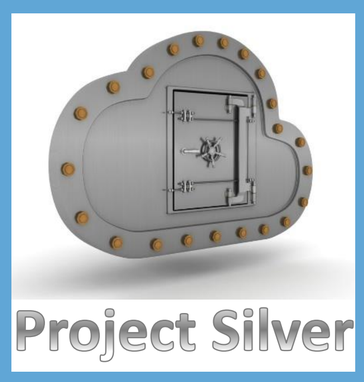


Mitigating Side Channels Using Statistical Privacy Mechanisms

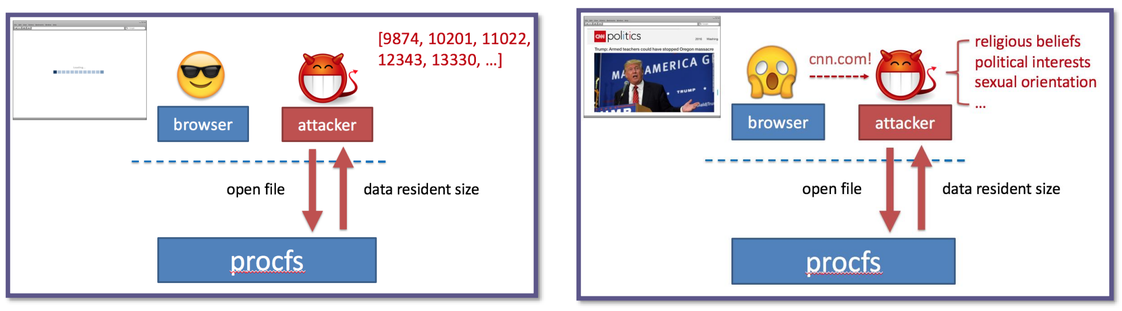
Qiuyu Xiao, Michael Reiter, Yinqian Zhang

University of North Carolina *qiuyu@cs.unc.edu*

Goals

Mitigate side channels in mobile and cloud computing environments while maintaining platform utility

- **Storage side channel**: attacker infers sensitive information from metadata or management data (e.g., resource usage info from the *proc filesystem*)
  - Example: one process infers the web page visited by a browser process by measuring its *data resident size* values repeatedly



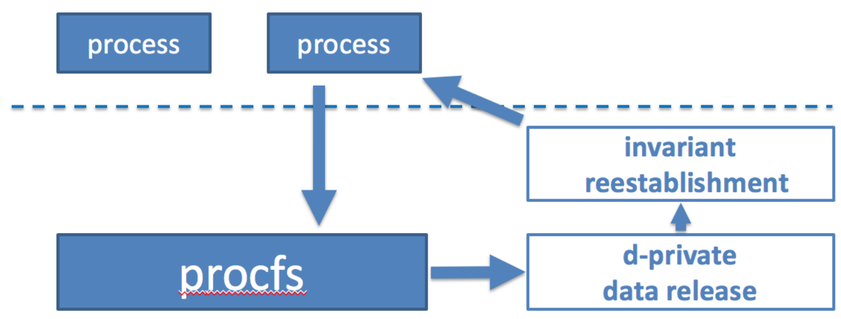
Website Fingerprinting Attack via Storage Side Channel

- **Timing side channel**: attacker infers sensitive data (e.g., crypto keys) by timing events on the computer (e.g., cache accesses to data) using real-time clocks

Approach

First noise data using a *d-private mechanism*, and then *reestablish invariants* to maintain the utility of the data before it is output

- Example design for procfs:

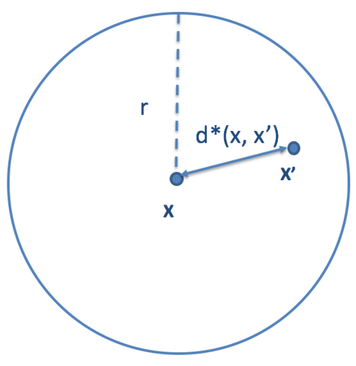


d-privacy

- As the attacker observes noised data, it must infer whether the original data is introduced by the sensitive action or an insensitive one
- M: d-private mechanism, d^*: distance metric
- If underlying X and X' are close ($d^*(X, X') < r$), and ϵ is small, then noised observation $\tilde{X}$ is similarly distributed

$$P(M(X) = \tilde{X}) \leq \exp(\epsilon \times d^*(X, X')) \times P(M(X') = \tilde{X})$$

- As number of observations grows, noise must increase
- Principled design gives *provable* guarantees
- Applies to both storage and timing channels of many types



Invariant reestablishment

- Noising observations will break some applications
- Solution: Reestablish invariants on noised values, so they appear as unnoised values
- Invariant reestablishment does not erode the privacy achieved by the d-private mechanism
- Example invariants from procfs:

One-field Invariants	Multiple-field Invariants
totalVM ≥ 0	totalVM ≥ sharedVM
utime[i] ≥ utime[i − 1]	hiwaterVM ≥ filePages
starttime[i] = starttime[i − 1]	execVM ≥ filePages + swapEnts

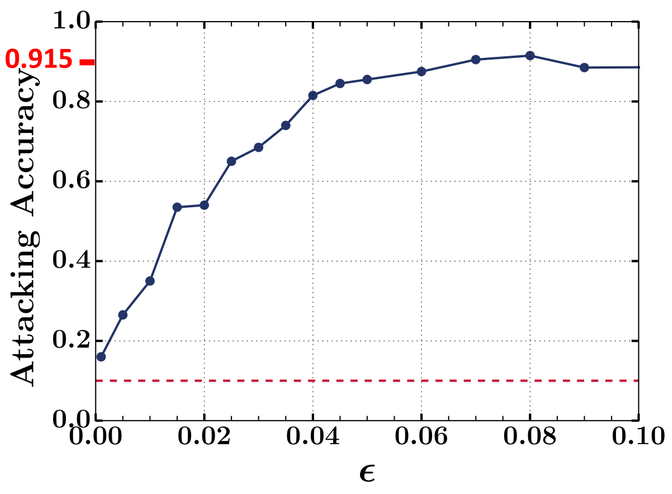
Implementation

- Implemented prototype to address storage side channels in procfs on Linux
- Prototype is implemented in Ubuntu 14.04 with kernel version 3.11.
- d-private mechanism is implemented as a kernel routine
- Invariant reestablishment functionality is implemented in a user-space daemon

Results

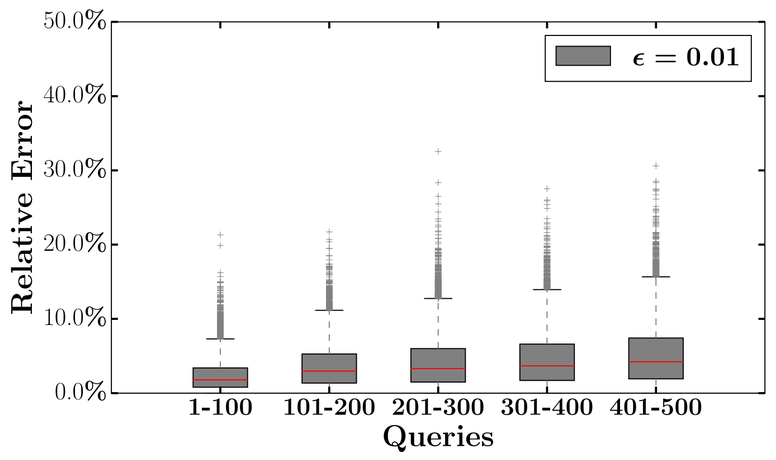
Example security eval:

- Infer the web page (from a set of ten web pages) visited by the browser based on its *data resident size*
- Without protection, the attacking accuracy is 0.915



Utility evaluation:

- Relative error measured for the *data resident size* when ϵ is set to 0.01
- The relative error is less than 10% most of the time



- Rank accuracy by the *top command* based on the *resident size field* when ϵ is set to 0.01
- The utility of *top* is maintained