

On the Privacy of Private Browsing

Kiavash Satvat, Matt Forshaw, Feng Hao, Ehsan Toreini

Newcastle University

DPM'13

Introduction

- 2005, Safari first introduced private browsing
- Today, private browsing has become an integrated feature in all major browsers
- How many people use it in the real world?
 - 19% based on a survey (Aggarwal et al, 2010)
 - 2.4 billion Internet users (world stat, 2012)
 - Roughly, 450 millions users of private browsing
- How secure is private browsing?

Threat model

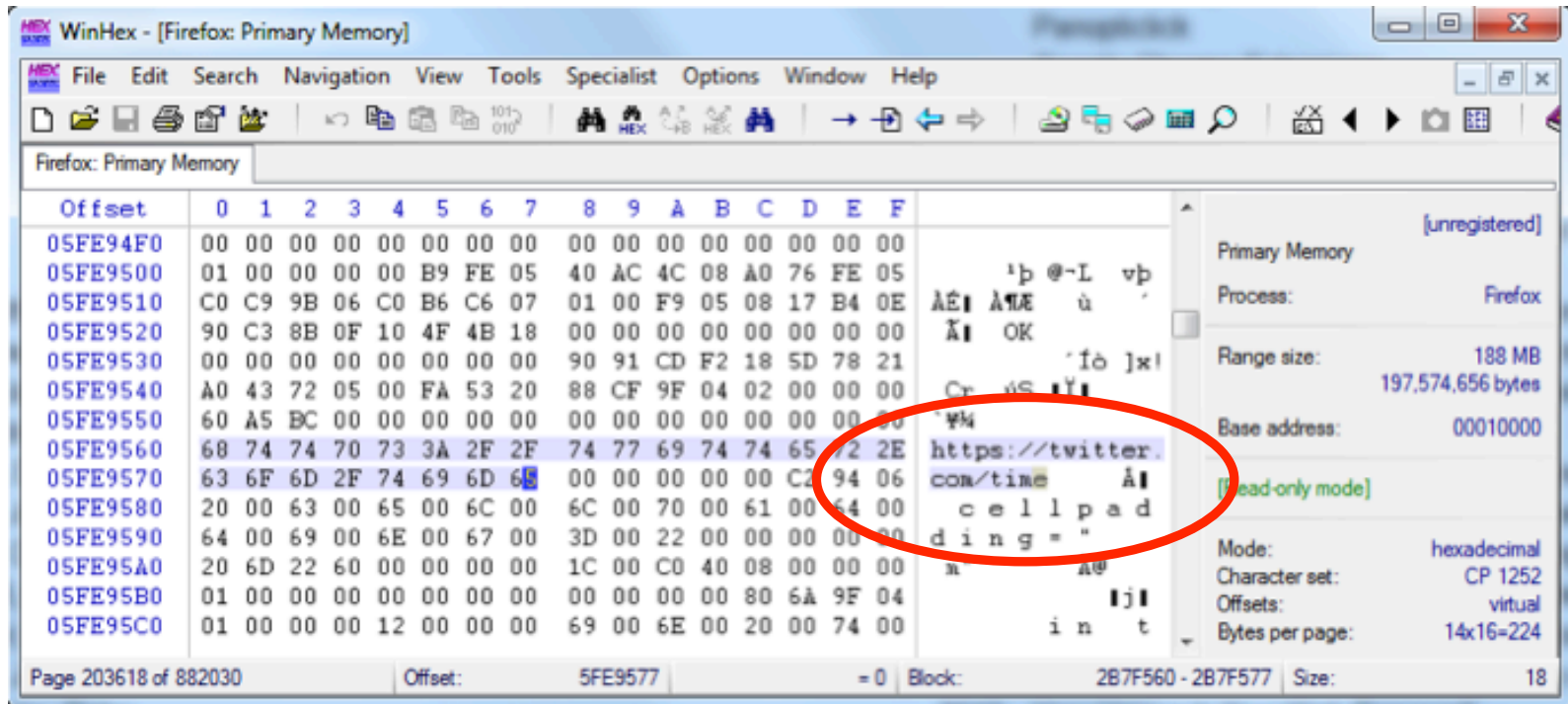
- First, need to define what is meant by “secure”
- Local attacker
 - Capability: full physical access to the computer after private session, but not before
 - Goal: discover any sensitive information related to the private session
- Remote attacker
 - Capability: able to engage with user through http (e.g., news website)
 - Goal: discover if the user is in the private session

Summary of attacks

	Firefox	Chrome	IE	Safari
Domain name system	✓	✓	✓	✓
Memory inspection	✓	✓	✓	✓
File timestamp	—	✓	—	✓
Index.dat *	N/A	N/A	✓	N/A
SQLite database crash *	✓	✓	N/A	✓
SQLite added bookmark *	✓	✓	N/A	✓
Profile recovery *	✓	✓	N/A	N/A
Extension *	✓	✓	—	✓
Cross-mode Interference *	N/A	✓	N/A	N/A
Hyperlink attack	✓	✓	✓	✓
Timing attack *	✓	✓	—	✓

- * **new** results discovered by our work
- We will select only a few attacks to present here

Local attack – memory inspection



- Artefacts about private browsing scattered in memory even after the browser is closed

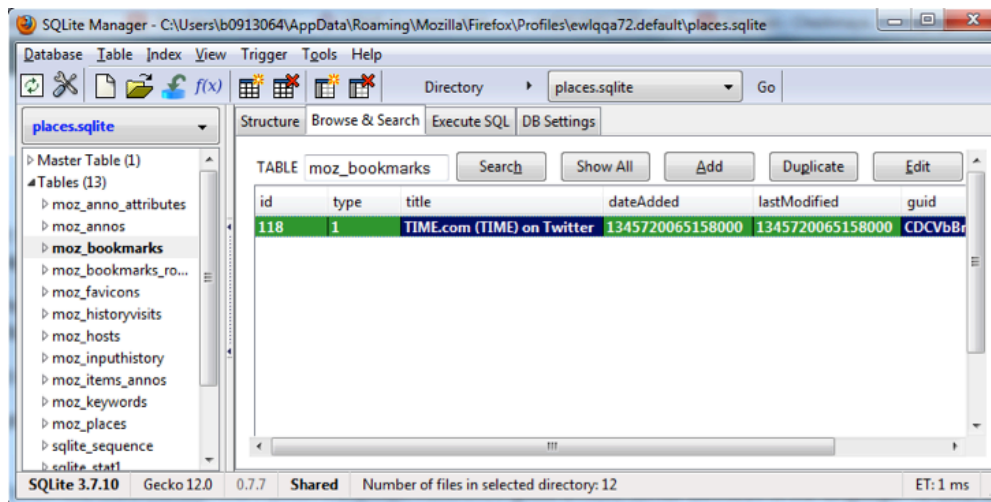
SQLite Database

- SQLite: an open source database used by Firefox, Chrome and Safari to store user profile
- In normal cases, it seems all browsers have removed private browsing records successfully
- However, it is essential to also test **edge cases**:
 - When the browser crashes
 - When the user adds a bookmark

When the browser crashes

- May happen due to overload, manual termination etc
- Firefox (**minor**)
 - WAL files left on disk
 - Indicate occurrence of private browsing and times
- Chrome (**minor**)
 - Journal files left on disk
 - Indicate occurrence of private browsing and time
- Safari (**serious**)
 - Doesn't use in-memory SQLite
 - Inserts records of private browsing and deletes later
 - But in case of crash, private browsing records will persist

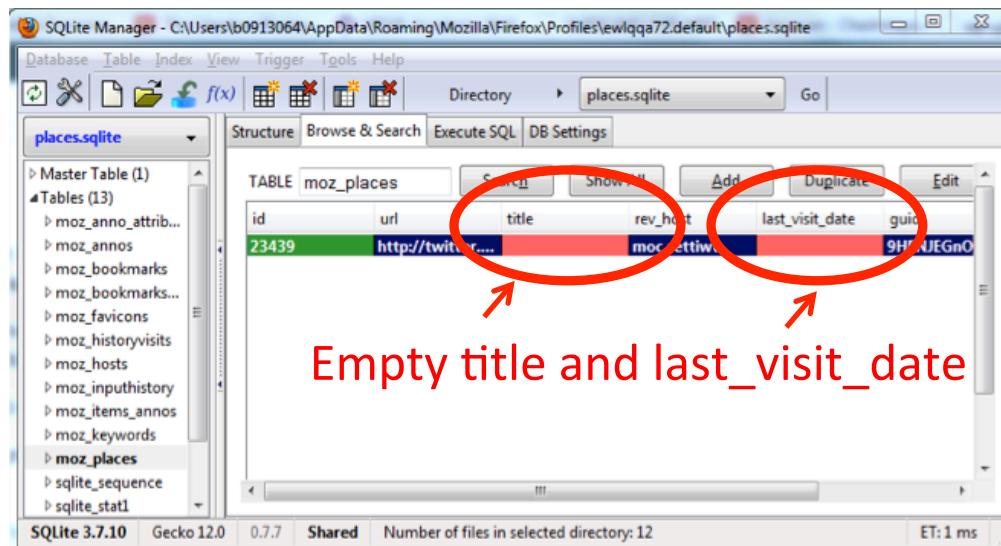
Adding a bookmark (Firefox)



The screenshot shows the SQLite Manager interface with the 'places.sqlite' database selected. The 'moz_bookmarks' table is displayed with the following structure and data:

id	type	title	dateAdded	lastModified	guid
118	1	TIME.com (TIME) on Twitter	1345720065158000	1345720065158000	CDCVbBr

Moz_bookmarks
(table)



The screenshot shows the SQLite Manager interface with the 'places.sqlite' database selected. The 'moz_places' table is displayed with the following structure and data. Red circles and arrows highlight the empty 'title' and 'last_visit_date' fields in the record:

id	url	title	rev_host	last_visit_date	guid
23439	http://twitter.com/...		moz-ttww		9H...UEGn0

Empty title and last_visit_date

Moz_places
(table)

Adding a bookmark (Chrome)

SQLite Manager - C:\Users\A6313193\AppData\Local\Google\Chrome\User Data\Default\History

Database Table Index View Trigger Tools Help

Directory (Select Profile Database) Go

History

Master Table (1)
Tables (9)
downloads
keyword_search_terms
meta
presentation
segment_usage
segments
urls
visit_source
visits
Views (0)
Indexes (12)
Triggers (0)

Structure Browse & Search Execute SQL Profile Directory

TABLE urls Search Show All Add Duplicate Edit Delete

id	url	title	visit_count	typed_count	last_visit_time	hidden	favicon_id
29074	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764372838...	0	0
29075	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764373277...	0	0
29076	http://waitingf...	Waiting for Winter	1	0	1300764374730...	0	0
29077	http://www.fac...	Waiting for Winter	1	0	1300764374730...	0	0
29078	http://www.fac...	Waiting for Winter	2	0	1300764374847...	0	0
29079	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764375895...	0	0
29080	https://mail.go...	Inbox - matthewforshaw@gmail.com...	3	0	1300764391871...	0	0
29081	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764376984...	0	0
29082	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764377571...	0	0
29083	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764385284...	0	0
29084	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764388192...	0	0
29085	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764389148...	0	0
29086	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764390762...	0	0
29087	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764394816...	0	0
29088	https://mail.go...	Inbox - matthewforshaw@gmail.com...	1	0	1300764395000...	0	0
29089	https://mail.go...	Inbox (1) - matthewforshaw@gmail.c...	1	0	1300764416363...	0	0
29090	http://en.wiki...	Bodger & Badger - Wikipedia, the f...	0	0	13007644432...	1	0

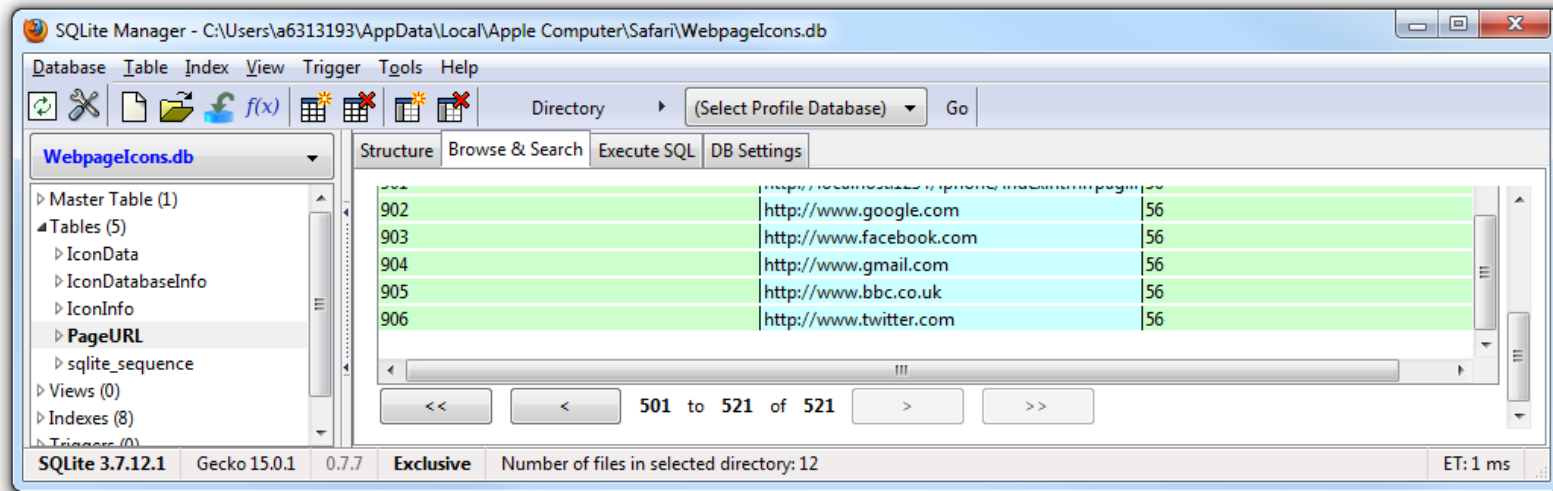
<< < 12801 to 12860 of 12860 > >

SQLite 3.7.14.1 Gecko 19.0 0.7.7 Exclusive Number of files in selected directory: 12 ET: 15 ms

Vist_count = 0

Hidden = 1

Adding a bookmark (Safari)

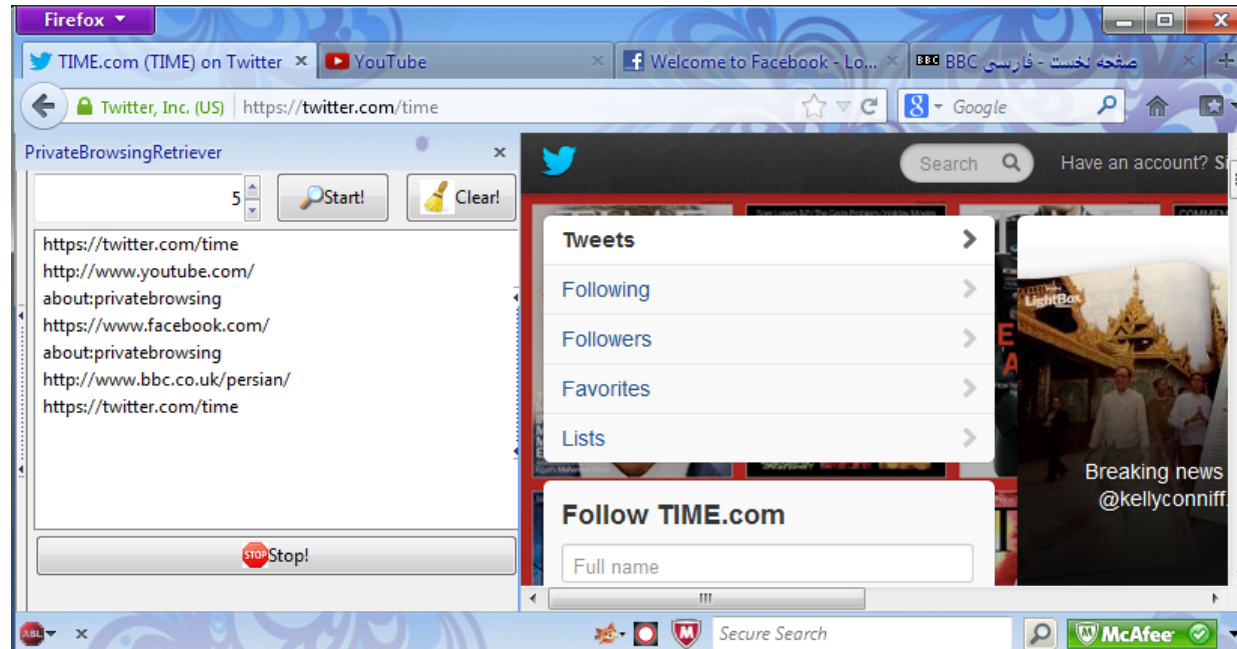


- (**serious**) Once the user adds one bookmark, all websites visited in private mode will persist in the database.
- We filed a bug report (#14685058)
 - 12/08 (Apple): “Engineering has determined that this is not to be fixed.”
 - 13/08, we asked Apple to clarify the decision.
 - 18/08 (Apple): “After much deliberation, engineering has removed this feature.”

Browser extensions

- Browser extensions pose a realistic threat to break privacy of private browsing.
- We tested four latest browsers in 2013
 - Firefox: extension enabled by default (**vulnerable**)
 - Safari: extension enabled by default (**vulnerable**)
 - Chrome: extension disabled by default (good)
 - IE: extensions disabled by default (good)

Firefox extension (proof of concept)



- Records all user activities in private session
- Then sends to a remote server

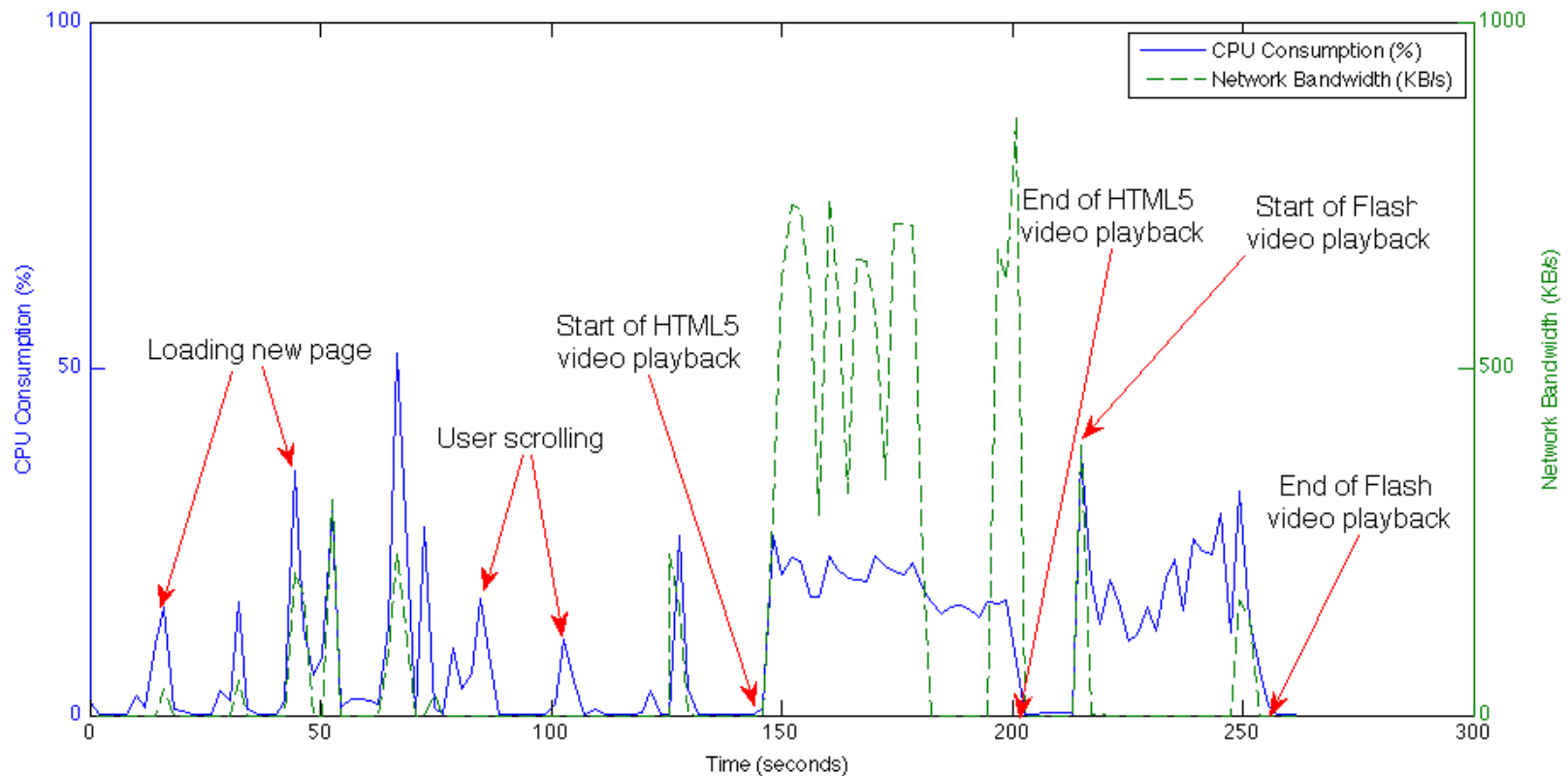
Addressing the threat of extensions

- One straightforward solution is to disable extensions by default in the private mode
- Adopted by Google Chrome and Microsoft IE
- However, we still need to be careful.

Cross mode interference

- Chrome allows two modes to run in parallel
 - Normal mode window: extension **enabled**
 - Private mode window: extension **disabled**
- However, since the two windows share some common resources
- Attacker may exploit cross mode interference

Example of cross mode interference



- Our suggested countermeasure: always run in a single mode

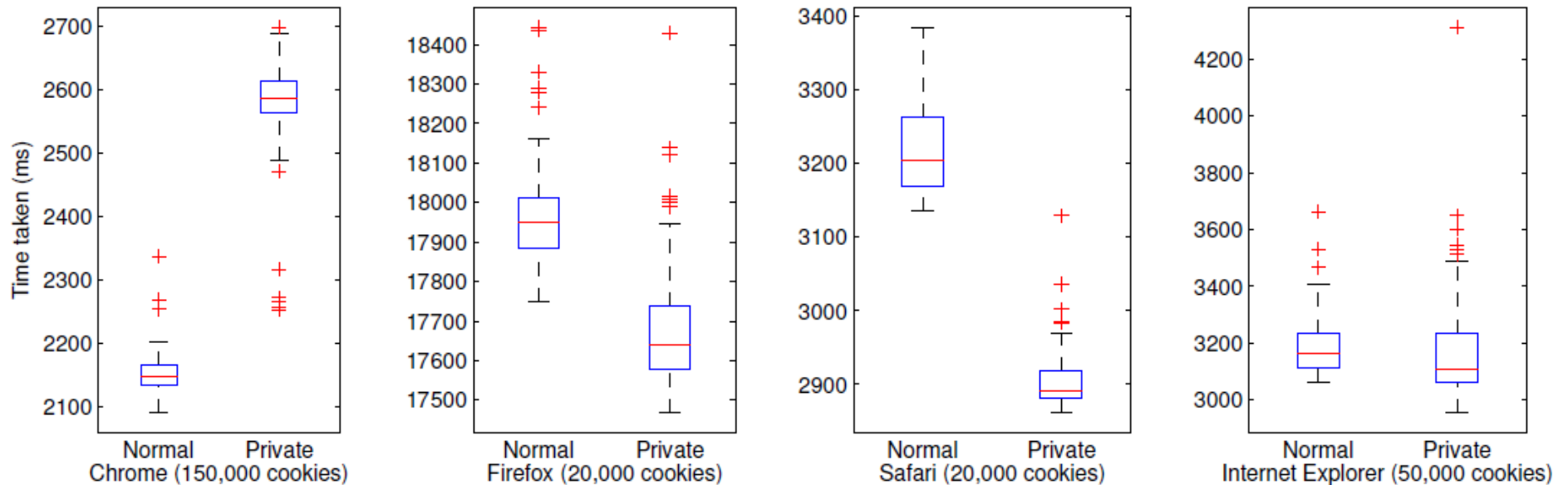
Remote attacks

- Goal of attack: remote website wishes to find out if the user is in the private mode.
- E.g., if the user is in the private mode, remote website may push more adult-oriented content or advertisement.
- Hence, we consider **the fact of using private browsing a privacy feature itself.**

Example: cookie timing attack

- The time it takes to write cookies is different between the usual and private modes.
- We conducted extensive experiments to collect data.

Results (box plots)



- With the exception of IE, the timing difference between the two modes is significant.

Conclusion

- Is private browsing private?
- We took a forensic approach
 - Defined a threat model to define “security”
 - Evaluated against local/remote attacks
 - Validated all previously known attacks
 - Discovered several new attacks
- For further details
 - See the paper and also open source code