

## SIDE-CHANNEL-LEAKS IN WEB APPLICATIONS: A REALITY TODAY, A CHALLENGE TOMORROW


Shuo Chen      Rui Wang, XiaoFeng Wang and Kehuan Zhang








IEEE Symposium on Security and Privacy  
Oakland, California  
May 17<sup>th</sup>, 2010

### PC App vs. Web App

**Traditional PC application** 

**Web application** (1) split between client and server  
(2) state transitions driven by network traffic

Worry about privacy?  
Let's do encryption.

### Side-Channel Leaks

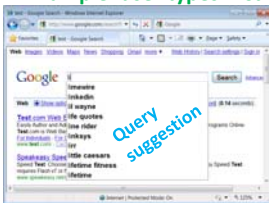
- The eavesdropper cannot see the contents, but can observe :
  - number of packets, timing/size of each packet
- Previous research showed privacy issues in various domains:
  - SSH, voice-over-IP, video-streaming, anonymity channels (e.g., Tor)
- Our motivation and target domain:
  - target: today's web applications
  - motivation: Software-as-a-Service (SaaS) becomes mainstream, and the web is the platform to deliver SaaS apps.


### Our Main Findings

- Surprisingly detailed user information is being leaked out from several high-profile web applications
  - personal health data, family income, investment details, search queries
  - (Anonymized app names per requests from related companies)
- The root causes are some fundamental characteristics in today's web apps
  - stateful communication, low entropy input and significant traffic distinctions.
- Defense is non-trivial
  - effective defense needs to be application specific.
  - calls for a disciplined web programming methodology.

### Google bing YAHOO!

Scenario: search using encrypted Wi-Fi WPA/WPA2.  
Example: user types "list" on a WPA2 laptop.





Query suggestion

Attacker's effort: linear, not exponential.  
Consequence: Anybody on the street knows our search queries.

### OnlineHealth<sup>A</sup> ("A" denoting a pseudonym)

- A web application by one of the most reputable companies of online services
- Illness/medication/surgery information is leaked out, as well as the type of doctor being queried.
- Vulnerable designs
  - Entering health records
    - By typing – auto suggestion
    - By mouse selecting – a tree-structure organization of elements
  - Finding a doctor
    - Using a dropdown list item as the search input

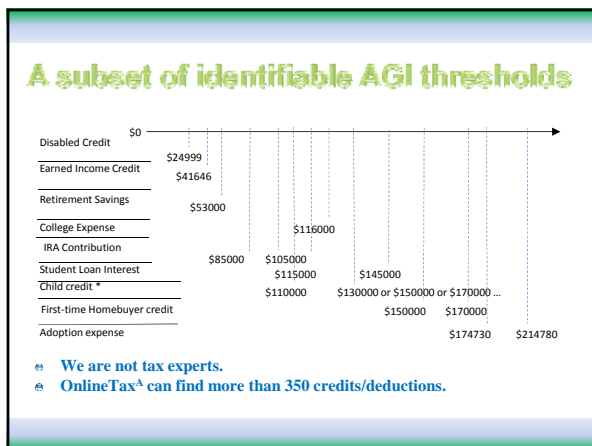
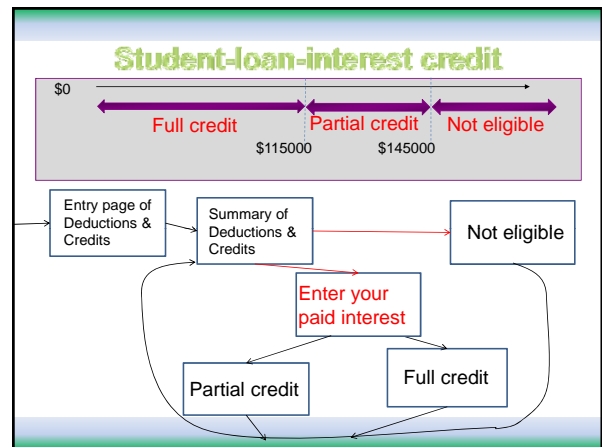
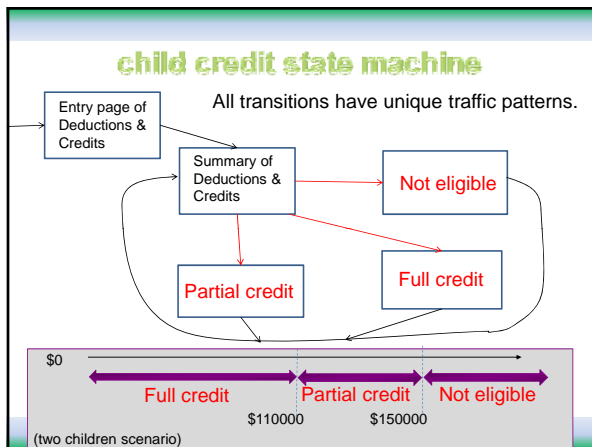
## Attacker's power

Entering health records: no matter keyboard typing or mouse selection, attacker has a **2000x ambiguity reduction power.**

Find-A-Doctor: attacker can uniquely identify the specialty.

## OnlineTax<sup>A</sup>

- It is the online version of one of the most widely used applications for the U.S. tax preparation.
- Design: a wizard-style questionnaire
  - Tailor the conversation based on user's previous input.
- The forms that you work on tell a lot about your family
  - Filing status
  - Number of children
  - Paid big medical bill
  - The adjusted gross income (AGI)

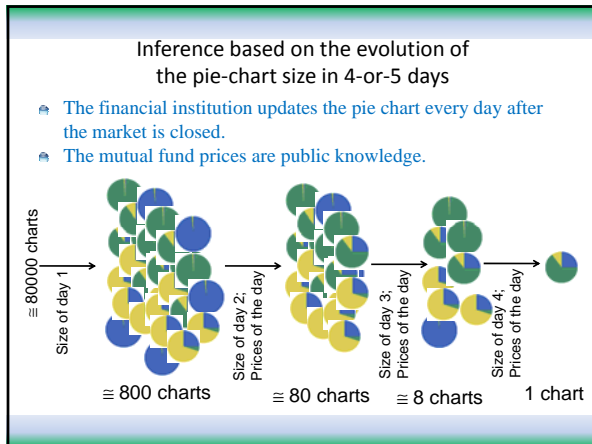


## OnlineInvest<sup>A</sup>

A major financial institution in the U.S.

Which funds you invest?

- No secret.
- Each price history curve is a GIF image from MarketWatch.
- Everybody in the world can obtain the images from MarketWatch.
- Just compare the image sizes!

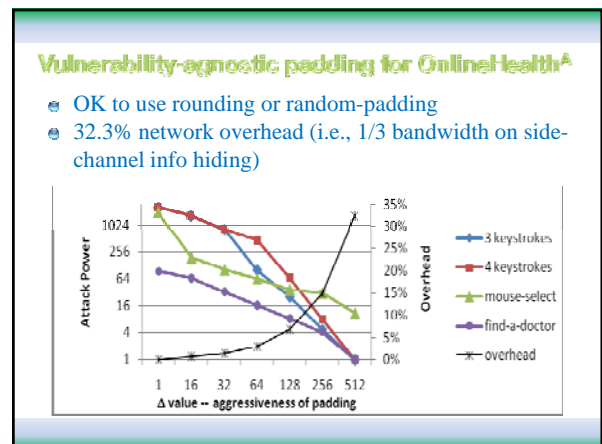


Root causes: some fundamental characteristics of today's web applications

- ### Fundamental characteristics of web apps
- Significant traffic distinctions
    - The chance of two different user actions having the same traffic pattern is really small.
    - Distinctions are everywhere in web app traffic. It's the norm.
  - Low entropy input
    - Eavesdropper can obtain a non-negligible amount of information
  - Stateful communication
    - Many pieces of non-negligible information can be correlated to infer more substantial information
    - Often, multiplicative ambiguity reduction power!

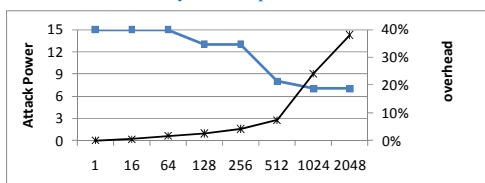
Challenging to Mitigate the Vulnerabilities

- ### Why challenging?
- ☉ Traffic differences are everywhere. Which ones result in serious data leaks?
    - ☉ Need to analyze the application semantics, the availability of domain knowledge, etc.
    - ☉ Hard.
  - ☉ Is there a vulnerability-agnostic defense to fix the vulnerabilities without finding them?
    - ☉ Obviously, padding is a must-do strategy.
      - ☉ Packet size rounding: pad to the next multiple of  $\Delta$
      - ☉ Random-padding: pad  $x$  bytes, and  $x \in [0, \Delta)$
    - ☉ We found that even for the discussed apps, the defense policies have to be case-by-case.



### Vulnerability-agnostic padding for OnlineTax<sup>A</sup>

- Neither rounding nor random-padding can solve the problem.
  - Because of the asymmetric path situation



### Vulnerability-agnostic padding for Google search

- Rounding is not appropriate, because
  - Google's responses are compressed.
  - The destination networks may or may not uncompress the responses
    - E.g., Microsoft gateways uncompress and inspect web traffic, but Indiana University does not.
    - rounding before the compression → Indiana Univ. still sees distinguishable sizes;
    - rounding after the compression → Microsoft still sees distinguishable sizes

### Vulnerability-agnostic padding for OnlineInvest<sup>A</sup>

- Random padding is not appropriate, because
  - Repeatedly applying a random padding policy to the same responses will quickly degrade the effectiveness.
    - Suppose the user checks the mutual fund page for 7 times, then
      - 96% probability that the randomness shrinks to  $\Delta/2$ .
  - OnlineInvest<sup>A</sup> cannot do the padding by itself
    - Because the browser loads the images from MarketWatch.

### SaaS and Cloud-Computing



Need to develop a disciplined methodology for side-channel-info hiding

### Conclusions

- Side-channel-leaks are a serious threat to user privacy in the era of SaaS.**
- Defense must be vulnerability-specific, and thus non-trivial.**
- Call for future research on the programming practice for protecting online privacy.**

### Acknowledgements

- Ranveer Chandra – guidance on Wi-Fi experiments
- Cormac Herley – suggestion about using the pie-chart evolution in multiple days
- Emre Kiciman – Insights about the HTTP protocol
- Johnson Apacible, Rob Oikawa, Jim Oker and Yi-Min Wang