e-privacy XVIII *(2015)*

App, Store e società: permessi e profilazione

Antonio Langiu
Nexa Center for Internet & Society,
Politecnico di Torino (DAUIN)

Cagliari, 16 Ottobre 2015

Struttura presentazione

- Concetti su App e Store
- In-App Advertising
- Analisi dei dati inviati dalle App

Concetti su App e Store

Sandbox

- In informatica con **sandbox** si intende un meccanismo di sicurezza per isolare ed **eseguire codice non fidato**. Questo viene fatto eseguendo l'applicativo in un ambiente ristretto e controllando le risorse esterne a cui esso ha accesso.
- Nei sistemi Linux un concetto di Sandbox viene implementata con **Selinux** o **Apparmor**

Implicazione utente: i permessi

- Nei modelli basati sule App, l'utente può concedere o negare una serie di permessi
- Chi sviluppa un'App dichiara di quali permessi questa avrà bisogno
- L'utente può concedere i permessi
  - in maniera statica
  - in maniera dinamica

Confronto App Store

- Il confronto è stato fatto sia nei sistemi desktop che nei sistemi mobile
- Android:
  - permessi rilasciati in maniera statica (Take it or leave it decision)
  - alta granularità
- iOS
  - permessi rilasciati in maniera dinamica
  - bassa granularità

Implicazione sulla privacy

- Che tipo di dati può raccogliere un App?
  - rubrica
  - galleria fotografica
  - elenco chiamate
  - messaggi
  - dati sulla salute (HealthKit iOS)

A chi interessano questi dati?

- Sviluppatori:
  - Advertisement
  - Analytics
  - Tracking e crash-reporting
- Advertiser
  - Cross-app tracking
  - Targeted advertising

In-App Advertising

They don’t need to know your name…

“To approach individuals with customized advertising, you have to know who they are. Or at least, you have to gather enough personal information about them that their identity could be easily figured out.”

“They don’t need to know your name to know who you are”

Online Ads vs. Privacy, The New York Times, May 2007 http://nyti.ms/1PvycxH

Come vengono implementate le pubblicità nelle App?

- Lo sviluppatore si registra sul sito dell'Advertiser
- Scarica una libreria e la inserisce nel progetto
- Questa libreria viene fornita precompilata, per cui lo sviluppatore non sa cosa essa faccia realmente
- La libreria per funzionare può aver bisogno di permessi aggiuntivi che lo sviluppatore deve occuparsi di ottenere

Cross-App Tracking

- Le librerie pubblicitarie, per migliorare il servizio offerto, hanno bisogno di raccogliere dati da diverse app, in modo da poter profilare al meglio l'utente
- Per fare questo hanno bisogno di identificarlo univocamente, questo viene fatto identificando il dispositivo mediante un identificatore unico

Identificatori iOS

- UDID (Unique device identifier)
- OpenUDID
- IDFA (Identifier for Advertisment)
- IDFV (Identifier for Vendor)
- OpenIDFA

iOS Dev. Program License Agreement

"You and Your Applications (and any third party with whom you have contracted to serve advertising) may use the Advertising Identifier, and any information obtained through the use of the Advertising Identifier, only for the purpose of serving advertising. **If a user resets the Advertising Identifier, then You agree not to combine, correlate, link or otherwise associate, either directly or indirectly, the prior Advertising Identifier and any derived information with the reset Advertising Identifier."**

Fingerprint del dispositivo

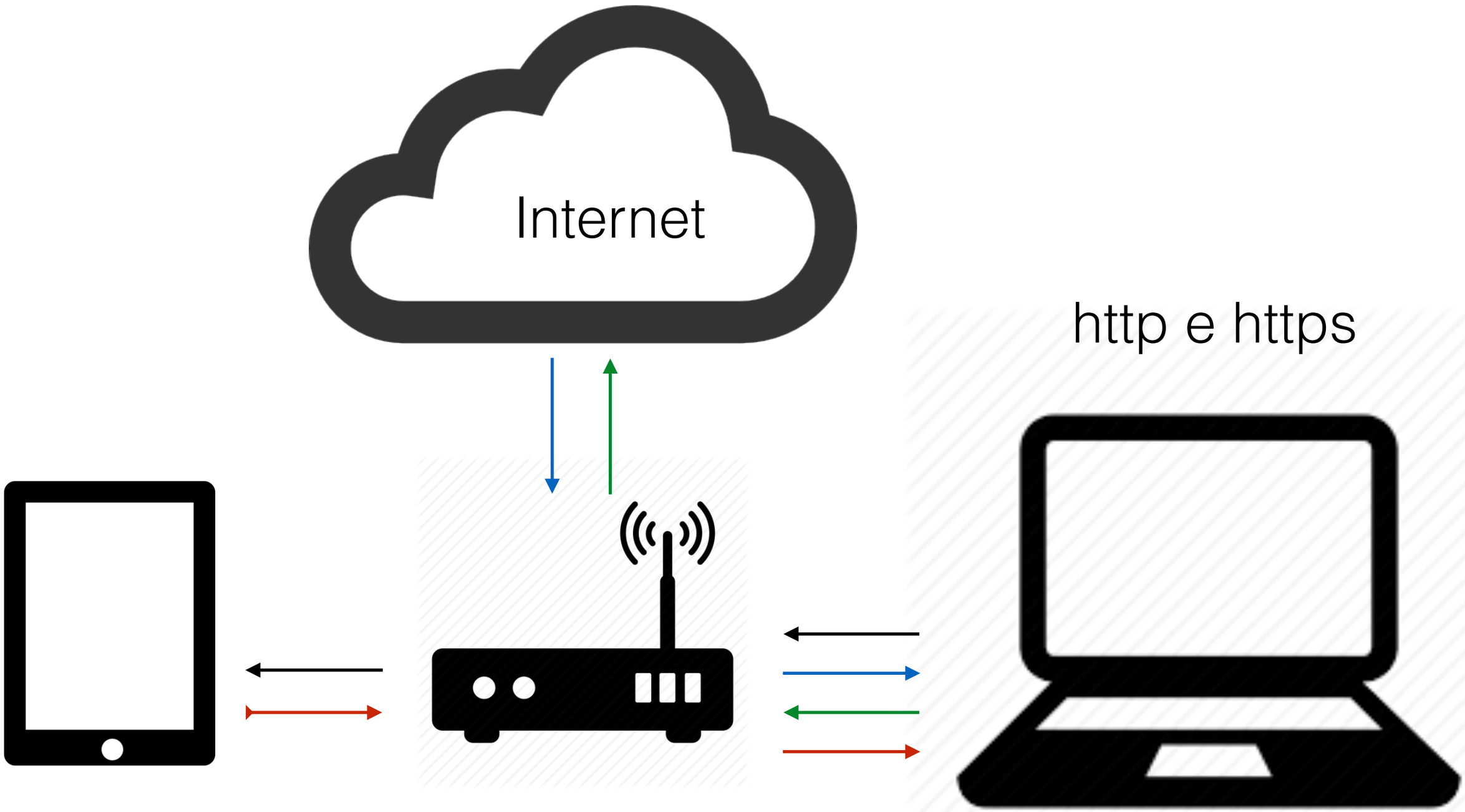
- Si intende l'operazione di generazione di un codice "unico" e persistente per un dispositivo
- I dispositivi attuali hanno una moltitudine di sensori con caratteristiche uniche
  - Fotocamera -> Noise Pattern
  - Accelerometro -> Configuration values (esposto al browser senza nessun permesso)
- Web -> HTTP Cookies
- Web -> Canvas Fingerprinting

Analisi dei dati inviati dalle App

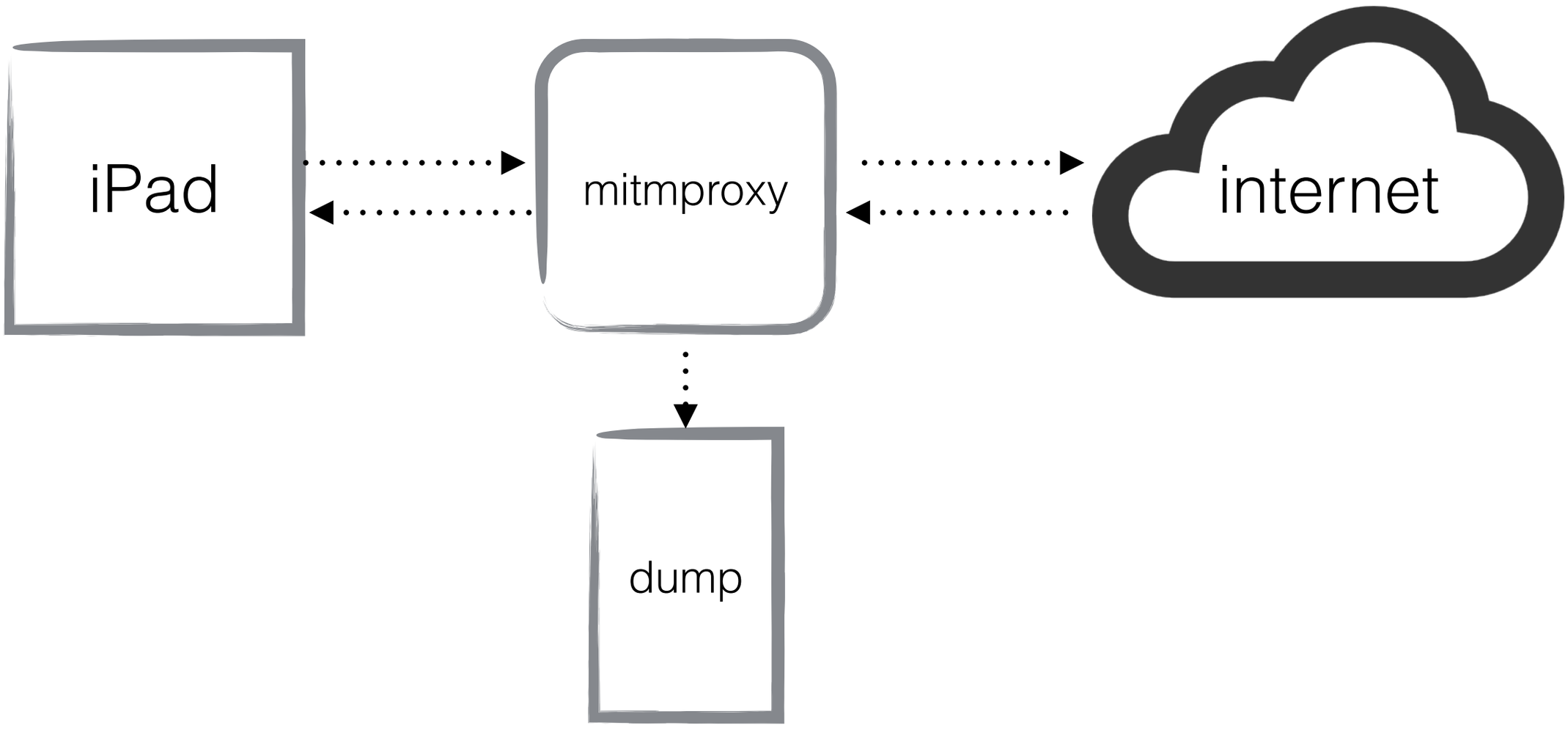
Raccolta dati con *mitmproxy*

- Per comprendere il funzionamento delle librerie pubblicitarie si è utilizzato un software chiamato **mitmproxy** per intercettare e salvare i dati inviati dalle App (sia via HTTP, sia via HTTPS)

Funzionamento mitmproxy



Funzionamento mitmproxy



Esempio di pacchetto analizzato

"app": "542576fcc26ee42ca7e405ed","identity": "930000000a7575696400026

"bundle": "17",

"bundle_id": "com.gramgames.1010",

"carrier": {

"carrier-name": "3 ITA", "iso-country-code": "it", "mobile-country-code": "222", "mobile-network-code": "99"

},

"country": "IT", "device_dimensions": {

"height": 1024,

"width": 768 },

"device_family": "Universal",

"device_type": "iPad3,3",

d6163696400290000003266353637326362373636393162393839626264323032326135333439393339613264376239353200026966610021000000356562623937633262613730343363343834643933643930626564646138633300026966760021000000613393434663766343031343463663961323532383938313866326237336163000 0",

"language": "it",

"model": "iPad",

"os": "8.2",

"reachability": -1,

"retina": true,

"retry_count": 0,

"sdk": "5.1.2",

"session": 1…

Esempio codifica id

“identity":
"930000000a7575696400026
d6163696400290000003266353
6373263623736363931623938 3
962626432303232613533343933
933396132643762393532000 26
966610021000000356562623 93
763326261373034336334383 46
439336439306265646461386 33
300026966760021000000613 39
343466376634303134346366 39
613235323839383138663262 37
3361630000"

“identity”:
uuidmacid)2f5672cb76691
B989bbd2022a5349939a2
→ D7b952**ifa**5ebb97c2ba704
3c484d93d90bedda8c3**ifv**a
3944f7f40144cf9a2528981
8f2b73ac

Analisi statica librerie

- Hopper Disassembler
  - Ricerca all'interno delle librerie di stringhe e procedure relative al tracking
  - Ricerca metodi per offuscamento dati

```
sub_100009c50:
      push        rbp                                     ; XREF=0x100005497,
      mov         rbp, rsp
      sub         rsp, 0x10
      cmp         dword [ds:0x10000F988], 0x0
      je          0x100009CC0
sic Block Input Regs: <nothing> -  Killed Regs: rax rdx rbp rsi rdi
      mov         rax, qword [ds:imp___got____stdoutp]
      mov         rdi, qword [ds:rax]
      call        imp___stubs__fileno
      mov         rsi, 0x80067409
      lea         rdx, qword [ds:0x10000F83E]   ; ""
      mov         edi, eax
      mov         al, 0x0
      call        imp___stubs__ioctl
      mov         rdx, qword [ds:imp___got____stdinp]
      mov         rdi, qword [ds:rdx]
      mov         dword [ss:rbp-0x10+var_8], eax
      call        imp___stubs__fileno
      mov         rsi, 0x80067409
      lea         rdx, qword [ds:0x10000F838]
      mov         edi, eax
      mov         al, 0x0
      call        imp___stubs__ioctl
      mov         dword [ss:rbp-0x10+var_4], eax
sic Block Input Regs: rbp -  Killed Regs: rax rsp rbp
      mov         eax, dword [ss:rbp-0x10+var_12] ; XREF=0x100009c62
      add         rsp, 0x10
      pop         rbp
      ret
```

Librerie analizzate

- DoubleClick
  - Le app ricevono due Javascript offuscati e inviano una richiesta indietro con alcuni dati sul dispositivo, come la piattaforma, il modello, la regione, il nome dell'app
- Flurry
  - Usa HTTP!
  - Serializza i dati con avro
- Chartboost
  - JSON via HTTPS

Http vs Https

- Molte librerie utilizzano connessioni in chiaro
- Alcune inviano dati in JSON mentre altre utilizzano tecniche di serializzazione (avro)
- Serializzare può far risparmiare centinaia di Gb di traffico da gestire sul server
- HTTPS ha un costo quando si parla di un gran numero di connessioni

Future works

- Estendere lo studio a un numero maggiore di App automatizzando il processo
- Aggiornare lo studio alle ultime versioni dei sistemi operativi mobile
- Fare un confronto sui permessi richiesti da una stessa App su iOS e su Android
- Pubblicare report sull’advertisment per iOS in relazione alla privacy

Grazie!

- Antonio Langiu
- E-mail: antonio.langiu@studenti.polito.it
- Bio: http://nexa.polito.it/people/alangiu
- Twitter: https://twitter.com/antonio_langiu
- GitHub: https://github.com/antoniolangiu/