Goal: detect private information leaving the device
- E.g. GPS position, contacts, photos

Approach: Track the origin of each byte of data

Each data byte $b$ has an associated set $S_b$ defining sources its value is derived from

Alert when data from sensitive sources is sent out
- Sink: network sockets
TaintDroid

For the most part, taint tag propagation proceeds as one might expect. Instructions always overwrite the destination register; therefore, unary operations set the taint tag of the destination register to that of the source register, and binary operations (e.g., $a = b + c$) set the taint tag of the destination register to the union of the taint tags of the two source registers (e.g., $\tau(a) \leftarrow \tau(b) \cup \tau(c)$). For the implementation, the union is simply a bitwise OR of the taint tag bit vectors. However, there are several cases where the taint propagation is not straightforward (e.g., for array indexes and object references). A full propagation logic and discussion is provided in our original paper. 9
Keep track of security level of each process in system
- Current and maximum level

Labels form a lattice

When process attempts to read data from source:
- Allow if source-level ≤ max-level
- (HWM) Increase current level to source-level ∨ current-level
BLPDroid

- Process = app
- Labels:
  - G = GPS
  - P = Photo
  - C = Contacts
- Network sink label: { }
CSE 127: Computer Security
Advanced Web Security

Kirill Levchenko

November 16, 2017
Same Origin Policy

- Code can only access data from the same origin
  - Origin: (scheme, domain, port) triple
  - E.g. (http, cseweb.ucsd.edu, 80)

- Typical context: JavaScript executing in browser
  - Main page loads iframe element with content from another site
  - JavaScript from main page cannot access data inside iframe

- Why do we need the same origin policy?
Same Origin Policy

- **Want:** Combine user-specific content from distinct sites
  - iframes, XMLHttpRequest, embedded Java, embedded Flash
- User does not trust sites with others’ data
- Sites (origins) do not trust each other
- Sites must be isolated from each other
SOP as Multilevel Security

- All content (code and data) carries its own label (origin)
  - Can change one’s origin to less specific domain only
- Can write via HTTP request to any origin
- Can read only from same origin
- Can get and set cookie only for same origin

- What is SOP protecting?
  - Secrecy? Integrity? Availability?
Domain Hierarchy

document can change its own origin to less specific domain
SOP as Multilevel Security

- What if Web used BLP model?

- Labels (origins) don’t form a lattice
  - No label that contains both ucsd.edu and google.com
  - No way to specify more than one origin in content

- Must declassify to send to another origin
  - Both sender and receiver must declassify to common domain
From the Aether to the Ethernet

Yossef Oren and Angelos D. Keromytis
USENIX Security 2014
ETHERNET

ANTENNA INPUT

(slides source: USENIX Security 2014 presentation)
Smart TV

- Can watch broadcast TV
- Can watch streaming content
  - Netflix, YouTube, Hulu, etc.
- Can browse Web

TV-era non-interactive content
Web-era interactive content
HbbTV

- Web content delivered in broadcast television stream
  - HTML, JavaScript
- Content overlaid on broadcast
- User can interact with content
"Tout le monde aime la Chine"

Les invités du jour :
- Pascal Blanchard : Historien, spécialiste du fait colonial.
- Jean-Paul Tchang : Spécialiste de l'économie chinoise.
- Dorian Malovic : Chef du service Asie au quotidien La Croix.

17h45 - 18h45
- JavaScript application starts executing automatically when user starts watching channel
- Application can access content from broadcast stream or Internet (via broadband)
HbbTV Security

- Broadcast stream is not authenticated
  - Authentication would require extensive non-existent PKI
- No way to disable application
HbbTV Security

- Broadcast and broadband content together in one place
- Need a security policy to restrict interaction
  - Can broadband content access broadcast content?
  - Can broadcast content access broadband content?
  - Can broadband content access other broadband content?
- HbbTV uses same origin policy
- Broadcast content explicitly specifies its own origin
HbbTV SOP

- Broadcast content explicitly specifies its own origin
- Broadcast application can say “I am facebook.com” and TV browser will allow it to access facebook.com content
- This completely nullifies same origin policy
- Huge security problem
  - Exploitation requires unscrupulous broadcaster
  - Anyone with TV-band transmitter
 Sorry seems to be the hardest word

Police, Rams duel over apology

The latest Ferguson confrontation isn’t on the city’s streets, but between St. Louis County police and the St. Louis Rams. FULL STORY

• NEW Brown’s stepfather investigated
• Students across U.S. hold ‘die-ins’
• 11 times athletes took a stand
• The Ferguson issue that wasn’t
• Opinion: Racial justice? Try this

THE LATEST
• Report: ISIS leader’s wife, son detained
• Prince Harry reveals his secret
• Deaths reveal new world of drug dealing

THE CNN GUNS PROJECT

A girl's gun club

Julianna Crowder has guns — lots of them — and she knows how to use them. She’s making sure other women do, too. FULL STORY
Iframes

- Frame contains content from another origin
  - Origin’s cookies sent when retrieving content
- Framed content cannot be accessed by enclosing document (same origin policy)
- Displayed to user in style specified by enclosing page
- User can interact with framed content
Iframe Security

- Same origin protects direct access by enclosing page
- But enclosing page controls layout and presentation
  - Page position and CSS
- Potential for enclosing page to manipulate presentation of framed content to mislead user
Clickjacking

- **Clickjacking**: Inducing user to click an element of page having side effects that the user does not realize

- Using iframes: obscure UI element in frame so user does not understand what they are clicking on
  - Can also make iframe completely transparent
Likejacking

Prove you’re human, click on “blue”
Likejacking
Clickjacking Defenses

- Don’t allow transparent iframes?
  - How transparent is transparent?
  - Can magnify and show 1 pixel of content.

- Don’t allow resizing iframes?
  - Load lots of identical frames

- Don’t allow frames?
  - Web will stop working
Clickjacking Defense

- **Framebusting:** go to framed page ("busting out" of frame)
- Could also prevent content from rendering in iframe
- Preferred defense against clickjacking
- **Simple fragment**
  http://en.wikipedia.org/wiki/Framekiller

  
  <script type="text/javascript">
  if(top != self) top.location.replace(location);
  </script>
How It Happened

- Clickjacking defense straightforward once you know it
- But why is clickjacking a problem in the first place?
- Who is to blame for it?
How It Happened

- Initially Web pages were just static content
- Persistent sessions allowed interactive sites
- Original frames (using `<frame>`) allowed split-pane sites
  - Very ugly; will not show screenshot example
  - User still clearly interacting with original site
- Iframes: site embedded in page
  - A better, much less ugly frame
How It Happened

- **Original security assumption:** users can see site with which they are interacting
- A series of small feature additions erode assumption
  - Each apparently innocent
- Where did we go wrong?
- How do we prevent it?