HAILS
Protecting Data Privacy in Untrusted Web Apps
Amit Levy
with Deian Stefan & David Mazieres
Stanford SCS Lab

- Formal work
- Language support for IFC
- Side-channels
- Mostly focused on building systems
  - tcpcrypt, HiStar, Cinder, CoralCDN...
- Today - Hails
Our Private Data is Everywhere
It’s Connected!

GitHub: Develop

Google Health

Basecamp API

Foursquare Developers
This is great!

- Richer experience
- Mashups
- Personalization
- Cheap to innovate
- Lots of options for consumers
But it’s also a problem

- Can’t enforce policies on other applications
- Must resort to coarse grain access control
  - Yelp *can* access my Facebook data and do it’s bidding
  - or Yelp *cannot* access data at all
- Over-share but under-deliver
  - Forced to choose between privacy and features
  - Get neither
Web APIs Today

Amazon

Zynga

Picnik

Rackpace

GitHub

ACME Datacenter

CorpBook API Server

CorpBook DB

CorpBook Front-end

Limited

Full

Jeff from HR
What about centralizing?

ACME Datacenter

GitHub

CorpBook

Picnik

CorpBook DB

Jeff from HR
Centralizing Not Enough

GitHub
CorpBook
Picnik

ACME Datacenter

Jeff from HR

Jane from sales

Full
Limited
Centralizing Not Enough
How to reduce unemployment with...

How to cure cancer with...

How to alleviate hunger with...

How to protect web data privacy with HAILS
What is HAILS?

• Multi-Application Web Platform
• Information Flow Control (IFC) to enforce policies on data
• Leverages LIO framework in Haskell
• DCLabels for policies
• Enforces fine-grained policies on untrusted apps with high performance
Goal

To allow untrusted web applications access to users’ entire data while ensuring that they do not violate policies set on that data and without sacrificing functionality.
Why IFC?

• Today’s web policies restrict what data apps can see or who they can connect to
• Real concerns relate to where data can flow
• Replace “Picnik can see my photos because I trust Picnik not to show them to my boss.”
• With “Any app can see my photos as long as my boss doesn’t see them.”
HAILS Web Stack

Hails

Hails Jail

GitHub

Picnik

Zynga

LIO

DCLLabel

Haskell Runtime

Database
HAILS Architecture

- Picnik
- CorpBook
- GitHub
- CorpBook DB

Jeff from HR
HAILS Architecture

- Picnik
- GitHub
- CorpBook
- CorpBook DB

Jane from sales

[HAILS to Picnik]
[GitHub to CorpBook]
[CorpBook DB to CorpBook]
[Jane from sales]

[Red 'X']
DCLLabels

- Disjunction Category Labels
  - (“amit” or “deian” or “david”)
  - Matches the kinds of policies we want to express in the web
- Powerful enough to express today’s policies:
  - (“amit” or “deian” or “david”) and “corpbook”
DCLabels in Action

GitHub

Picnik

GitHub

HAILS

CorpBook DB

Lisa from Mgmt.

Jeff from HR

Jane from sales

("Jeff" or "Lisa") and "Picnik"
GitStar

• An extensible social code hosting application
• Like GitHub but **better!™**
• How would we build if from many small mutually distrustful components
• Project management, issues, messaging, newsfeed, wiki etc’ are all separate apps
• They rely on each other’s data
• Launch by April 11th!
Check us out!

- [http://github.com/scslab](http://github.com/scslab) (soon to be at gitstar.com!)
- hails
- gitstar
- dclabel
- lio
Thanks! Questions?
Lifetime of a request

- TCB accepts HTTP request
- HAILS login
- Proxy to app with clearance based on user
- Label starts low
- Reading from database raises label
- Label check by HAILS on HTTP response
Hails DB Model

Database ("gitstar")

DBLabel - (ALL) (ALL)

"messages"
(ALL) (ALL)

"news feeds"
(ALL) ("gitstar")

"projects"
(ALL) ("gitstar")

(ALL) ("#linux" ∨ "gitstar")

(ALL) ("#hails" ∨ "gitstar")

("#ms_dos" ∨ "gitstar") ("#ms_dos" ∨ "gitstar")
Hails Policy Modules

• Policy modules moderate unlabeled DB with labeled apps
• Each policy module “owns” a single database
• Transforms unlabeled MongoDB documents (JSON)
• Defines which collections are available
Hails Policy Modules

\[
\begin{align*}
\text{lcollections} &= \text{newDC} (\langle\rangle) ("gitstar" :: \text{String}) \\
\text{lpub} &= \text{newDC} (\langle\rangle) (\langle\rangle)
\end{align*}
\]

\[
\begin{align*}
\text{projectsCollection} &:: \text{DC (Collection DCLabel)} \\
\text{projectsCollection} &= \text{collection} \ "\text{projects}\" \text{lpub lpub} \$
\begin{align*}
\text{RawPolicy} (\langle\text{doc} \rightarrow \text{newDC} (\langle\rangle) ("#" ++ \text{doc} ! "name" .\/. "gitstar")\rangle) \\
[("name", \text{SearchableField}) \\
("repo", \text{FieldPolicy} (\langle\text{doc} \rightarrow \text{newDC} ("#" ++ \text{doc} ! "name" .\/. "gitstar") (\langle\rangle)\rangle))]
\end{align*}
\]

\[
\begin{align*}
\text{configDB} &:: \text{DBConf} \rightarrow \text{DC (Database DCLabel)} \\
\text{configDB \ conf} &= \text{do} \\
\text{db} &\leftarrow \text{labelDatabase \ conf \ lcollections \ lpub} \\
\text{let \ priv} &= \text{dbConfPriv conf} \\
\text{myUsersCollection} &\leftarrow \text{usersCollection} \\
\text{assocCollectionP priv myUsersCollection db}
\end{align*}
\]