Public Wi-Fi Hotspots

- Like a well in a village
- We gather there, pull up a bucket or two of “Internet”
- Look at the sign from the sponsor
- ... and move on.
What is a “Captive Portal”?

1. connect to Wi-Fi
2. captive portal
3. connect to the Internet
Why Captive Portal

- Omnipresent in Wi-Fi Hotspots
  - Used by you probably right now (in this very hotel)
- Has an elevated position on the network
- Man-in-the-Middle by design
  - Sponsors of a Wi-Fi want us to see their messages (and accept the disclaimer)
  - There is no standard for that
  - Let's inject it into your traffic...
As more sites go HTTPS & more Wi-Fi goes captive portal, I find myself treasuring short names of plain old HTTP sites that get MITMed faster.
Browser History Stealing, again?

- Baron, 2002
  - :visited link color
- Ruderman, 2000
  - :visited can load images
- Jang, 2010
  - Sites are actively trying to steal history
History, so what?

- Culture & Language
  - Amazon.fr, Amazon.jp
- Sexual orientation
  - grindr.com, transblog.de
- Partnership status
  - Okcupid.com, parship.com
- Employer
  - intranet.ibm.com

- Other websites that give interesting insights
  - Medical conditions
  - Political campaigns
  - Religious communities
HTTP Request w/Cookie

URL is entered by user
http://www.virtual.net/

Browser
- initial URL parse
- compare FQDN against cookie list
- compare path against FQDN matching cookies
- select cookie(s) to transmit
- browser does DNS lookup
- browser sends request to IP address/port
  - Request: "/"; Cookie: NAME1="foo";
  NAME2="bar"; ....

Server
HTTP Request w/Cookie

URL is entered by user
http://www.virtual.net/

Browser
- initial URL parse
  - compare FQDN against cookie list
  - compare path against FQDN
  matching cookies
- select cookie(s) to transmit
- browser does DNS lookup
- browser sends request to
  IP address/port
  - Request: "/"; Cookie: NAME1="foo";
    NAME2="bar"; ....

Server
Cookies (or not enough state for HTTP)

- Two kinds
  - Session cookies: usually forgotten when browser closed
  - Persistent cookies: stored on disk with expiry date
- Only depend on the FQDN and Protocol
  - XSS
  - XSRF
- HTTP set cookie also used for HTTPS
  - Insecure set cookies mixed into the cookies over HTTPS
Hotspot User

1. connect to hotspot
2. request random site
3. redirect to captive portal
4. request portal site
5. portal site containing references (e.g. <IMG>) to foreign sites
6. request to external site (incl. possible tracking cookies)
7. portal presents user with "connect" button
8. user clicks on "connect" button in portal
9. user can now load Internet content
10. background tasks (e.g. instant messenger) contact their home servers

WiFi Hotspot with captive

(original suppressed request)

(portal uses collected cookies to enrich data)

Internet
http://cnn.com (+cookies)

302 redirect
login.hotspotsys.com/login

login.hotspotsys.com/login

<html>…. <img
href=”a.com/probe123”>

http://a.com/probe123 (+cookies)
http://b.com/probe123 (+cookies)

request to external site
(incl. possible tracking cookies)
reference to external HTTP-URL

cookie included in req.

Yes

browser acquired cookie at an earlier visit

user visited site before

No

site never visited before

site visited, but cookie expired

site visited, but does not use cookies

site visited, but uses only secure cookies

history status unclear
Sure, crypto will save us!

- HTTPS hides content (and therefore cookies)
  - But is not used by default (e.g., when URL entered in location bar)
- HTTP Strict Transport Security (HSTS)
  - Site announces availability of HTTPS with same content as via HTTP
  - Client caches this
  - Uses HTTPS by default next time for this site
Sure, crypto will save us!

- HTTPS hides traffic (and cookies)
  - But is not used by default (e.g., when URL entered in location bar)
- HTTP Strict Transport Security (HSTS)
  - Site announces availability of HTTPS with same content as via HTTP
    - Client caches this
    - Uses HTTPS by default next time for this site
reference to external HTTP-URL

- type of browser request
  - HTTPS
    - cached or preloaded HSTS info
      - site on HSTS preload list
        - Yes
          - user visited site before
        - No
          - site never visited before
    - browser acquired cookie at an earlier visit
      - Yes
        - site visited, but uses only secure cookies
      - No
        - site visited, but does not use cookies
  - cookie included in req.
    - No
      - site visited, but cookie expired
    - Yes
      - site visited, but does not use cookies

history status unclear
Which users are affected?

- Everyone who uses the standard browser to login into the captive portal (mobile, notebook, ...)
  - Even VPN users
- Android and iOS introduced captive portal detectors
  - Primarily for convenience – starts stripped down browser
  - The online test is very easy to fool, since based on HTTP
  - User will use main browser to login, exposing their history
Captive Portal Detection

• A convenience feature
  • iOS since Version 4
  • Android since 4.2 – used default browser
  • Android since 5 – uses captive portal browser

• Test is easy to circumvent – HTTP!
  • http://clients3.google.com/generate_204
  • http://captive.apple.com/hotspot-detect.html
Which sites affected?

- Uses long-term (persistent) cookie
  - e.g., for session, tracking, or configuration
  - Can be set via
    - HTTP Header
    - Javascript
- Uses HSTS header
Top 1K
82-92%

Top 200K
59-86%
Implementation

- POC implementation
  - Speed up by...
  - Marking probe request with a special string
  - Returning 1x1 pixel or killing connections
- Caching DNS
  - Still one DNS request for each probed site
- Added de-anonymization
  - e.g., via amazon.com
Solutions & Countermeasures?

Transmission

Hotspot Eco-System
Hotspot Ecosystem

There is no standard for hotspot splash screen display

- Choose to present message in-band
- By redirecting/tampering with traffic
- Some do this also for SSL traffic
  - e.g., via DNS _portal.local
Countermeasures

On the Client

- Better captive portal detection
- Private browsing mode for portal
- Same-Site Cookies
  - Circumvention with one more fake indirection step
- Hotspot 2.0
  - Not widely supported
  - Solution for seamless roaming, not for showing banners and ads to the customer

On the Server

- HTTPS-only cookies
- Google is changing an increasing number of Client-APIs to require HTTPS connections
Conclusion

- Captive Portals (& MITM) can learn about the
  - current session
  - past browsing sessions
  - Even for VPN users
- Side channels
  - Cookies
  - HSTS
- Alexa Top 1K
  - 82-92%
- Alexa Top 200K
  - 59-86%
Browser History Stealing with Captive Wi-Fi Portals

Adrian Dabrowski, Georg Merzdovnik, Nikolaus Kommenda, Edgar Weippl
adabrowski@sba-research.org

Twitter: @atrox_at

2016-05-26