Google Safe Browsing: Privacy and Security

Amrit Kumar

Univ. de Grenoble Alpes & Privatics team, INRIA

June 4, 2015

Outline

Outline

Demo time !

d99q.cn

Google Safe Browsing

- **Started** in 2008 by Google and used by :
  - Google Chrome
  - Mozilla Firefox
  - Apple Safari
  - Opera
- **Impact :** billions of users according to Google
- **Goals :** prevent users from visiting
  - *phishing sites*
  - *malwares sites*
- **Methodology :** blacklist
- API compatibility with `C#`, `Python` and `PHP`
- Cloned by Yandex.

Safe Browsing Lookup API

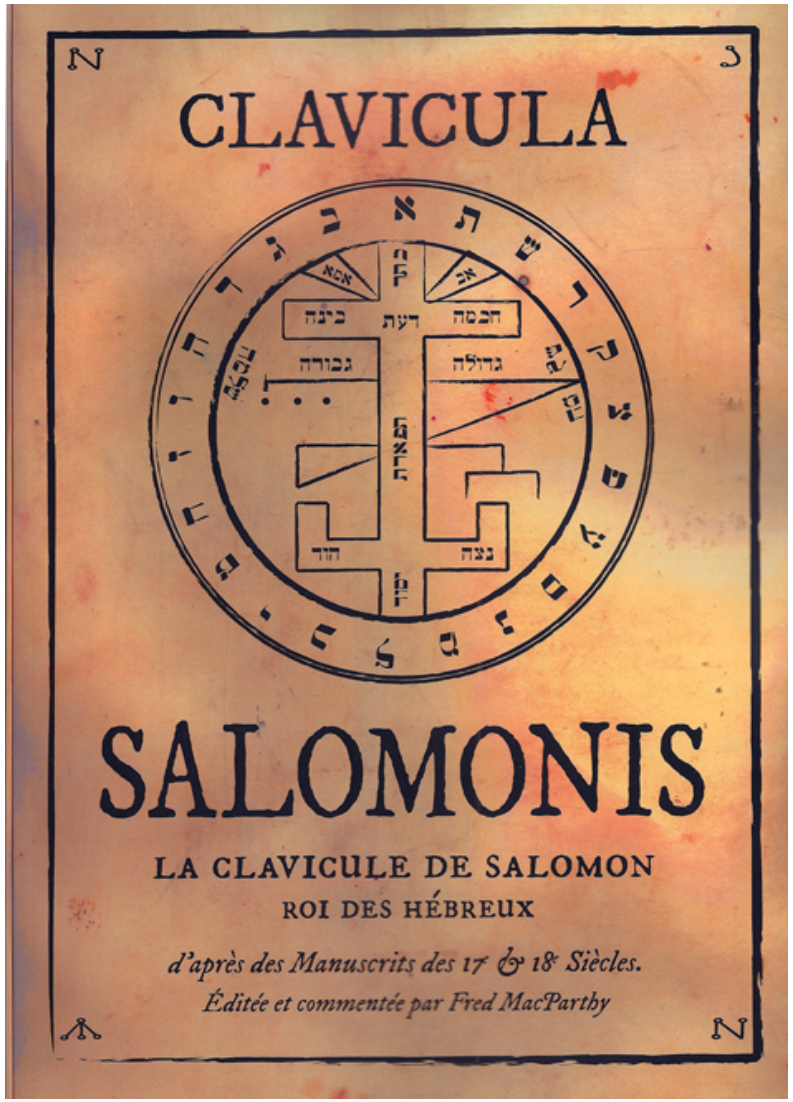
- GOOGLE **crawls** the web to seek phishing and malwares URLs **to feed a blacklist** on their servers.

- **How to use ?**
  Ask GOOGLE's server using a simple `HTTP GET` request.

  `https://sb-ssl.google.com/safebrowsing/api/lookup?`

- **Issues :**
  - bad scaling
  - privacy issue

The first blacklist



72 demons in the catalog

Agares
Aim
Alloces
Amdusias
Amon
Amy
Andras
Andrealphus
Andromalius
⋮

Identifying demon

- **Problem :** Is a hand book. **How to make it a pocket book ?**
- **Solution : Lossy compression.**

Ag
Ai
Al
Am
An
⋮

- From 72 names to 50 prefixes (30% compression).
- From 518 characters to 100 (80% compression).

False positives

- `Hollande` → `Ho` is not in the pocket book. `Hollande` isn't a demon.
- `Valls` → `Va` is in the pocket book.
  But `Valls` isn't in the complete catalog.
  ⇒ false positive !
- If a prefix is in the compressed list :
  - Inconclusive : requires a verification from the handbook
  - For `Va`, we would have : `Valefar`, `Vapula` et `Vassago`.
  - Check among the full words.
- **Solution is interesting if false positives are small in number.**

Google Safe Browsing (GSB) API v3

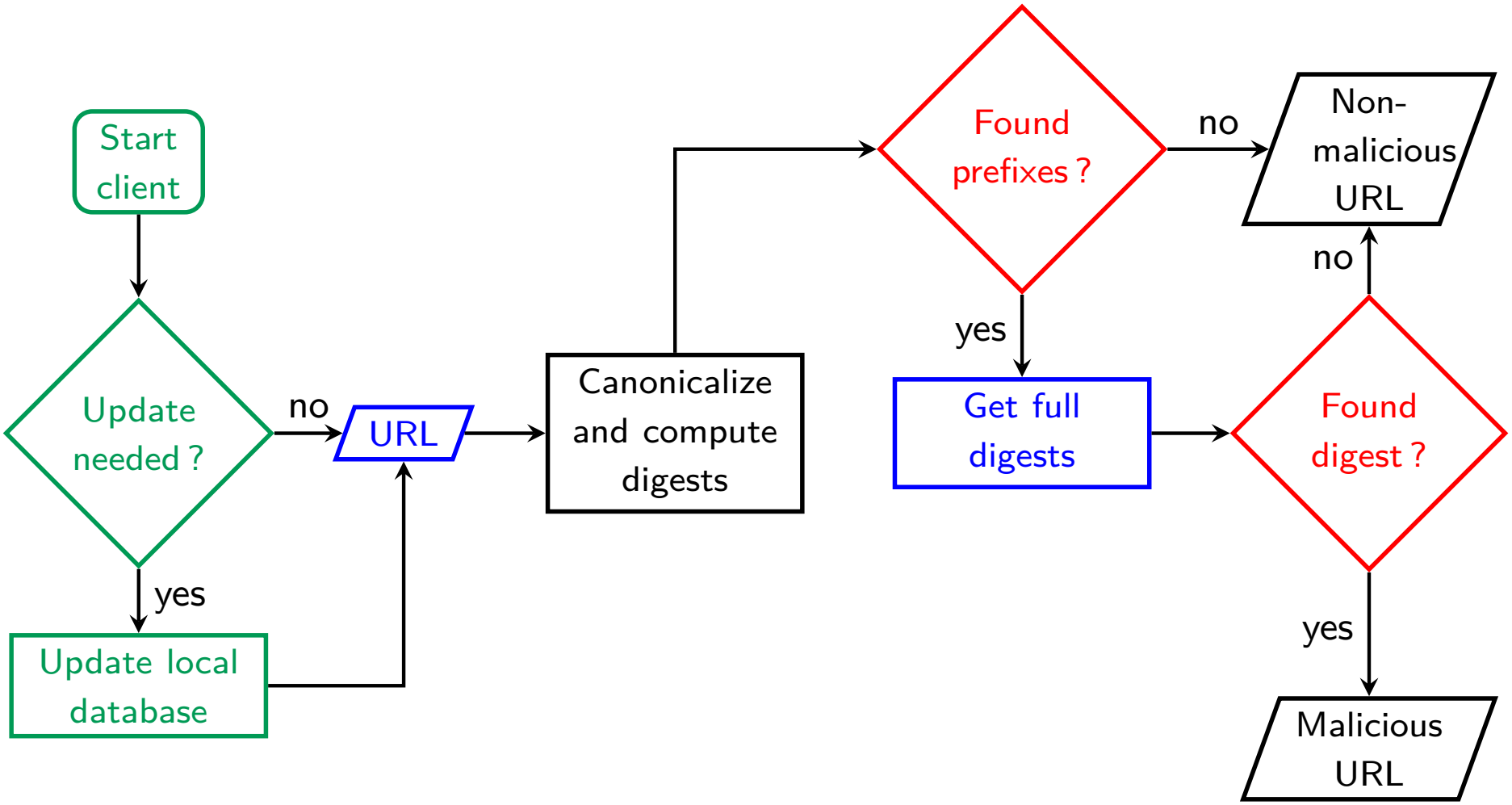
- The local lookups are done over these files :

List name	**Description**	**#prefixes**
goog-malware-shavar	malware	317,807
googpub-phish-shavar	phishing	312,621
goog-regtest-shavar	test file	29,667
goog-whitedomain-shavar	unused	1

- **Nearly $\approx$ 650000 entries overall.**
- We are not working on URLs themselves but on their digests. We only use the first 4 bytes of SHA-256 digest.

```
Prefix32(SHA256(www.example.com/))=0xd59cc9d3
```

GSB API v3


Start client
Update needed ?
no
yes
Update local database
URL
Canonicalize and compute digests
Found prefixes ?
no
Non-malicious URL
yes
Get full digests
Found digest ?
no
yes
Malicious URL

Yandex Safe Browsing (YSB)

Google's Evil Twin

Yandex Safe Browsing API

List name	Description	#prefixes
goog-malware-shavar	malware	283,211
goog-mobile-only-malware-shavar	mobile malware	2,107
goog-phish-shavar	phishing	31,593
ydx-adult-shavar	adult website	434
ydx-adult-testing-shavar	test file	535
ydx-imgs-shavar	malicious image	0
ydx-malware-shavar	malware	283,211
ydx-mitb-masks-shavar	man-in-the-browser	87
ydx-mobile-only-malware-shavar	malware	2,107
ydx-phish-shavar	phishing	31,593
ydx-porno-hosts-top-shavar	pornography	99,990
ydx-sms-fraud-shavar	sms fraud	10,609
ydx-test-shavar	test file	0
ydx-yellow-shavar	shocking content	209
ydx-yellow-testing-shavar	test file	370
ydx-badcrxids-digestvar	.crx file ids	*
ydx-badbin-digestvar	malicious binary	*
ydx-mitb-uids	man-in-the-browser	*
ydx-badcrxids-testing-digestvar	test file	*

Why 32-bit prefixes ?

Optmization

		Data structure (MB)	
SHA-256 prefix (bits)	**Raw data (MB)**	**size**	**Compr.**
32	2. 5	1.3	1.9
64	5.1	3.9	1.3
80	6.4	5.1	1.2
128	10.2	8.9	1.1
256	20.3	19.1	1

Why 32-bit prefixes ?

Privacy

Year	# unique URLs (GOOGLE)	# of domains
2008	1 Billion	177 Million
2012	30 Billion	252 Million
2013	60 Billion	271 Million

	M for URLs			M for domain		
ℓ (bits)	2008	2012	2013	2008	2012	2013
16	2^{28}	2^{28}	2^{29}	253	363	388
32	443	7541	14757	2	3	3
64	2	2	2	1	1	1
96	1	1	1	1	1	1

Outline

Highlights

- Google Chrome Privacy Notice on Safe Browsing.
  "*Google cannot determine the real URL from this information.*"
  (to be read prefixes)

- This statement is re-iterated in GSB usage in Mozilla Firefox.

- **Conclusion :** GSB must provide the same level of privacy than a *private information retrieval algorithm*.

- **Really ?**

Re-identification

URL	32-bit prefix
https://persyval-lab.org/content/edition/	0x2929f0b1
https://persyval-lab.org/content/	0xc99584e3
https://persyval-lab.org/	0x192af851

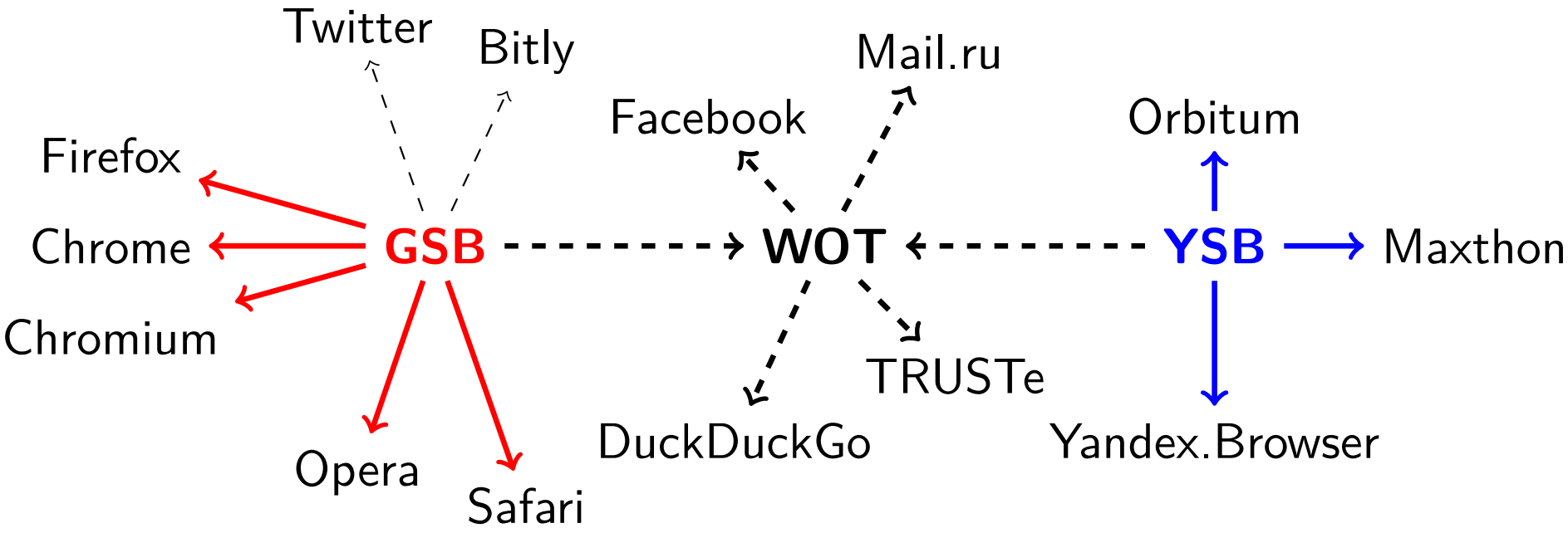
- **Problem with the false-positives :**
- 1 match : 0x2929f0b1 → no privacy issue.
- 2 matches : 0xc99584e3 and 0x192af851 → **Problem**.
- Sending several prefixes is indeed the case.
- **More problem with temporal correlation :**

URL	32-bit prefix
https://persyval-lab.org/phd/appel-2015/depot/	0x6e2abf0a
https://persyval-lab.org/phd/appel-2015/	0x79f13238

Interesting URLs

URL	matching decomposition	prefix
http ://fr.xhamster.com/user/video	fr.xhamster.com/	0xe4fdd86c
	xhamster.com/	0x3074e021
http ://nl.xhamster.com/user/video	nl.xhamster.com/	0xa95055ff
	xhamster.com/	0x3074e021
http ://m.mofos.com/user/login	m.mofos.com/	0x6e961650
	mofos.com/	0x00354501
http ://m.mofos.com/user/logout	m.mofos.com/	0x6e961650
	mofos.com/	0x00354501
http ://mobile.teenslovehugecocks.com/user/join	mobile.teenslovehugecocks.com/	0x585667a5
	teenslovehugecocks.com/	0x92824b5c
http ://fr.xhamster.com/user/kmille	fr.xhamster.com/	0xe4fdd86c
	xhamster.com/	0x3074e021
http ://de.xhamster.com/user/video	de.xhamster.com/	0x0215bac9
	xhamster.com/	0x3074e021
http ://nl.xhamster.com/user/ppbbg	nl.xhamster.com/	0xa95055ff
	xhamster.com/	0x3074e021
http ://nl.xhamster.com/user/photo	nl.xhamster.com/	0xa95055ff
	xhamster.com/	0x3074e021

Am I paranoid ?



- 65% of the browsers in use.
- Major social networks.
- Activated by default in some releases of Tor Browsers.

Orphans

		#full hash per prefix			
	list name	0	1	2	Total
Google	goog-malware-shavar	0.9%	99%	0.1%	317,807
	googpub-phish-shavar	0.9%	99%	0.1%	312,621
Yandex	ydx-malware-shavar	1.5%	98%	0.5%	283,211
	ydx-adult-shavar	43%	57%	0	434
	ydx-mobile-only-malware-shavar	6%	94%	0	2,107
	ydx-phish-shavar	99%	1%	0	31,593
	ydx-mitb-masks-shavar	100%	0	0	87
	ydx-porno-hosts-top-shavar	1%	99%	0	99,990
	ydx-sms-fraud-shavar	95%	5%	0	10,609
	ydx-yellow-shavar	100%	0	0	209

Popular orphans

		#Coll. with TopAlexa			
	list name	0	1	2	Total
Google	goog-malware-shavar	0	572	0	572
	googpub-phish-shavar	0	88	0	88
Yandex	ydx-malware-shavar	73	2,614	0	2,687
	ydx-adult-shavar	38	43	0	81
	ydx-mobile-only-malware-shavar	2	22	0	24
	ydx-phish-shavar	22	0	0	22
	ydx-mitb-masks-shavar	2	0	0	2
	ydx-porno-hosts-top-shavar	43	17,541	0	17,584
	ydx-sms-fraud-shavar	76	3	0	79
	ydx-yellow-shavar	15	0	0	15

Conclusion

- Google and Yandex can track users.
- Mysterious files : Presence of large number of orphans.
- Accountability ?
- Private Information Retrieval is the definitive answer, but ...

Outline

Unsafe Browsing

- SB architecture is meaningful if false positive probability is low.
- Can an attacker increase the false positive probability ?
  - Increase requests towards server.
  - Increase responses towards client.
  - Or both.
- Attack impact :
  - Challenges the design rationale of the verification algorithm.
  - Safe browsing can be potentially brought to its knees.
  - Consumes bandwidth on client's side.

Goal is to mount a DoS attack.

Attack routine

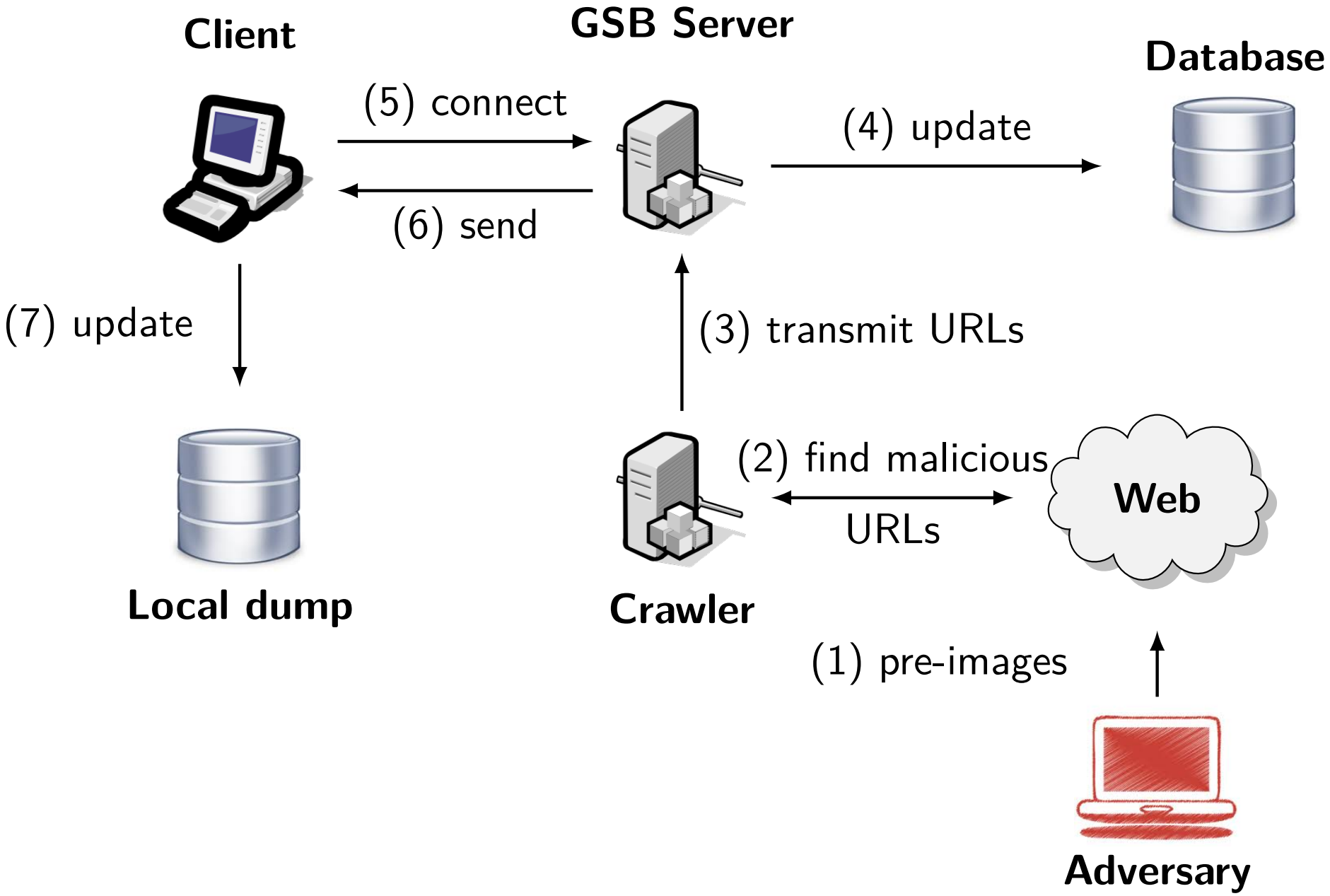
Step 1 : Generate false positives. Example : Hollande is frequently searched. We search for names with the same prefix :

Hochart
Houssin
Hoareau
Hocquet
Horn
⋮

Step 2 : Transform these names into demons and include them into the Key of Solomon.

Step 3 : Observe the impact.

Establishing the flow



Step 1 : Second pre-images

- Given a URL m, find $m' \neq m$ tel que :

$$\texttt{Prefix32(SHA256(}m\texttt{))} = \texttt{Prefix32(SHA256(}m'\texttt{))}$$

- 2^{32} brute-force computations to find such an m'.

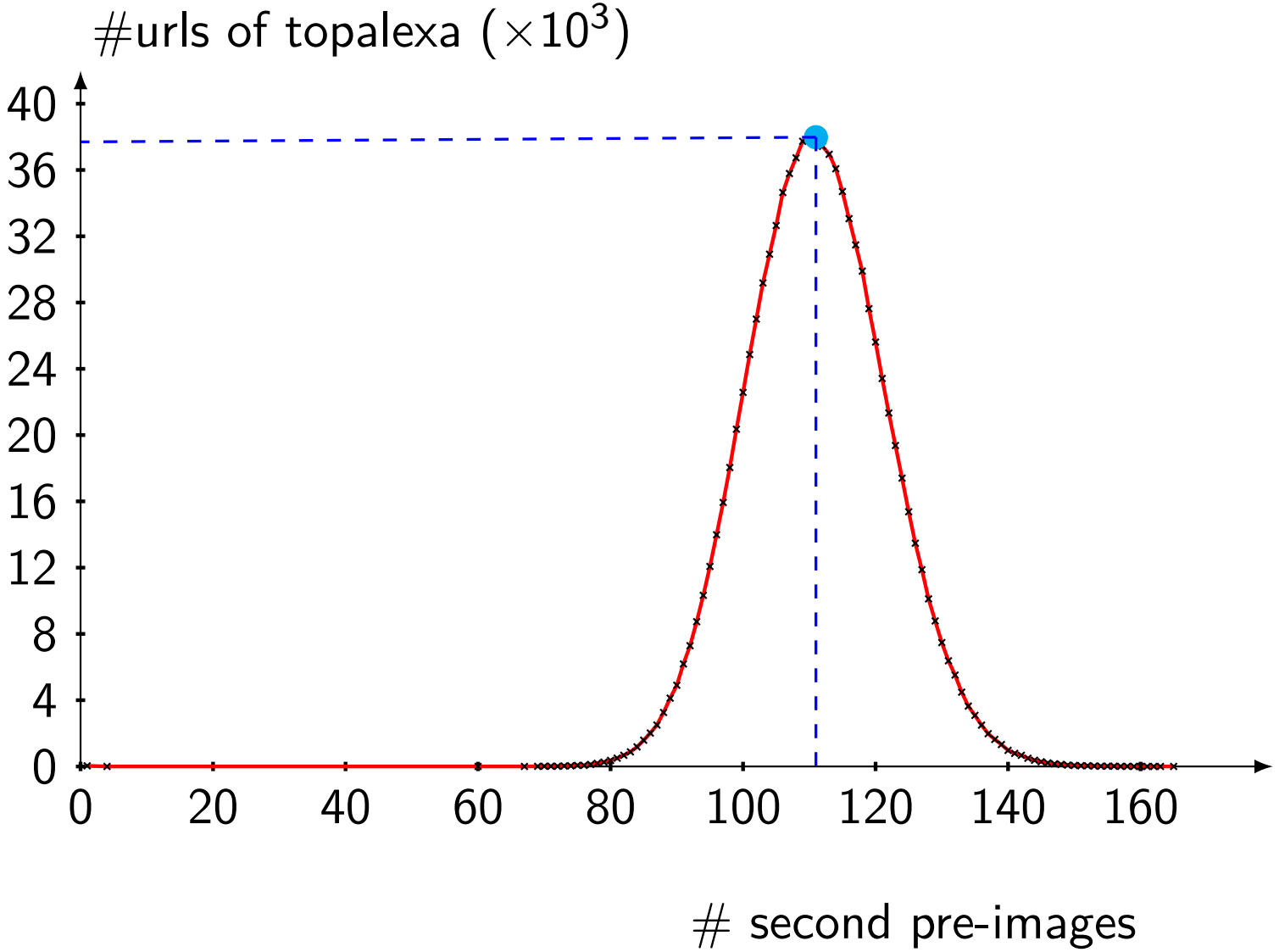
Generating Second pre-images

- Top 10^6 of Alexa

- 1 week of computation on 32 cores :
  - `Python`
  - `fake-factory 0.4.2` $\Rightarrow$ Human readable URLs

Results on TopAlexa

Around 111 Million second pre-images were generated.



Multiple second pre-images

Prefix	#	Alexa Site
0xd8b4483f	165	http://getontheweb.com/
0xbbb9a6be	163	http://exqifm.be/
0x0f0eb30e	162	http://rustysoffroad.com/
0x13041709	161	http://meetingsfocus.com/
0xff42c50e	160	http://js118114.com/
0xd932f4c1	160	http://cavenergie.nl/

Sample URLs :

- http://62574314ginalittle.org/
- http://chloekub.biz/id9352871

URL of Death

malicious URL	popular domain	prefix
deadly-domain.com/tag1/	google.com	0xd4c9d902
deadly-domain.com/tag1/tag2/	facebook.com	0x31193328
deadly-domain.com/tag1/tag2/tag3/	youtube.com	0x4dc3a769

- Generate a tree of URL on the same domain.
- Attacker needs to purchase only one domain.
- Second pre-image search is relatively less parellelizable.

Step 2 : Inclusion

- **Reporting to Google :**
  - google.com/safebrowsing/report_badware/
  - google.com/safebrowsing/report_phish/
- **Reporting to Google's sources :**
  - phishtank.com
  - stopbadware.org
- **Google Webmaster tools.**
- Inclusion is the most difficult part :
  - Ethical reasons.
  - Blackbox implementation on the Google side.

Step 3 : Consequences

- DoS : Increase in traffic towards SB server and its clients.
- **Discount :** 4 bytes sent, 5280 received.

	Amplification
Worst case	8
Average case	800
Best case	1320

- **Bonus :** browser's cache pollution !
- A prefix can only be queried every 45 min.
  - Browser must conserve the list of all corresponding hashes in the cache for 45 min.
  - **Consumes memory !**
- No `botnets` required.
- Clever crafting of malicious URLs.

Outline

Conclusion

- Privacy :
  - Safe Browsing is a useful service.
  - But, privacy policy is incorrect.
  - Has potential to track users.
  - But, no strong evidence.
- Security :
  - Attacks challenge the fundamental design rationale.
  - Challenge : GOOGLE servers are blackbox.
  - White-listing ?

Thank you !
Questions ?