objects we have focused on
  - What do we get from those
  - Chrome Ragamuffin architecture

what is chrome ragamuffin?

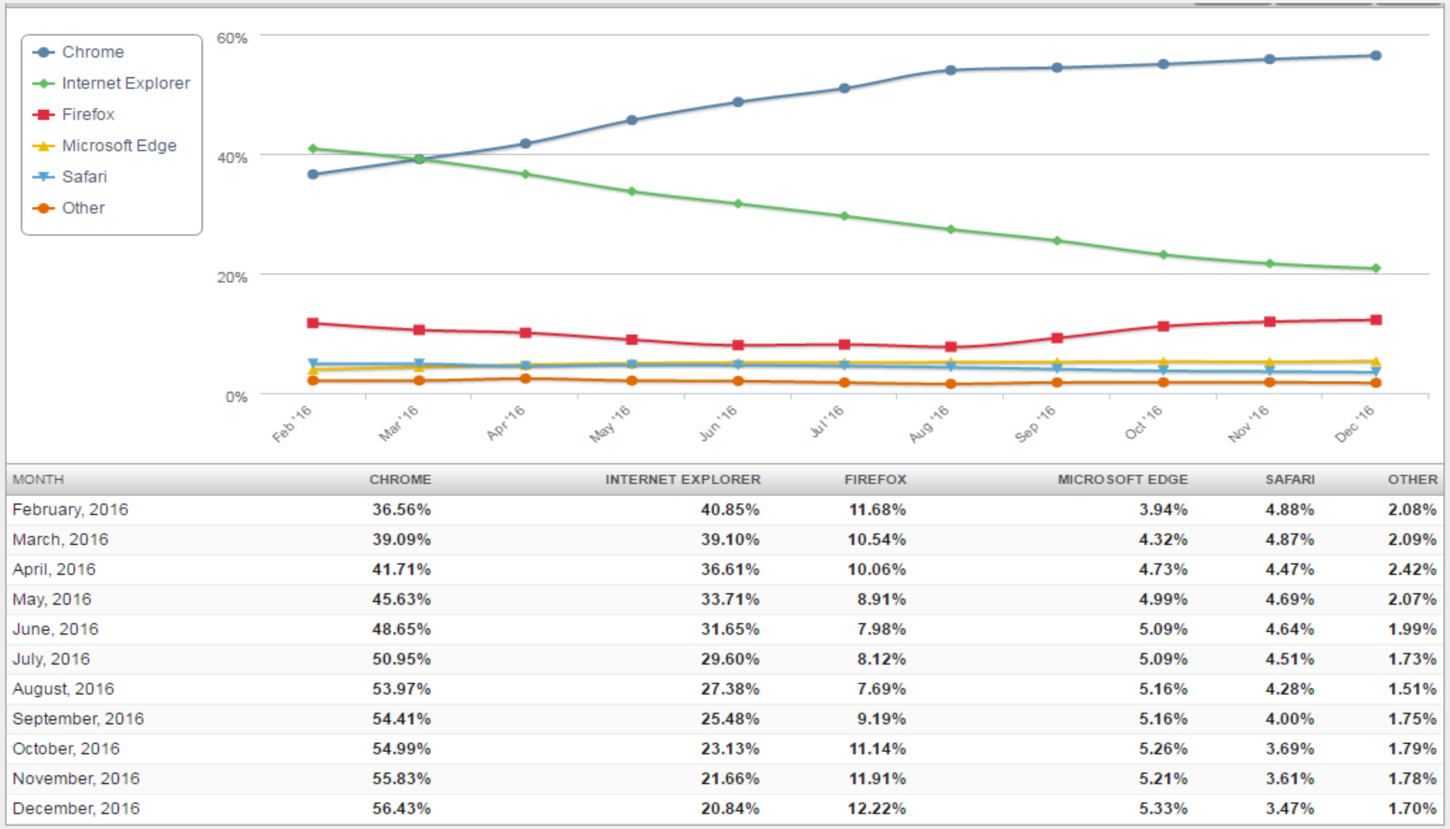
- A **research project** that aims to gather **useful artifacts** from the **whole** web browser **address space**
- We analysed the **source code** and main **data structures** to figure out which **artifacts** may be **interesting** to our **purposes**
- Now, we have been implementing the **PoC** using **Volatility Framework**

what chrome ragamuffin is not

- IDS/NIDS
- It’s **not** a **browser extension**
- It’s **not** an automatic agent able to detect **live threats**
- It’s **not** a plugin **designed** to analyse SQLite **databases**

truelit

why chrome ragamuffin should be useful



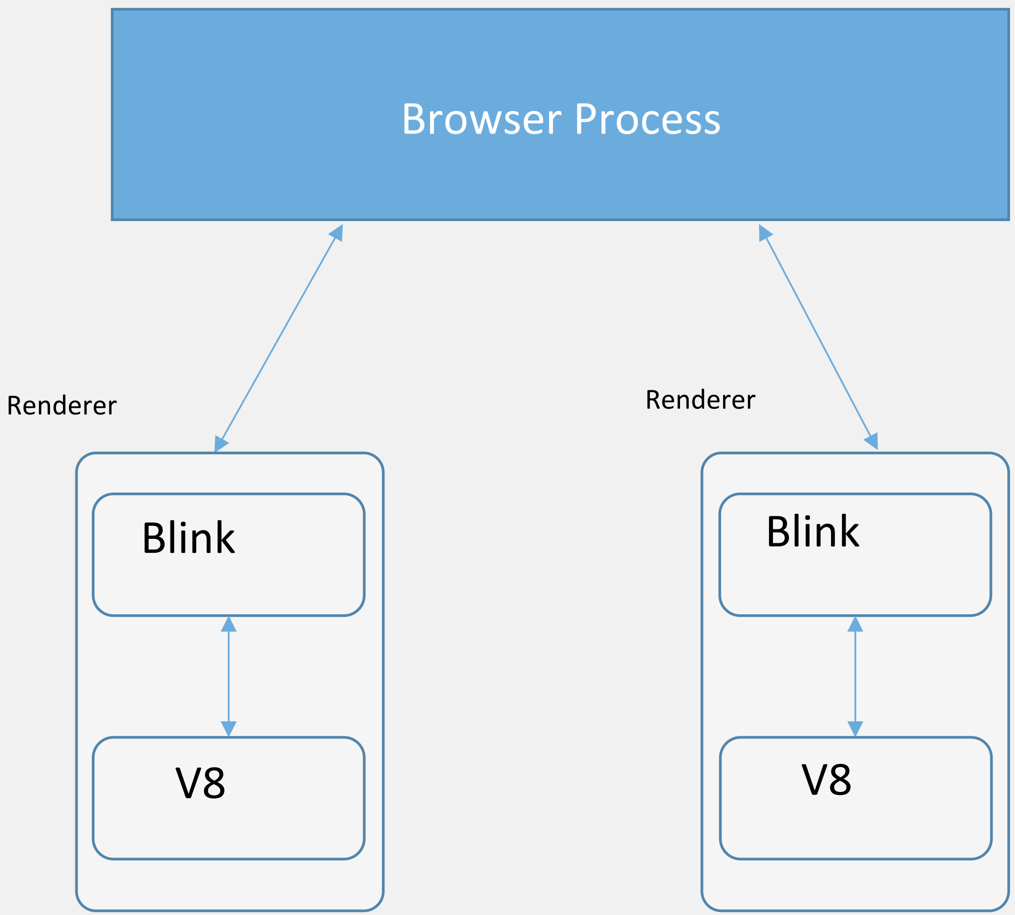
MONTH	CHROME	INTERNET EXPLORER	FIREFOX	MICROSOFT EDGE	SAFARI	OTHER
February, 2016	36.56%	40.85%	11.68%	3.94%	4.88%	2.08%
March, 2016	39.09%	39.10%	10.54%	4.32%	4.87%	2.09%
April, 2016	41.71%	36.61%	10.06%	4.73%	4.47%	2.42%
May, 2016	45.63%	33.71%	8.91%	4.99%	4.69%	2.07%
June, 2016	48.65%	31.65%	7.98%	5.09%	4.64%	1.99%
July, 2016	50.95%	29.60%	8.12%	5.09%	4.51%	1.73%
August, 2016	53.97%	27.38%	7.69%	5.16%	4.28%	1.51%
September, 2016	54.41%	25.48%	9.19%	5.16%	4.00%	1.75%
October, 2016	54.99%	23.13%	11.14%	5.26%	3.69%	1.79%
November, 2016	55.83%	21.66%	11.91%	5.21%	3.61%	1.78%
December, 2016	56.43%	20.84%	12.22%	5.33%	3.47%	1.70%

- **Google Chrome** is the **most used web browser** in the world
- Nowadays, there are a lot of tools to analyse **disk-based** artifacts**files** mapped in memory (**SQLite databases**)
- Now, with Ragamuffin, we can achieve an important goal:
  - get **valuable artifacts** from the **whole** address space.
  - put together objects to get a detailed **overview** about the **history navigation, web browser contents** and clues about **malicious activities** happened on it

truelit

forensic overview

google chrome overview



- The **Browser Process** represents a top-level browser window which **drives** the **Renderers** during the navigation by IPC system.
- Each **tab** is represented by a **Renderer Process** and communicates with the Browser Process to **access the general I/O activities** (Network/Disk Cache/ Storage)
- Each **tab contains** an instance of the **Blink** Engine (for interpreting and layout HTML) and of the **V8** JavaScript engine (to run JavaScript Code)

truelit

objects we have focused on

- Browser process:
  - WebContents: It contains all the information about a **tab**. Each **WebContents has** exactly one **NavigationController**; each **NavigationController belongs** to one **WebContents**.
  - NavigationController: a NavigationController maintains the **back-forward vector** for a WebContents and **manages all navigation** within that vector.
  - NavigationEntry: **NavigationController contains NavigationEntry** objects. They contain all the **information** required to **recreate** a **browsing state** like some **clear text title, URL, serialized information** related to form fields.

objects we have focused on

- Renderer process (Blink engine):
  - Document: a data structure which **describes an HTML/XML web page**. It **contains the metadata** of the web page (i.e. DOCTYPE, title, language) and the pointer to the **DOM**.
  - DOM: **Document Object Model** pointed from the Document and represents the page content in a **tree structure**.
  - MemoryCache: contains a map of **cached resources** required by a web page

objects we have focused on

what do we get from those?

Browser objects:

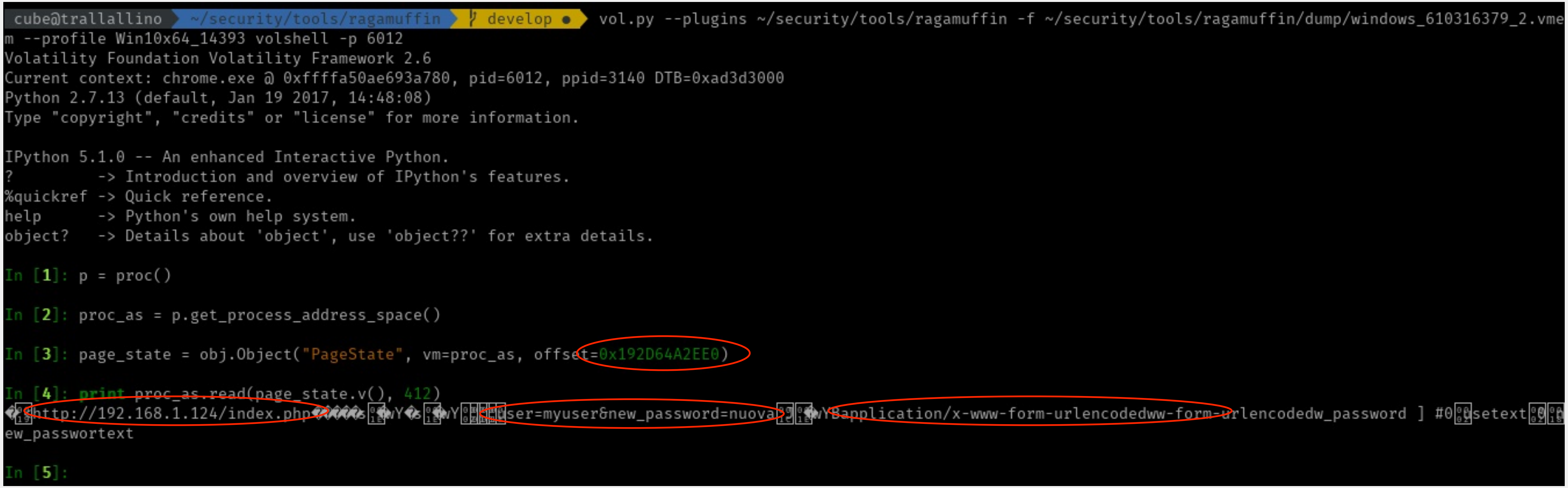
- Evidence: offset object, url, status code, method, transition, timestamp, restore type, page type, **form params**

Entry ID	Controller ID	Offset	Title	User typed url	Original request url	Status code	Method	Post params	Transition	Referer	Redirect Chain	UTC Timestamp	Restore type	Type page/post id
1	2	**0x192d2a034a0**	Nuova scheda	chrome://newtab/ies	"https://www.google.it/_/chrome/newtab?espv=2&ie=UTF-8"	200	GET	None	Uknown	None	None	**18/09/17 14.49**	Entry was not restored	None
1	2	0x192cefdd5e0	None	…	…	None	GET	None	Inner frame	None	None	18/09/17 14.49	None	**"<!--framePath //<!--frame0-->-->"**
2	2	0x192d2a03c60	None	http://192.168.1.124/notexists.html	**http://192.168.1.124/notexists.html**	0	GET	None	**Typed URL in the address bar**	None	None	18/09/17 14.50	**Entry was not restored**	ERROR
3	2	0x192d2a02900	Test	http://192.168.1.124/test.html	http://192.168.1.124/test.html	200	GET	None	Typed URL in the address bar	None	None	18/09/17 14.50	Entry was not restored	**NORMAL**
3	2	0x192D2508540	None	http://192.168.1.124/index.php	http://192.168.1.124/index.php	None	**POST**	**0x192D64A2EE0**	**POST REQUEST**	None	None	19/09/17 14.50	None	1.50575E+15

objects we have focused on
what do we get from those?

POST Params:

- With the memory address, we can dump the **PageState** object which contains **serialized information** about the **submitted form**



```
cube@trallallino  ~/security/tools/ragamuffin  develop ●  vol.py --plugins ~/security/tools/ragamuffin -f ~/security/tools/ragamuffin/dump/windows_610316379_2.vme
m --profile Win10x64_14393 volshell -p 6012
Volatility Foundation Volatility Framework 2.6
Current context: chrome.exe @ 0xffffa50ae693a780, pid=6012, ppid=3140 DTB=0xad3d3000
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: p = proc()

In [2]: proc_as = p.get_process_address_space()

In [3]: page_state = obj.Object("PageState", vm=proc_as, offset=0x192D64A2EE0)

In [4]: print proc_as.read(page_state.v(), 412)
��http://192.168.1.124/index.php�����s��wY�s��wY����user=myuser&new_password=nuova����wY8application/x-www-form-urlencodedww-form-urlencodedw_password ] #0��setext���
ew_passwortext

In [5]:
```

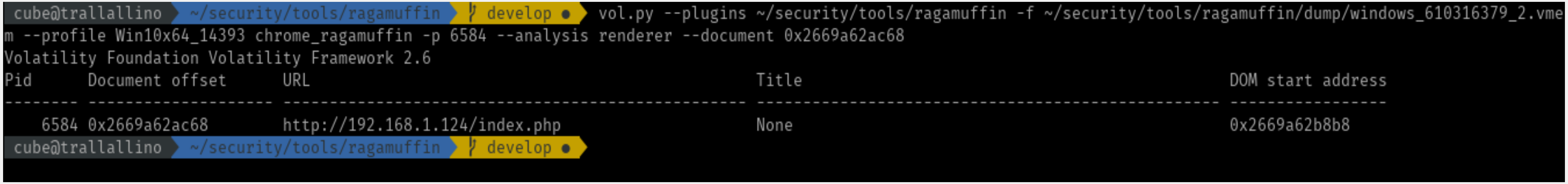
truelit

objects we have focused on

what do we get from those?

Renderer objects:

- Evidence: PID of the **tab** which **contains** the **specific document**, document **offset**, **URL** of the document, **title**, **<html>** node address of the DOM tree



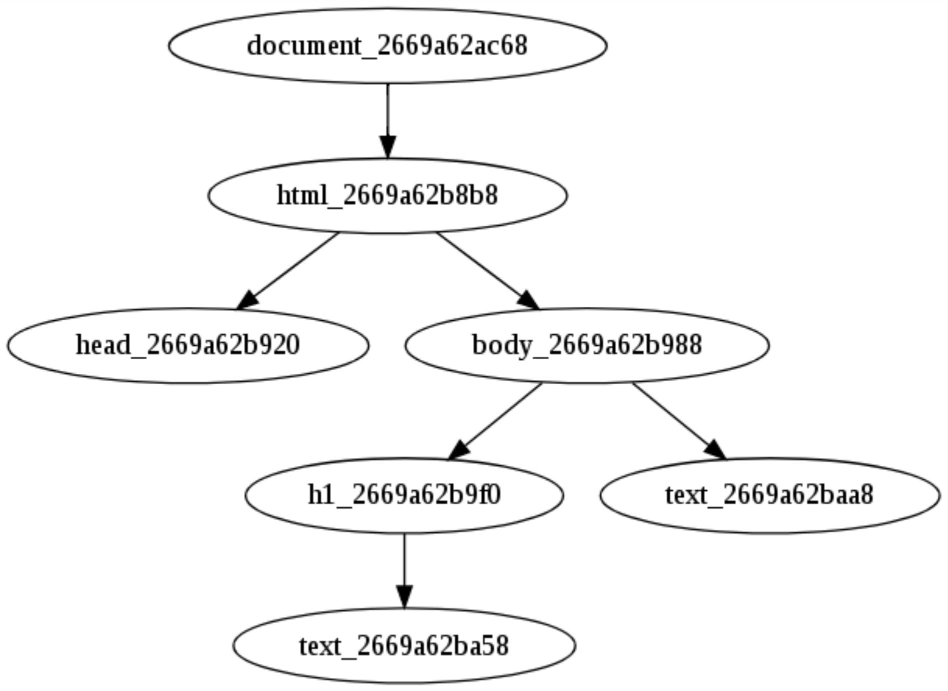
```
cube@trallallino  ~/security/tools/ragamuffin  develop ●  vol.py --plugins ~/security/tools/ragamuffin -f ~/security/tools/ragamuffin/dump/windows_610316379_2.vme
m --profile Win10x64_14393 chrome_ragamuffin -p 6584 --analysis renderer --document 0x2669a62ac68
Volatility Foundation Volatility Framework 2.6
Pid      Document offset      URL                                                Title                                              DOM start address
-------- -------------------- -------------------------------------------------- -------------------------------------------------- -----------------
    6584 0x2669a62ac68        http://192.168.1.124/index.php                     None                                               0x2669a62b8b8
cube@trallallino  ~/security/tools/ragamuffin  develop ●
```

objects we have focused on

what do we get from those?

Renderer objects:

Evidence: By the “**DOM start address**” field we can **get** the entire **Document Object Model tree** in its **dot** (**high-level** structure of the page) and **text** notation (**detailed** contents)



- Node tag: html
Node attributes: {lang=it}
Memory offset: 0x2669a62b8b8
- Node **tag**: head
Node attributes: {}
Memory offset: 0x2669a62b920
- Node tag: body
Node **attributes**: {class=test, id=123}
Memory offset: 0x2669a62b988
- Node tag: h1
Node attributes: {id=title}
Memory offset: 0x2669a62b9f0
- Node tag: **Text**
Content: **You've successfully changed your password**

chrome ragamuffin architecture

Implemented in two parts:

1. *libchrome_$release.py* library
   - We're reading the Chromium's source code and **extracting** the **objects** we interested in to **convert** them in **VTypes** (from **C** data structures to **Python** objects)
   - It handles the **extraction** of the **WTF::StringImpl** objects and other **platform-specific data types**
2. *chrome_ragamuffin.py* plugin
   - This is the **main plugin**.
   - It imports the *libchrome* library and use it to **scan** for the **signatures**, to make **validation** in order to exclude false positives and to render the **output**

chrome ragamuffin architecture

Plugin

Get a detailed overview of the Web Browser status:

- Detailed information about **navigation history**
- Memory **addresses** of the main objects involved
- Objects from the **renderer process** (third-party JS, iframe, DOM tree)

By volshell

Perform a lot of fun manual analysis!

- **Get deeper** in the address space (and **dump** a singular **object**)
- Unveil **relationship between objects**
- Analyze traces about **client-side attacks**

state of art

Other tools

- WebCapsule/ChromePic
  - **Instrumentation** of the web browser source code
  - Records and Replay **key logger**

- Chrome History (@superponible)
  - **SQLite databases** in memory (visited pages, cookies, search terms, downloaded file, visit details)
  - SQLite databases are **saved on disk**

state of art

Chrome Ragamuffin

- Pro
  - Agnostic approach
  - Whole address space (a lot of new artifacts)
  - Overcoming incognito mode

state of art

Chrome Ragamuffin

- Limitations
  - Garbage Collector (Olipan, Scavenger ecc.) **collects** unused **objects**

work in progress

- ⌛HTTP cached Reponse Body (Work in progress)
- ⌛MemoryCache (in-memory renderer cache) (Work in progress)
- ⌛V8 (for now, Isolate, Heap, Spaces, Page Memories) (Almost-Work-in-progress)

Linux/macOSx support (TODO)

truelit

thanks

- Join the project on github! Search for *cube0x8 (“cube” “zero” “x” “eight”) (https://github.com/cube0x8/chrome_ragamuffin)*
- Email: alessandro.devito@truel.it
- All of you
- BSides Zurich
- TRUEL IT
- All guys who helped me

truelit