

Covert Web-to-App Tracking via Localhost on Android

Breaking Cross-Context Isolation through an Unmediated Loopback Channel

Aniketh Girish, Gunes Acar, Narseo Vallina-Rodriguez, Nipuna Weerasekara, Tim Vlummens

PEARG, IETF
July 25, 2025
Madrid, Spain



uc3m

Universidad
Carlos III
de Madrid

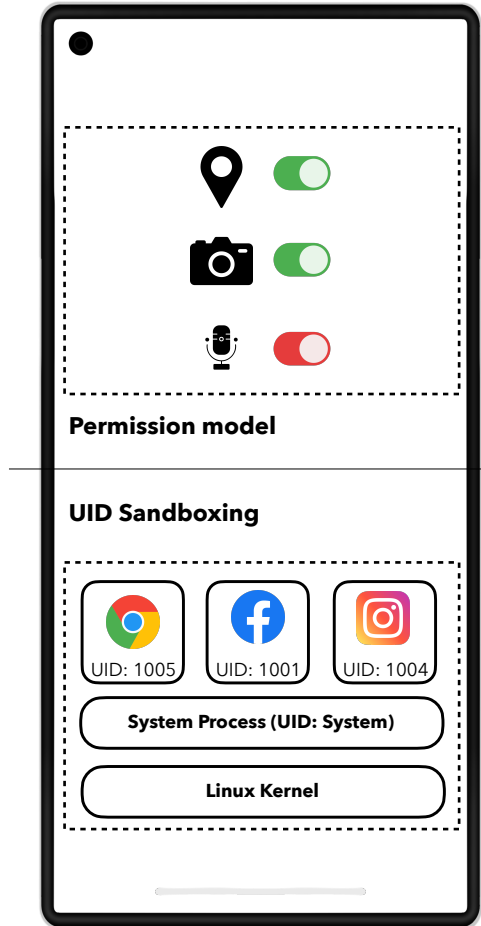
KU LEUVEN



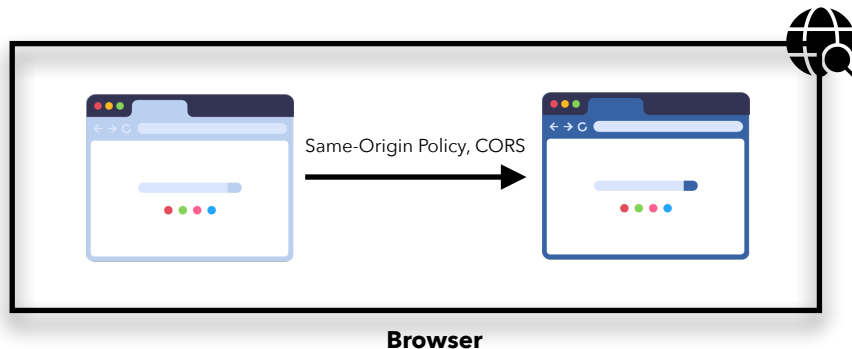
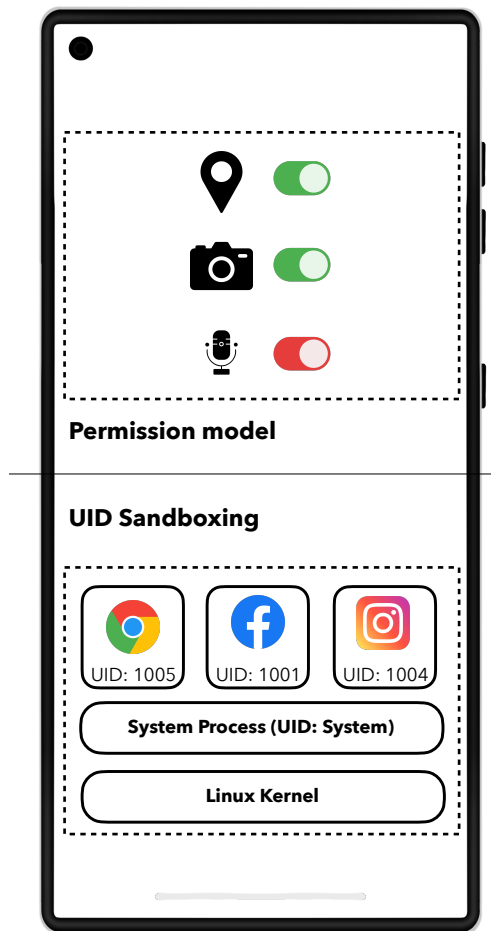
Radboud Universiteit



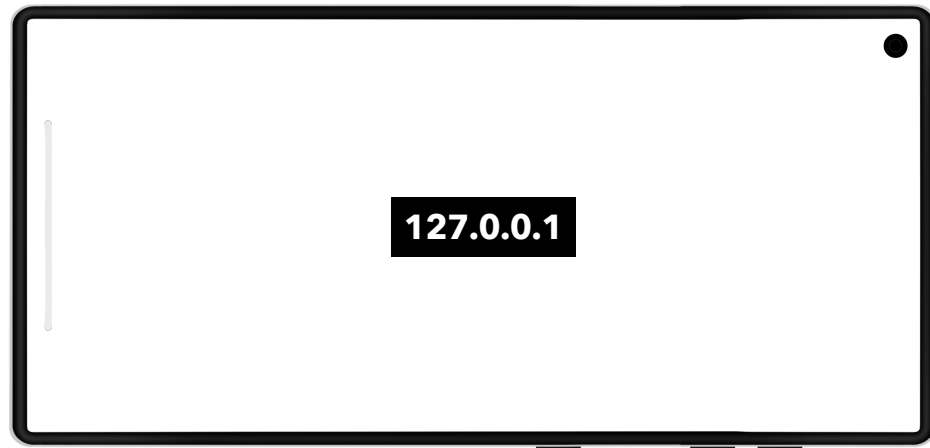
Modern Platforms Enforce Contextual Isolation



Modern Platforms Enforce Contextual Isolation



Localhost: A Quiet Cross-Context Channel



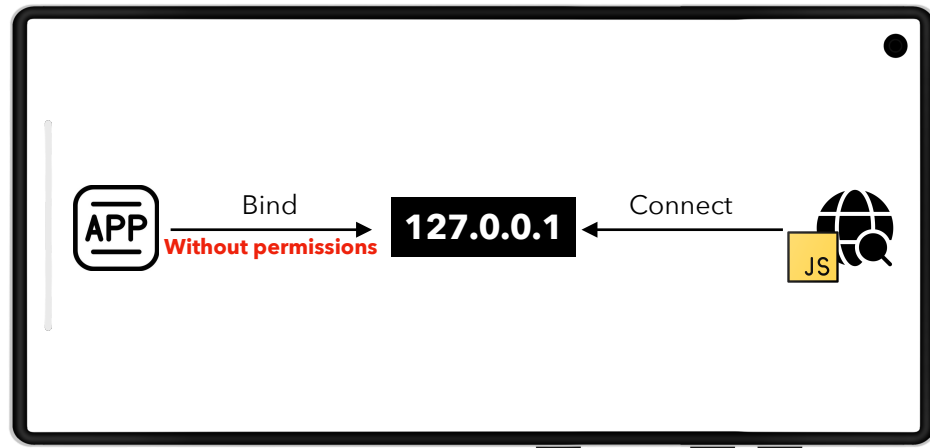
Localhost: A Quiet Cross-Context Channel

- Apps can bind to 127.0.0.1 without permissions.



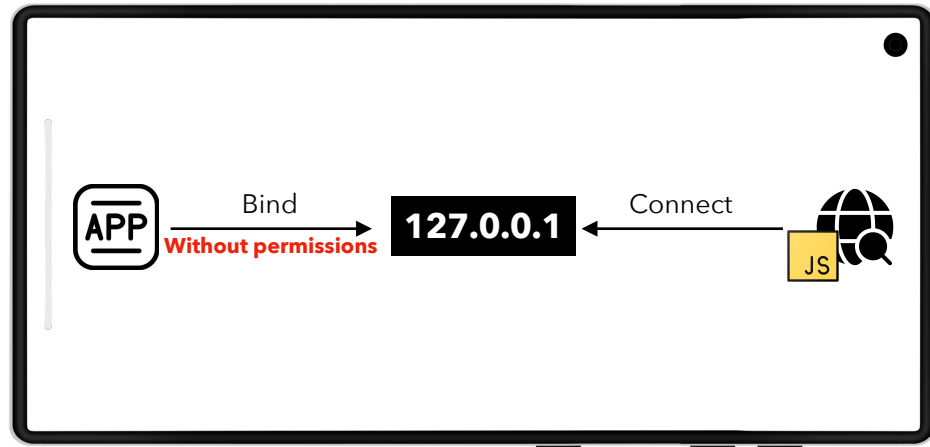
Localhost: A Quiet Cross-Context Channel

- Apps can bind to 127.0.0.1 without permissions.
- Browsers allow web JS to initiate connections to localhost.



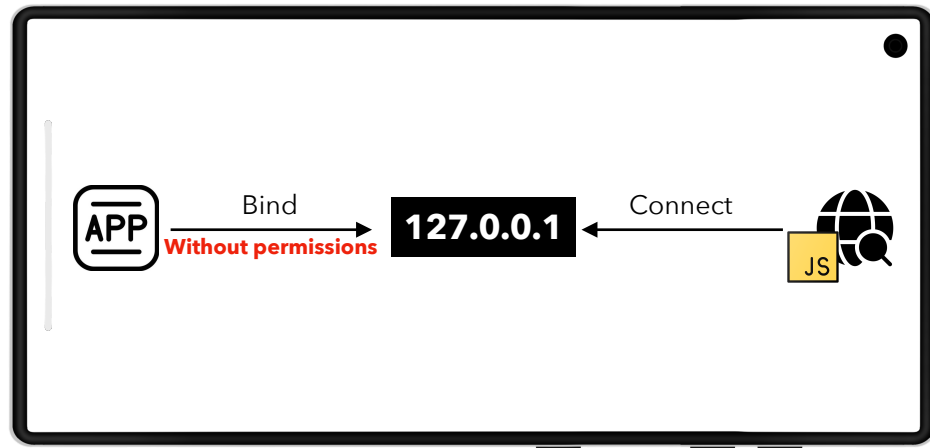
Localhost: A Quiet Cross-Context Channel

- Apps can bind to 127.0.0.1 without permissions.
- Browsers allow web JS to initiate connections to localhost.
- **Legitimate purposes:** used for testing, debugging, P2P fallback.



Localhost: A Quiet Cross-Context Channel

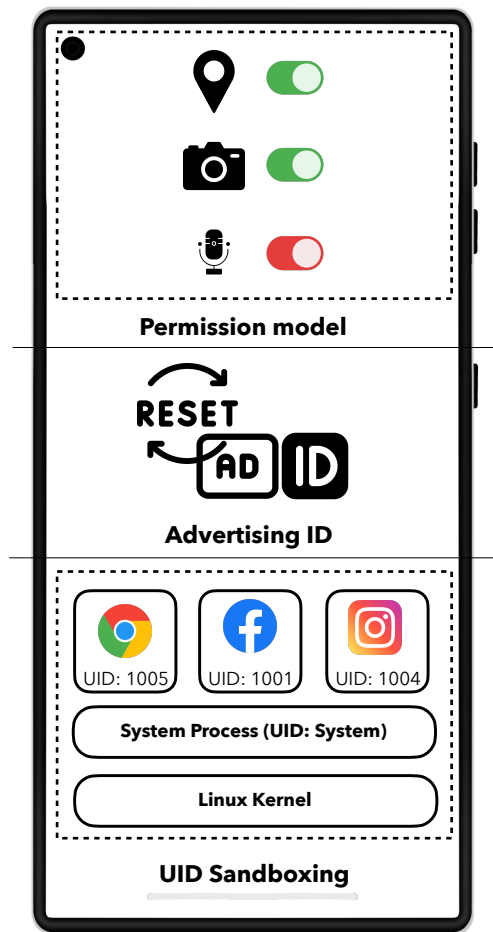
- Apps can bind to 127.0.0.1 without permissions.
- Browsers allow web JS to initiate connections to localhost.
- **Legitimate purposes:** used for testing, debugging, P2P fallback.



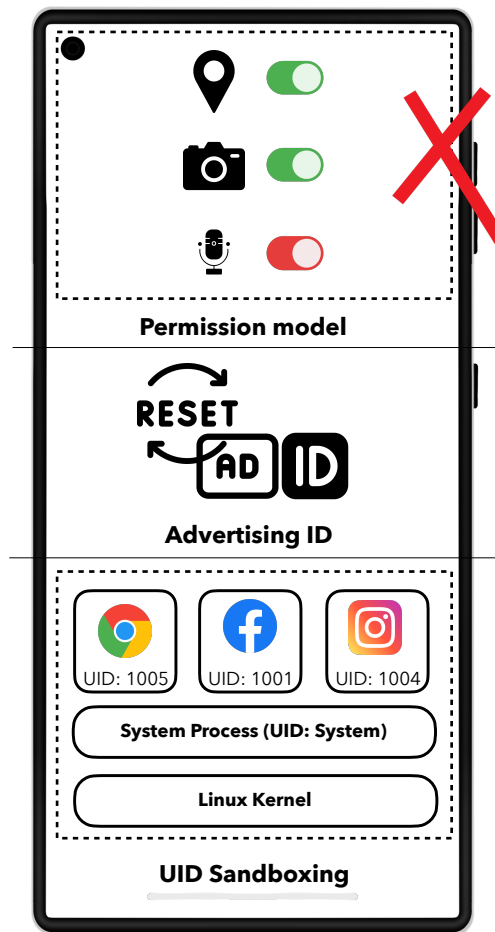
Any app can listen. Any website can send. And the user sees none of it.



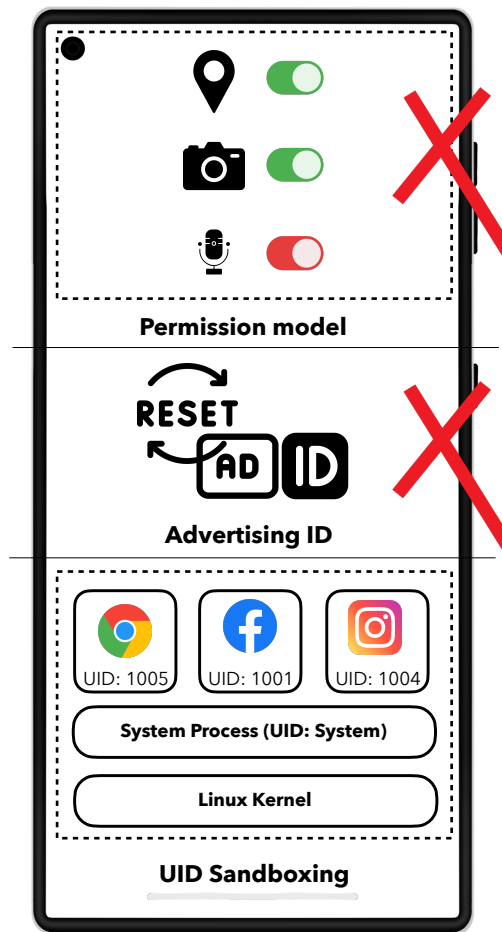
Existing Defenses are Ineffective



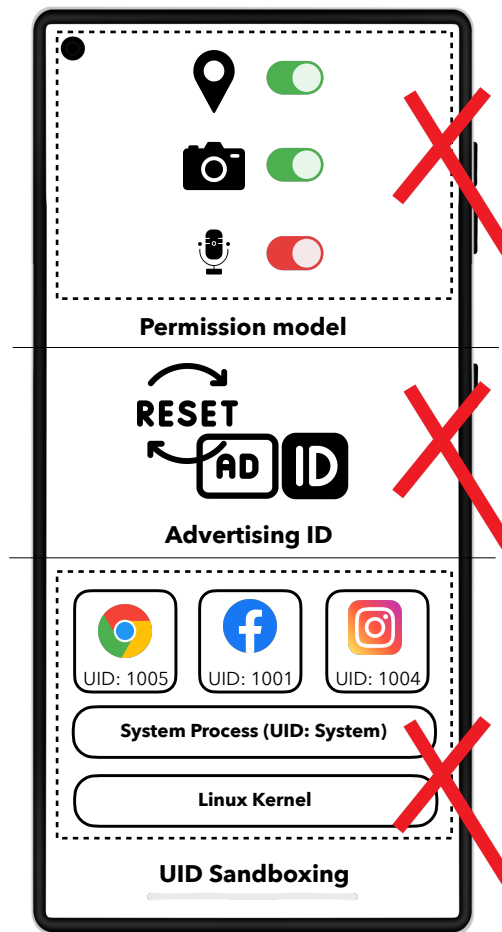
Existing Defenses are Ineffective



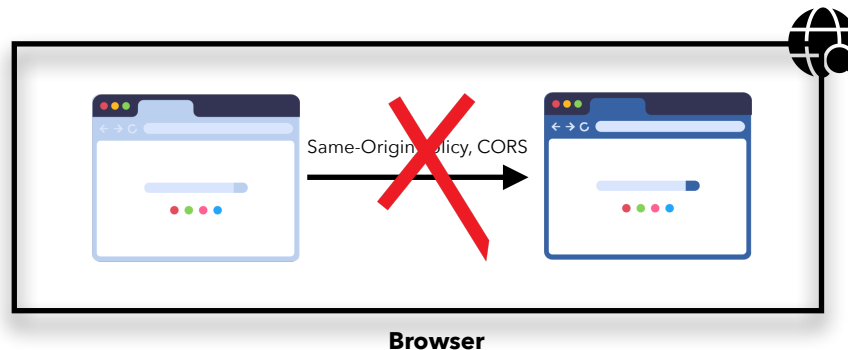
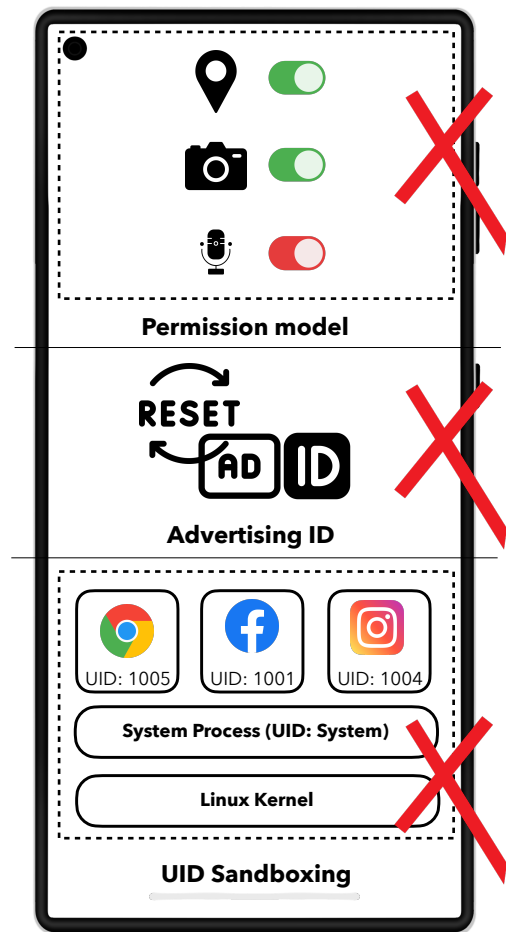
Existing Defenses are Ineffective



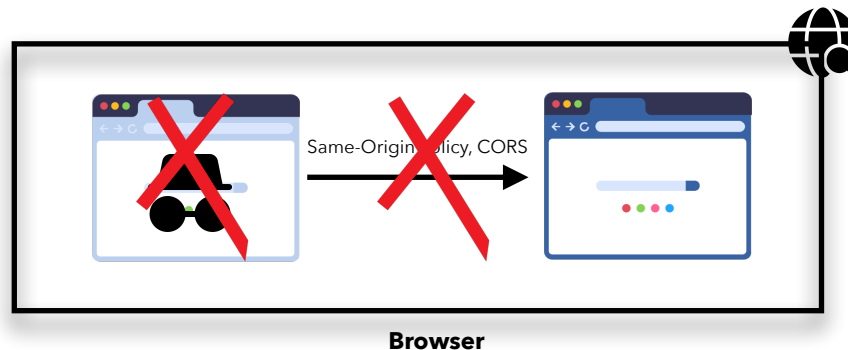
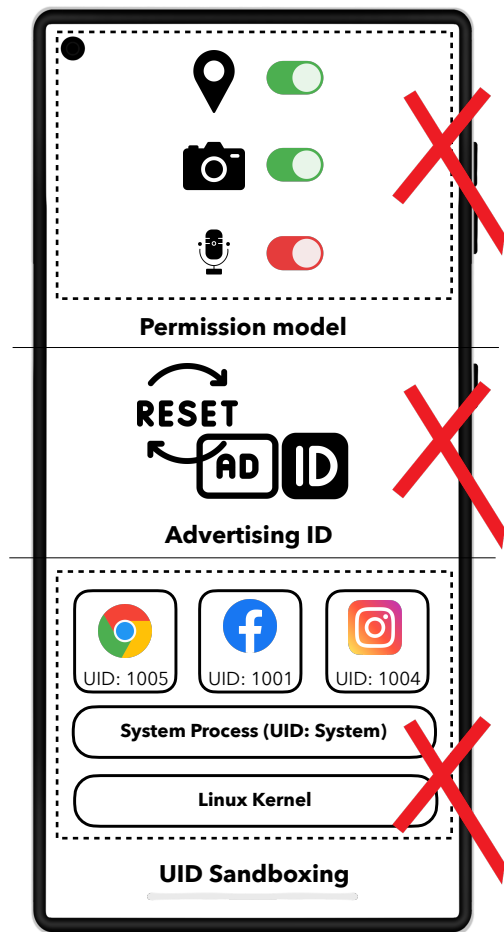
Existing Defenses are Ineffective



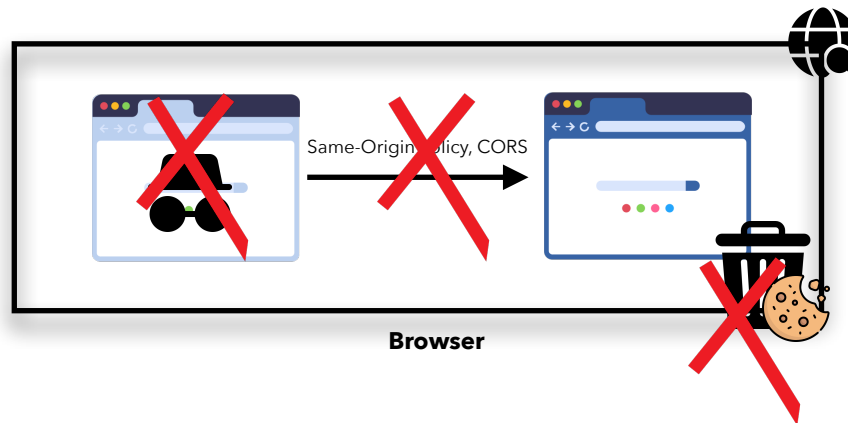
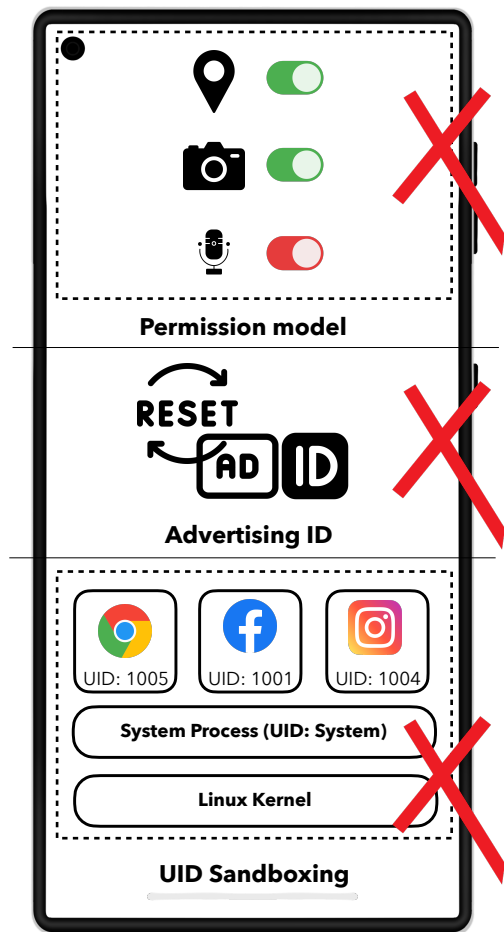
Existing Defenses are Ineffective



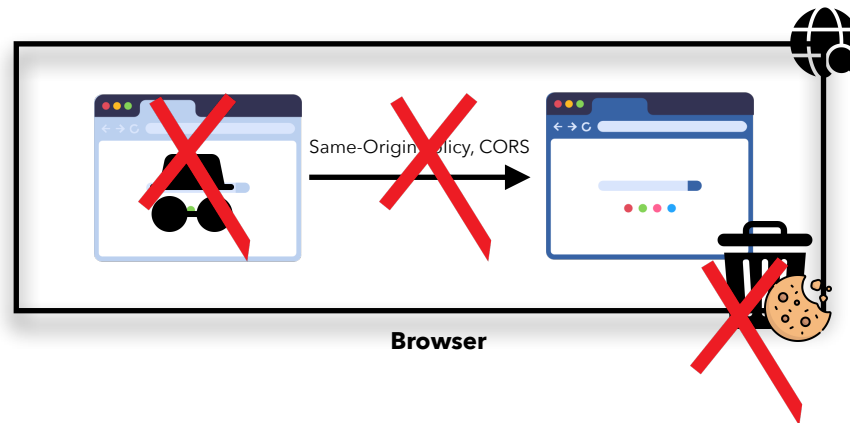
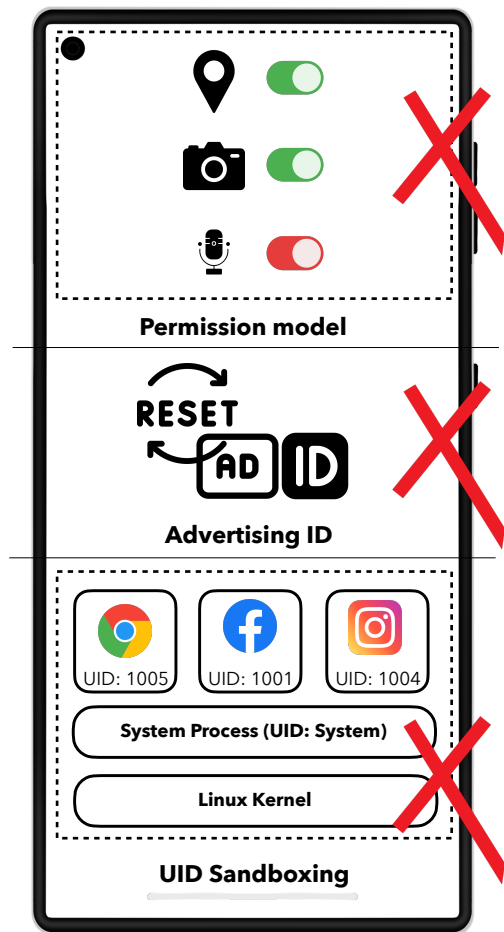
Existing Defenses are Ineffective



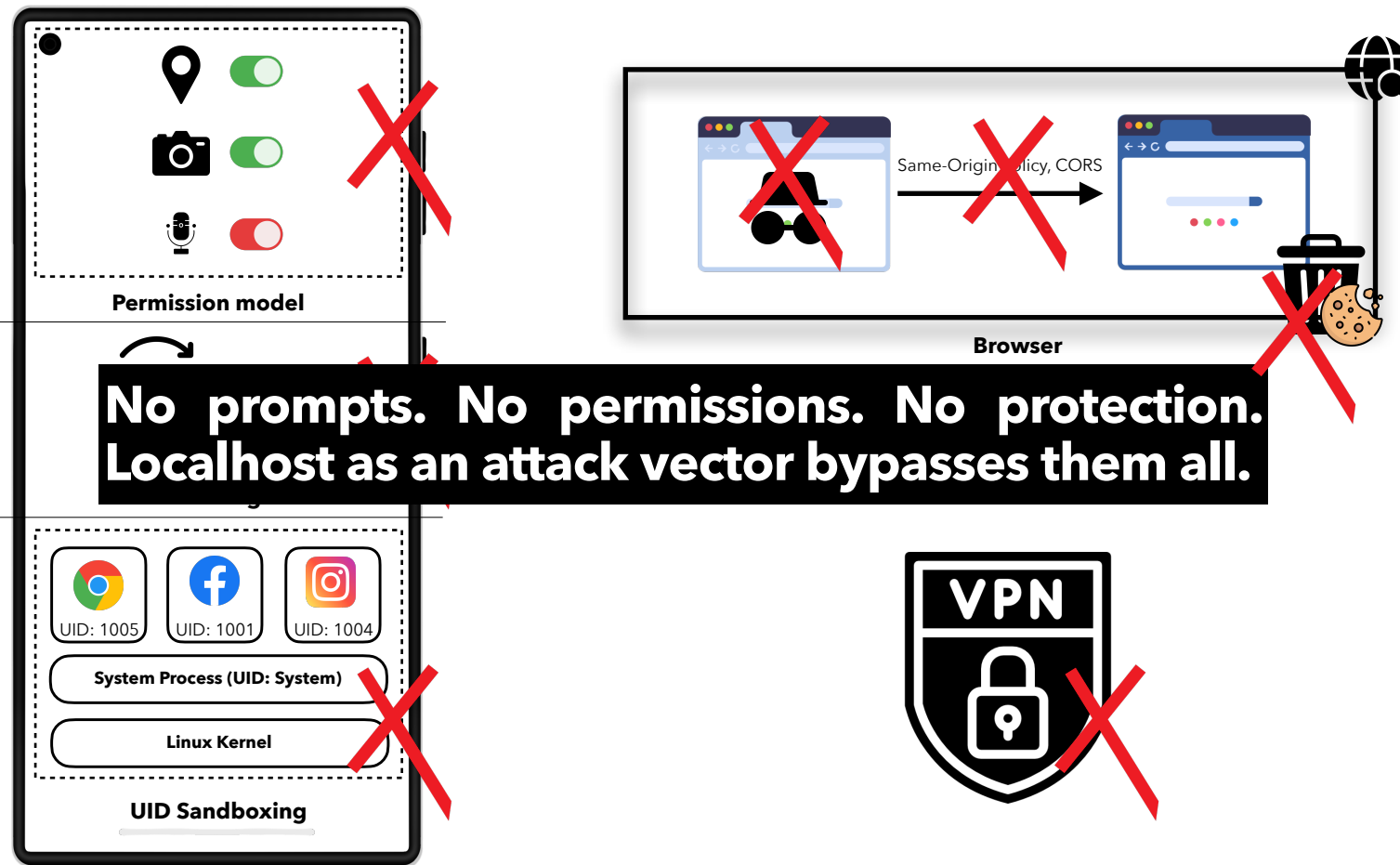
Existing Defenses are Ineffective



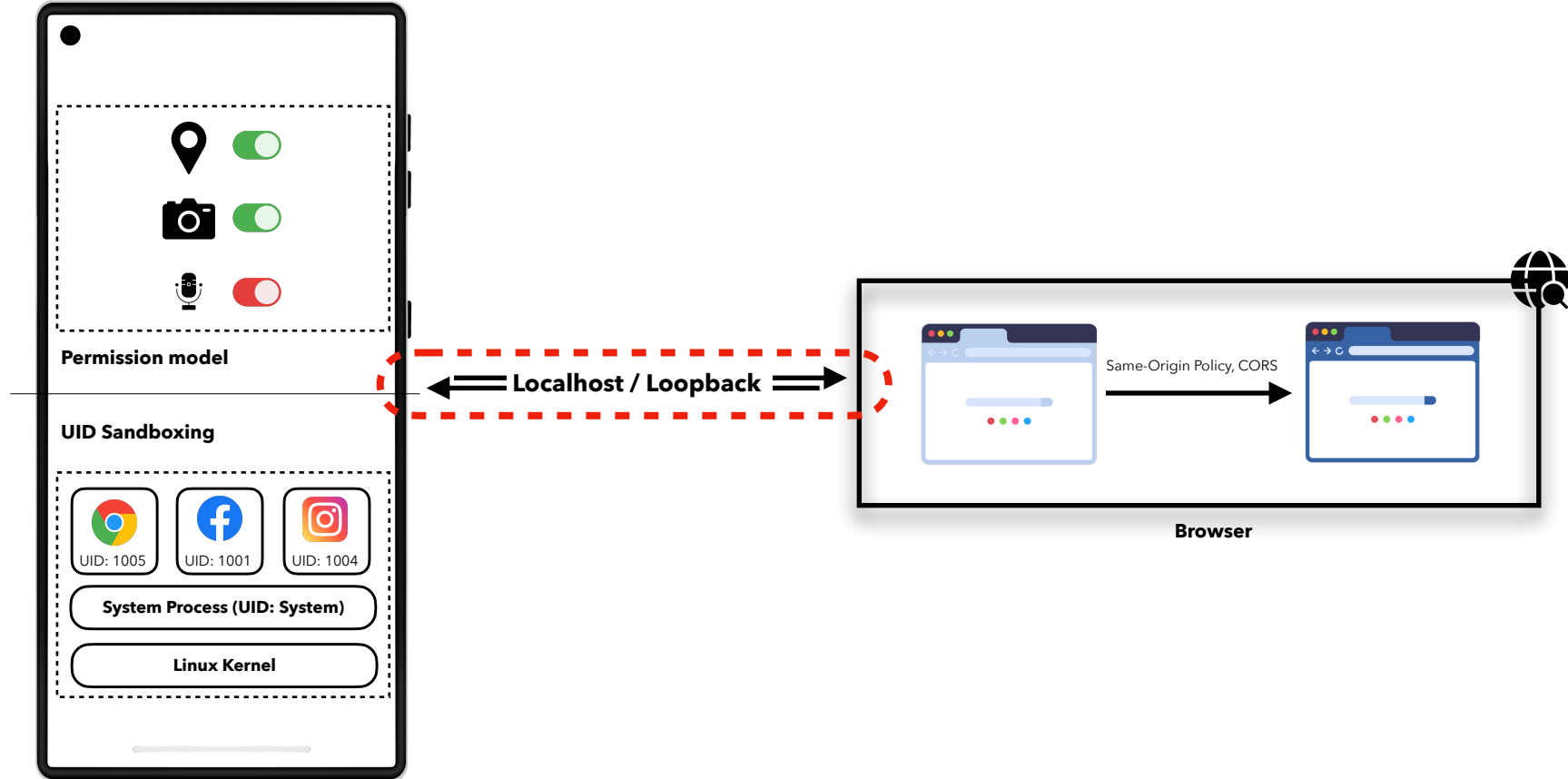
Existing Defenses are Ineffective



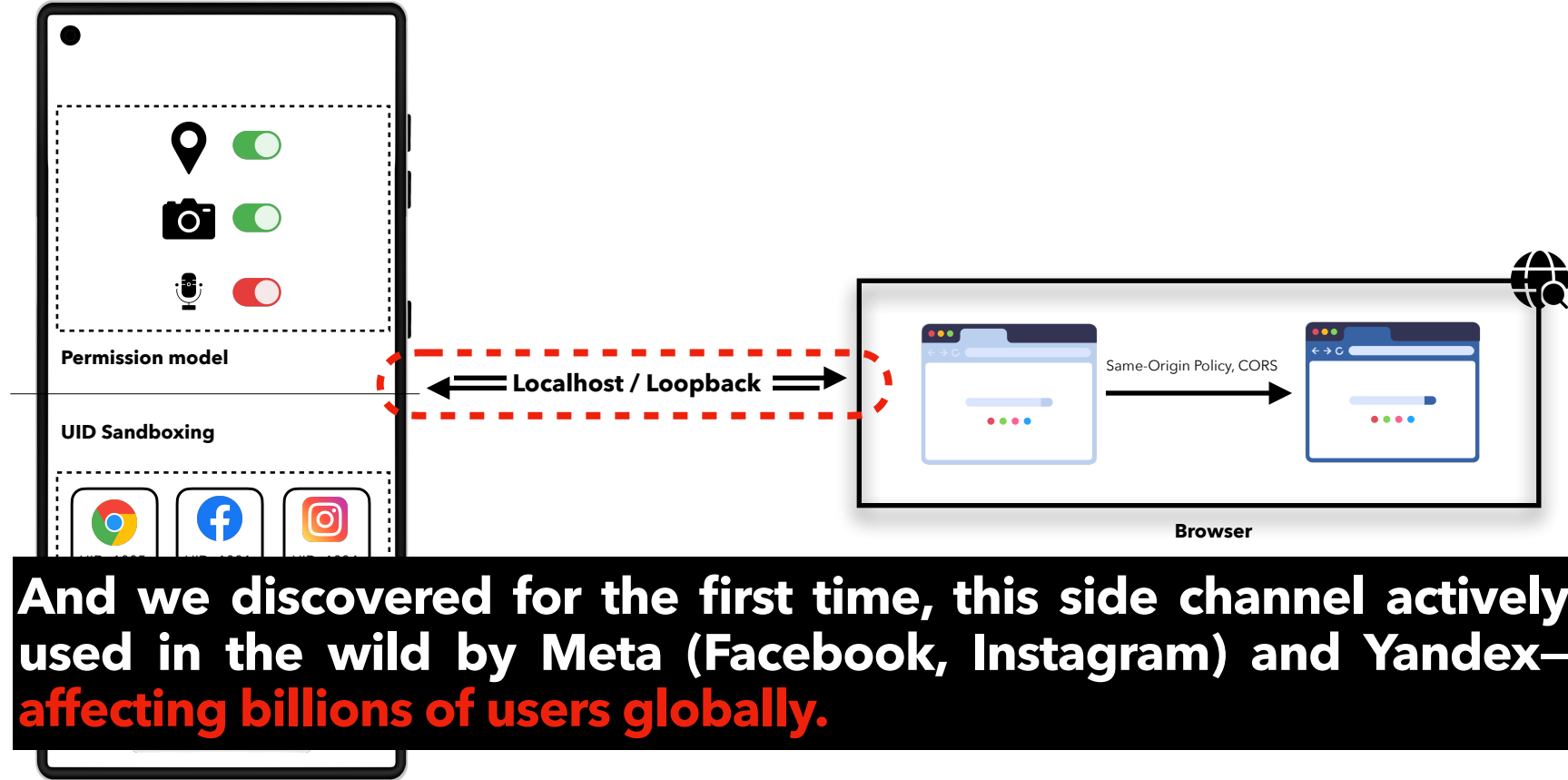
Existing Defenses are Ineffective



Do Sites Use Localhost to Communicate with Mobile Apps?

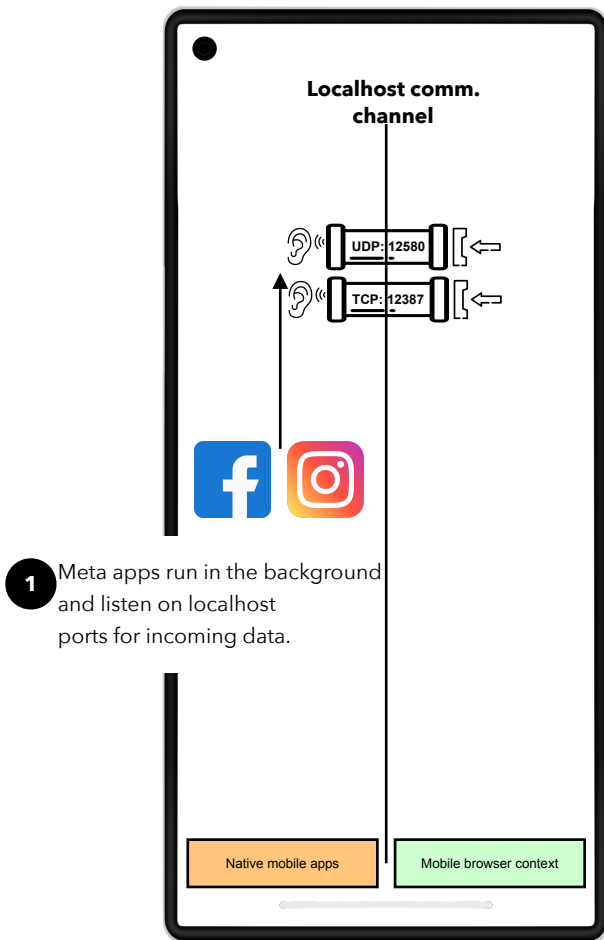


Do Sites Use Localhost to Communicate with Mobile Apps?

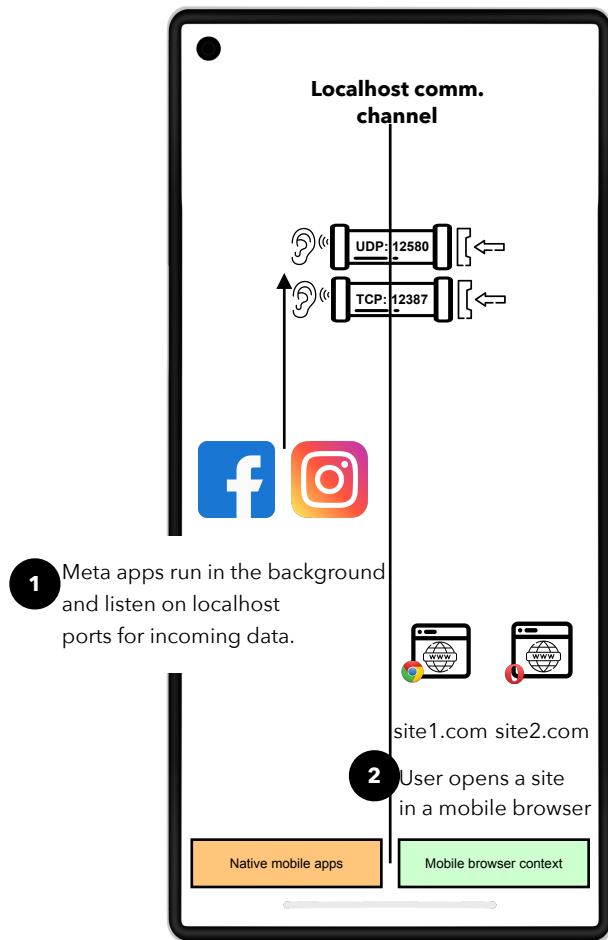


Meta's Side Channel

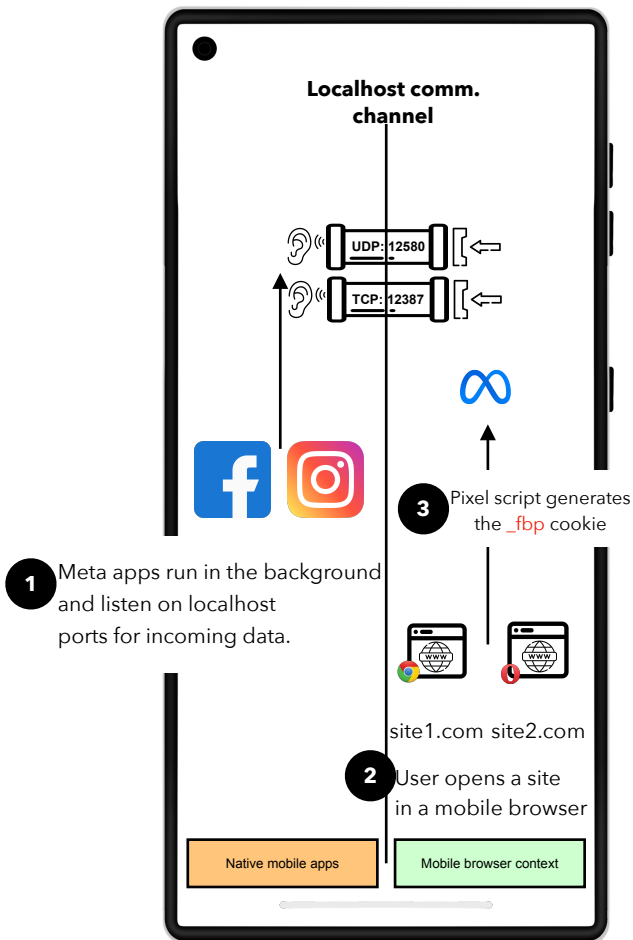
Technical Details



Technical Details

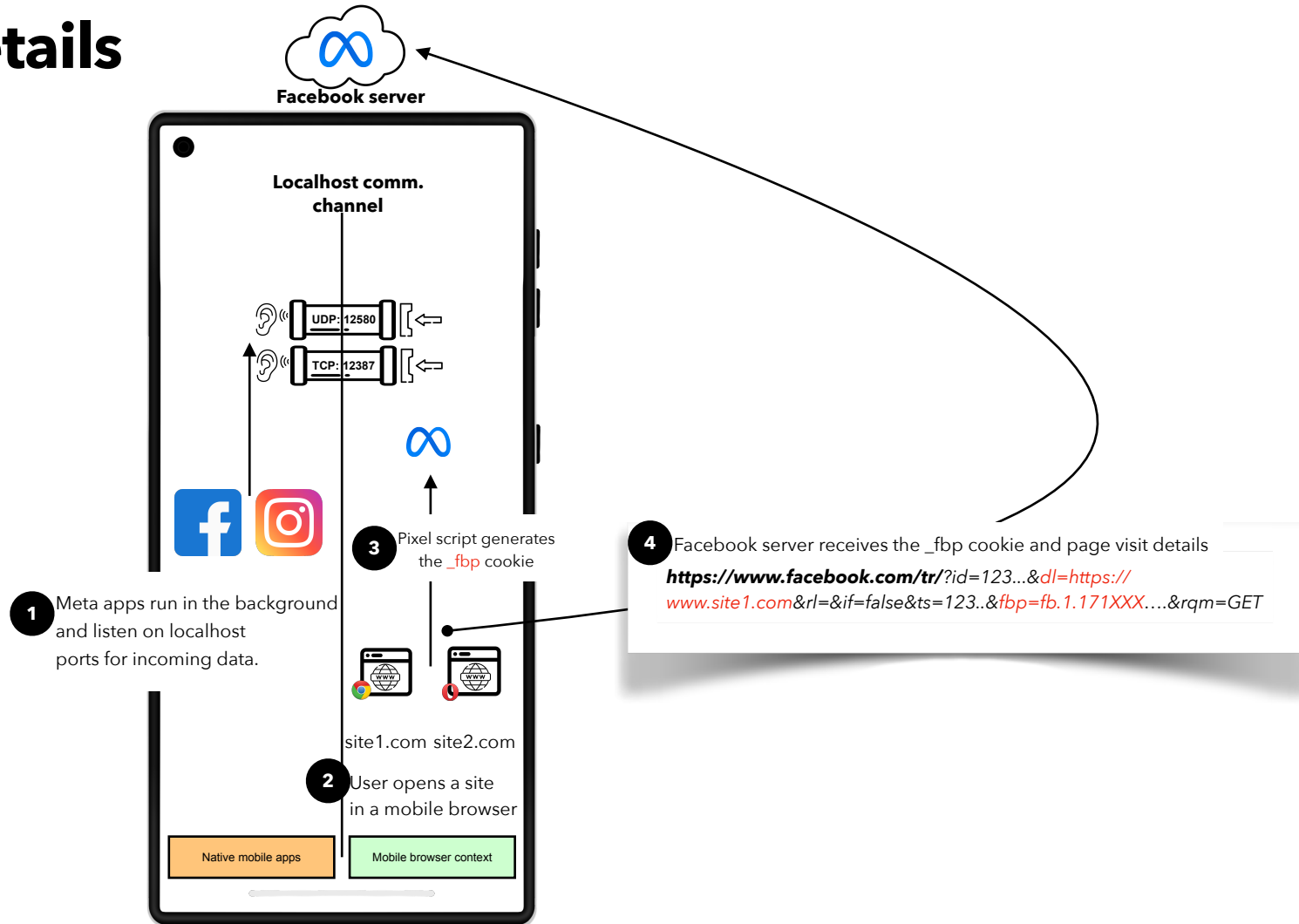


Technical Details

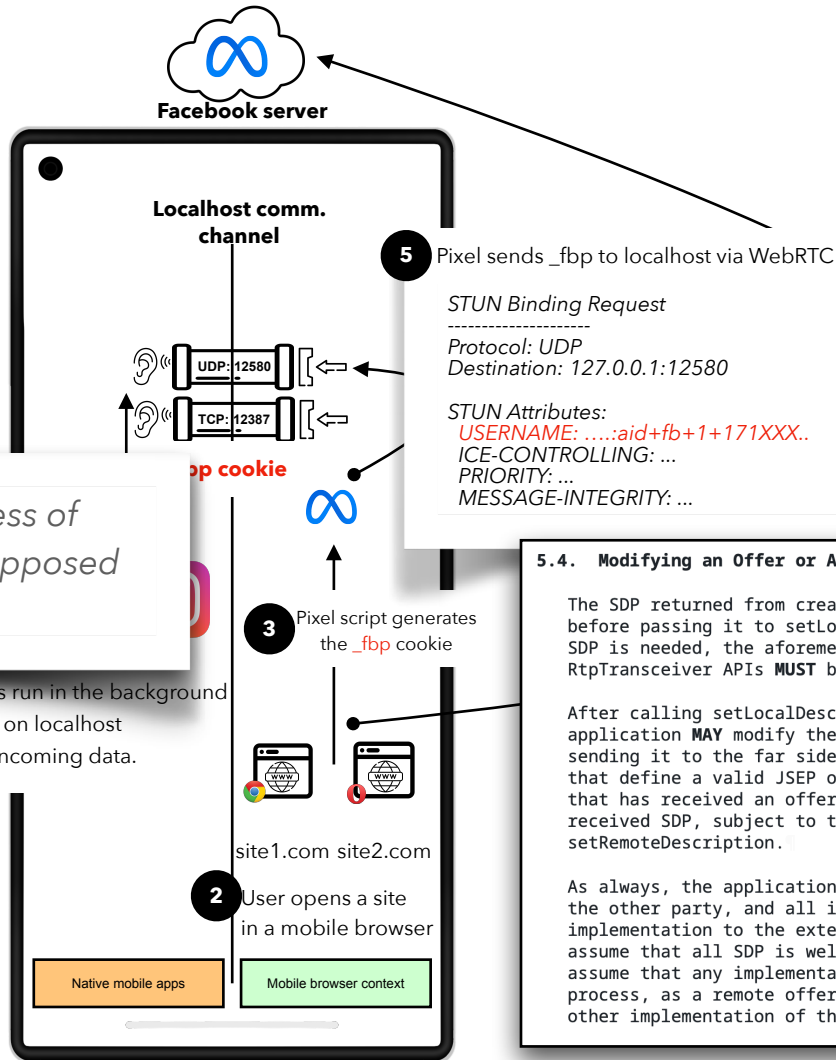


"When the Meta Pixel is installed on a website, and the Pixel uses **first-party cookies**, the Pixel automatically saves a **unique identifier** to an `_fbp` cookie **for the website domain** if one does not already exist."

Technical Details



Technical Details



"SDP munging refers to the process of changing the SDP manually as opposed to letting the WebRTC API's do it."

STUN Binding Request

Protocol: UDP
Destination: 127.0.0.1:12580

STUN Attributes:
USERNAME:aid+fb+1+171XXX..
ICE-CONTROLLING: ...
PRIORITY: ...
MESSAGE-INTEGRITY: ...

RFC 8829

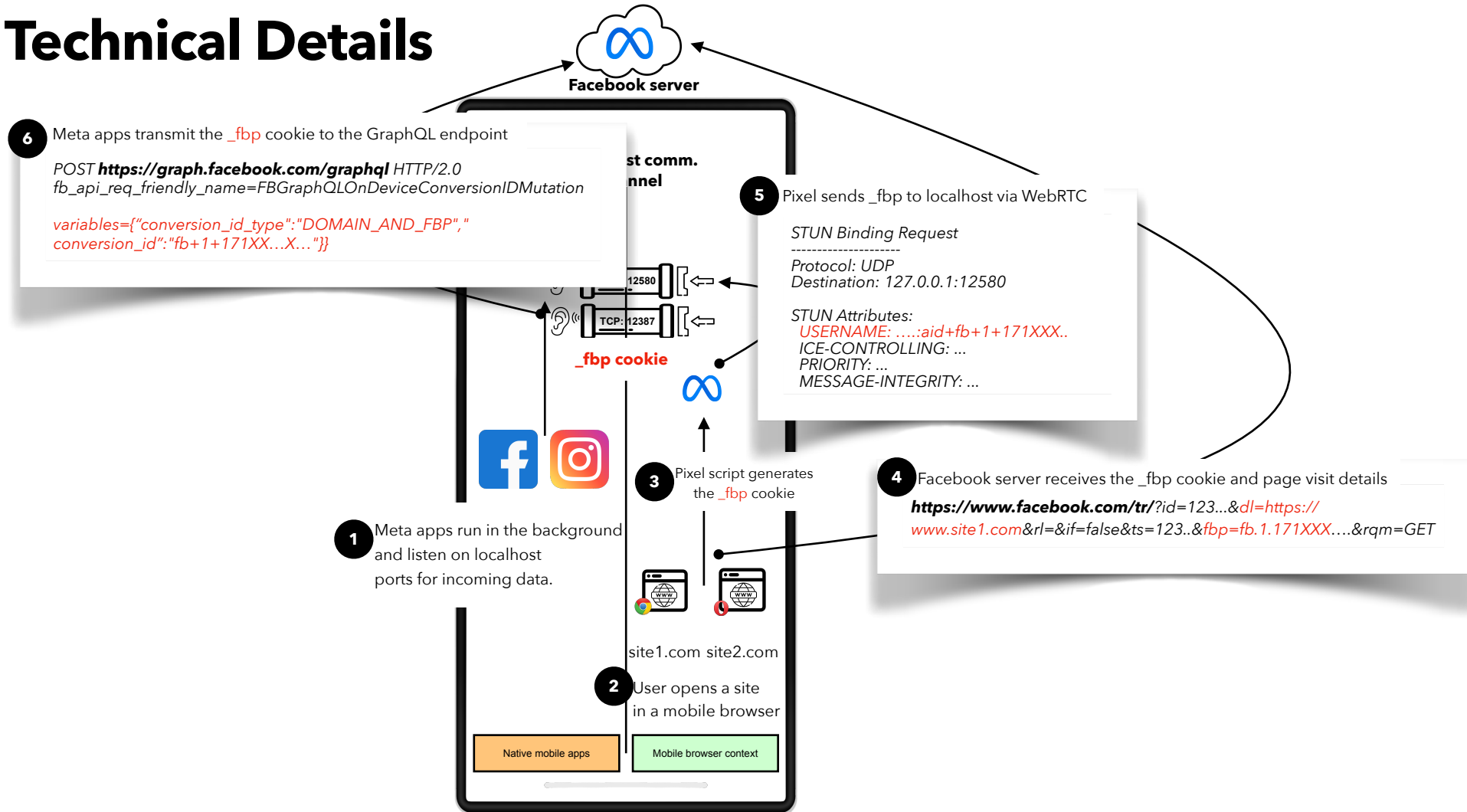
5.4. Modifying an Offer or Answer

The SDP returned from `createOffer` or `createAnswer` **MUST NOT** be changed before passing it to `setLocalDescription`. If precise control over the SDP is needed, the aforementioned `createOffer/createAnswer` options or `RtpTransceiver` APIs **MUST** be used.

After calling `setLocalDescription` with an offer or answer, the application **MAY** modify the SDP to reduce its capabilities before sending it to the far side, as long as it follows the rules above that define a valid JSEP offer or answer. Likewise, an application that has received an offer or answer from a peer **MAY** modify the received SDP, subject to the same constraints, before calling `setRemoteDescription`.

As always, the application is solely responsible for what it sends to the other party, and all incoming SDP will be processed by the JSEP implementation to the extent of its capabilities. It is an error to assume that all SDP is well formed; however, one should be able to assume that any implementation of this specification will be able to process, as a remote offer or answer, unmodified SDP coming from any other implementation of this specification.

Technical Details



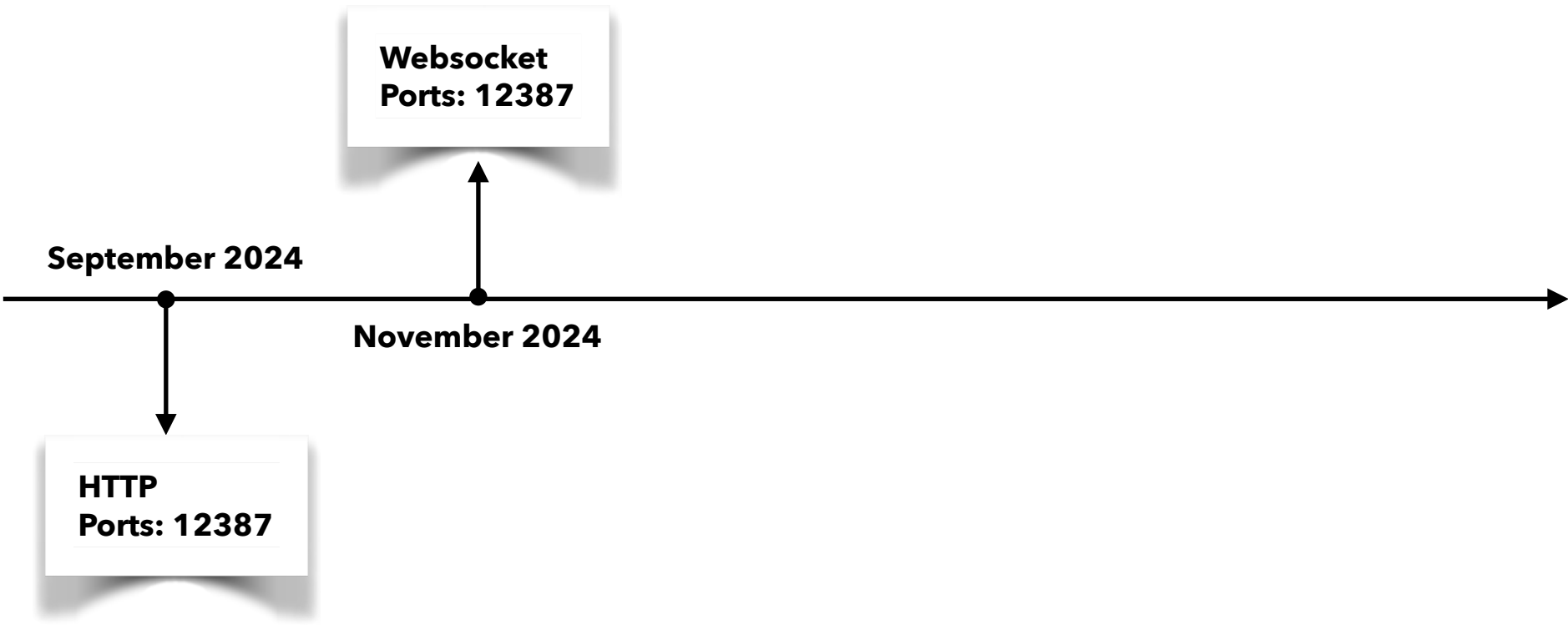
Meta's Side Channel: Behaviour Across Time

September 2024

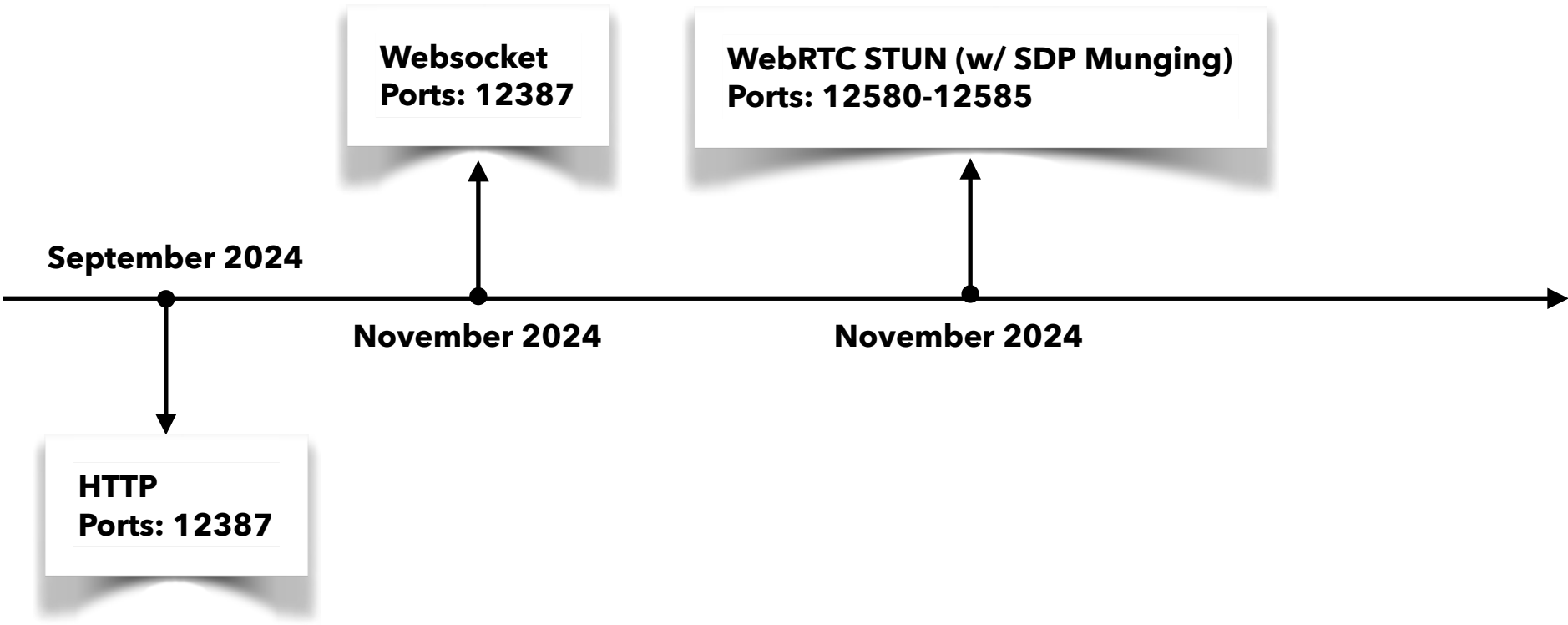


HTTP
Ports: 12387

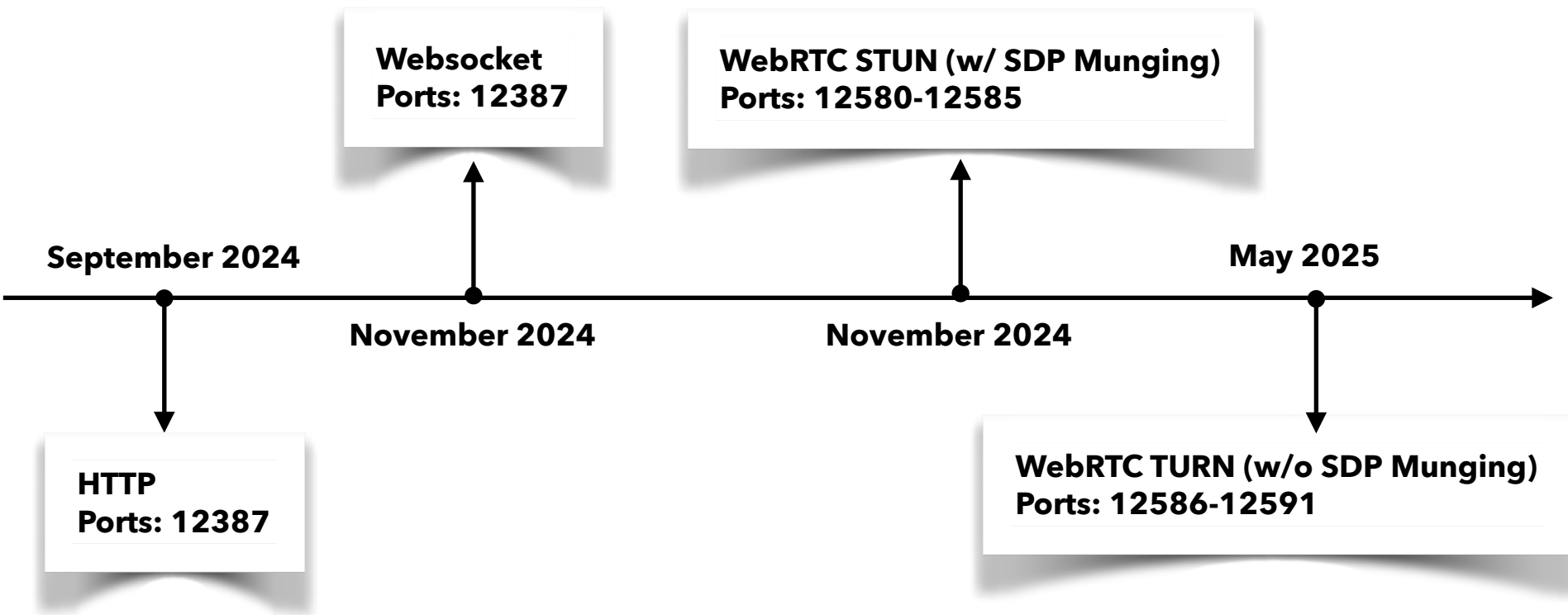
Meta's Side Channel: Behaviour Across Time



Meta's Side Channel: Behaviour Across Time



Meta's Side Channel: Behaviour Across Time



Were Website Owners Even Aware?

Community > Question

Why does my Pixel Javascript access http://localhost when in an embedded web view on Android

7

We have a meta tracking pixel on a client website, embedded and tracking correctly using google tag manager. We also show some of the content from the site embedded in a web view in their Android app. We've noticed that the Webview is throwing an error as the Pixel

and 9 disallows this as it could post ? 2. Why on ?
g/?

localhost.

ut 10 months

ction to
IMPLEMENTED



Glenn

Status

● Unresolved

Posted

10 months ago

Last Answered

8 months ago

Type

Unclassified

Meta

Docs

Tools

Support

Search

Log In

Support

Bugs

Community

FAQ

Platform Status

Community > Question

Facebook SDK config file making call to localhost

13

Since 5th September, our internal JS error tracking has been flagging failed fetch requests to localhost:12387. No changes have been made on our side, and the existing Facebook tracking pixel we use loads via Google Tag Manager.

Specifically, on closer inspection, the config file found at - <https://connect.facebook.net/signals/config/> - has this chunk of code within it that tries to make a call to localhost.

```
(function(a,b,c,d){var e={exports:{}},e.exports=function(){var f=a.fbq,f.execStart=a.performance&&a.performance.now&&a.performance.now(),if(!function(){var b=a.postMessage(function(){},{});if(!b){b={action:"FB_LOG",logType:"Facebook Pixel Error",logMessage:"Pixel code is not installed correctly on this page"},"error" in console&&console.error("Facebook Pixel Error: Pixel code is not installed correctly on this page");return!1}}return!0){})return f._fbeventsModules||(f._fbeventsModules={},f._fbeventsResolvedModules={},f.getFbEventsModules=function(a){f._fbeventsResolvedModules[a]||(f._fbeventsResolvedModules[a]=f._fbeventsModules[a]);return f._fbeventsResolvedModules[a]},f.fbIsModuleLoaded=function(a){return!!f._fbeventsModules[a]},f.fbIsModuleLoaded=function(a){f._fbeventsModules[b]=a});f.ensureModuleRegistered=function(b,a){f.fbIsModuleLoaded(b)||f._fbeventsModules[b]=a}});
```

```
f.ensureModuleRegistered("SignalsFBEvents.plugins.actionid",function(){return function(a,b,c,d){var e={exports:{}},e.exports=function(){var a=f.getFbEventsModules("SignalsFBEventsPlugin"),b=f.getFbEventsModules("SignalsFBEventsEvent"),c=b.setFBPId,f.getFbEventsModules("signalsFBEventsGetIsAndroid"),g=f.getFbEventsModules("signalsFBEventsGetIsAndroidIAW"),b=f.getFbEventsModules("SignalsFBEventsLogging");var
```

Status

● Unresolved

Posted

9 months ago

Last Answered

8 months ago

Type

Unclassified

Were Website Owners Even Aware?

Community > Question

Why does my Pixel Javascript access http://localhost when in an embedded web view on Android

7

We have a meta tracking pixel on a client website, embedded and tracking correctly using google tag manager. We also show some of the content from the site embedded in a web

Status

Unresolved

Meta



Olly

This seems to be fixed on my SDK as of 1st October between 19:00 and 20:00 BST. No acknowledgement has come from Meta at all on this though. My support request with them got a generic response and then ignored thereafter.

October 21 at 3:16 AM

```
(function(a,b,c,d){var e={exports:{};e.exports=function(){var f=a.fbq;f.execStart=a.performance&&a.performance.now&&a.performance.now();if(!function(){var b=a.postMessage||function(){};if(!b){b({action:"FB_LOG",logType:"Facebook Pixel Error",logMessage:"Pixel code is not installed correctly on this page"},"");"error" in console&&console.error("Facebook Pixel Error: Pixel code is not installed correctly on this page");return!1}return!0}())return f.__fbEventsModules||(f.__fbEventsModules={};f.__fbEventsResolvedModules={},f.getFbEventsModules=function(a){f.__fbEventsResolvedModules[a]||(f.__fbEventsResolvedModules[a]=f.__fbEventsResolvedModules[a]);return f.__fbEventsResolvedModules[a]};f.fbIsModuleLoaded=function(a){return!!f.__fbEventsModules[a]};f.ensureModuleRegistered=function(b,a){f.fbIsModuleLoaded(b)||(f.__fbEventsModules[b]=a)}});
```

```
f.ensureModuleRegistered("SignalsFBEvents.plugins.actionid",function(){return function(a,b,c,d){var e={exports:{};e.exports=function(){"use strict";var a=f.getFbEventsModules("SignalsFBEventsPlugin"),b=f.getFbEventsModules("SignalsFBEventsEvent s"),c=b.setFBP,d=f.getFbEventsModules("signalsFBEventsGetsAndroid"),g=f.getFbEventsModules("s ignalsFBEventsGetsAndroidIAW"),b=f.getFbEventsModules("SignalsFBEventsLogging");var
```

Type Unclassified

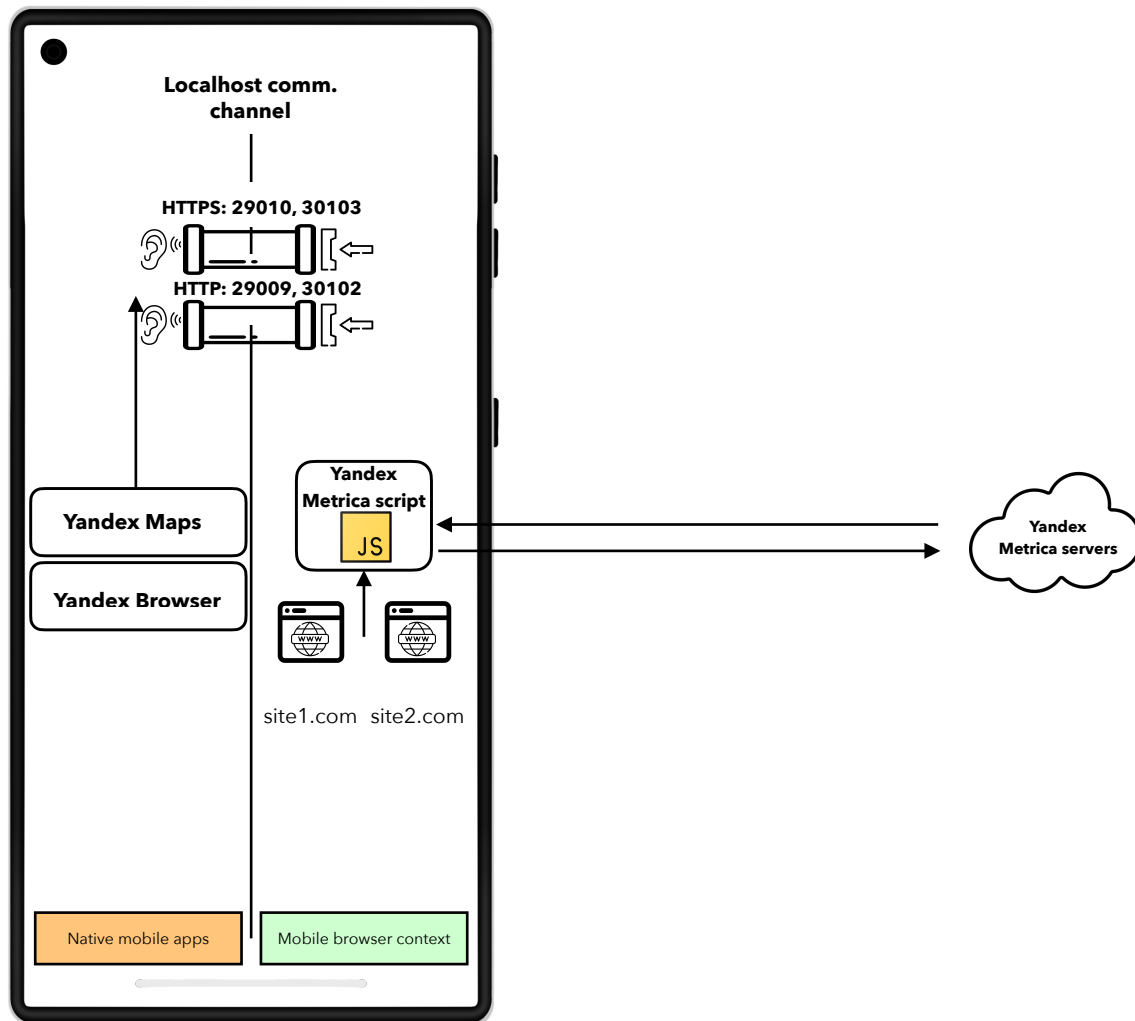
ction to
IMPLEMENTED



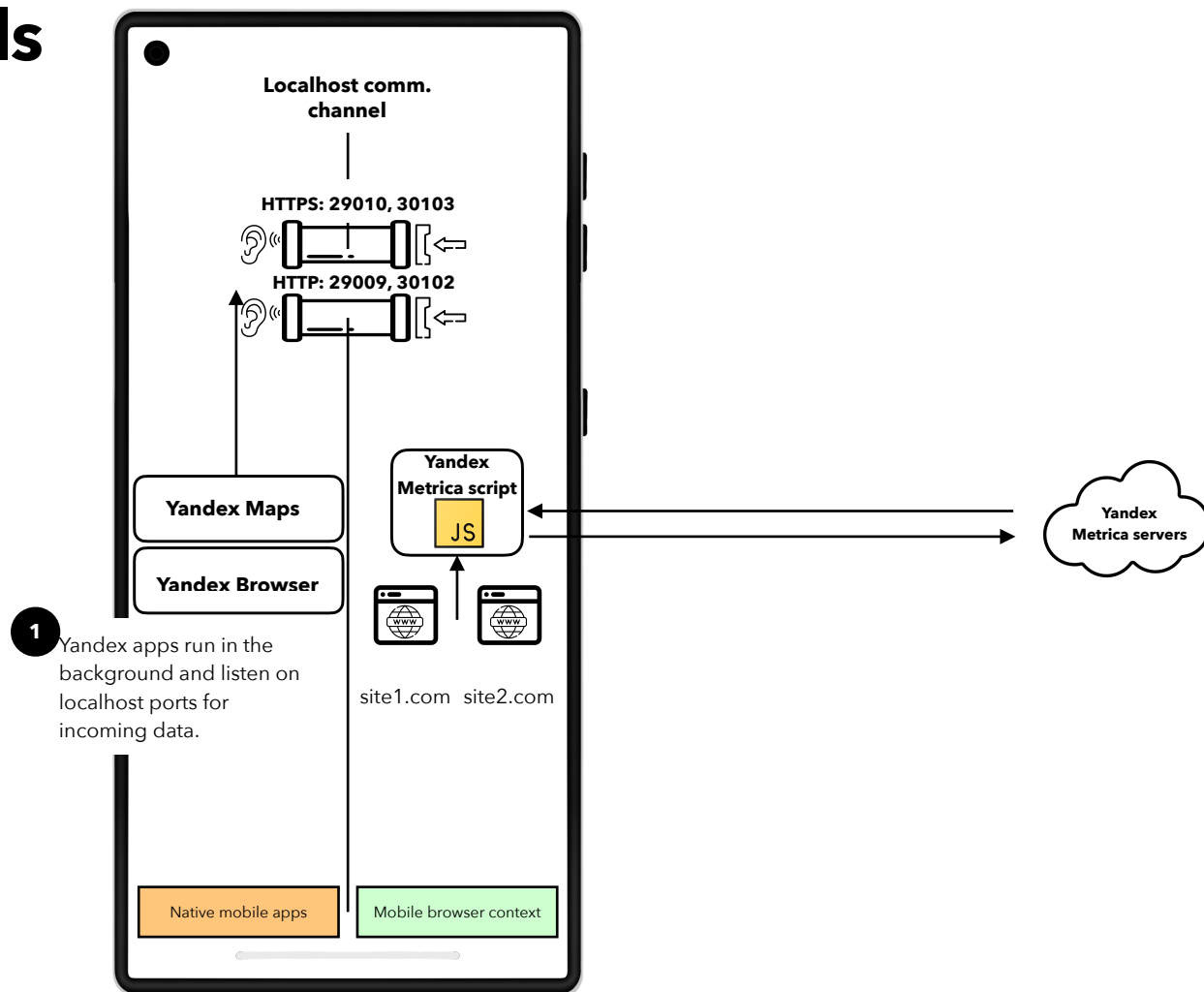
Glenn

Yandex's Side Channel

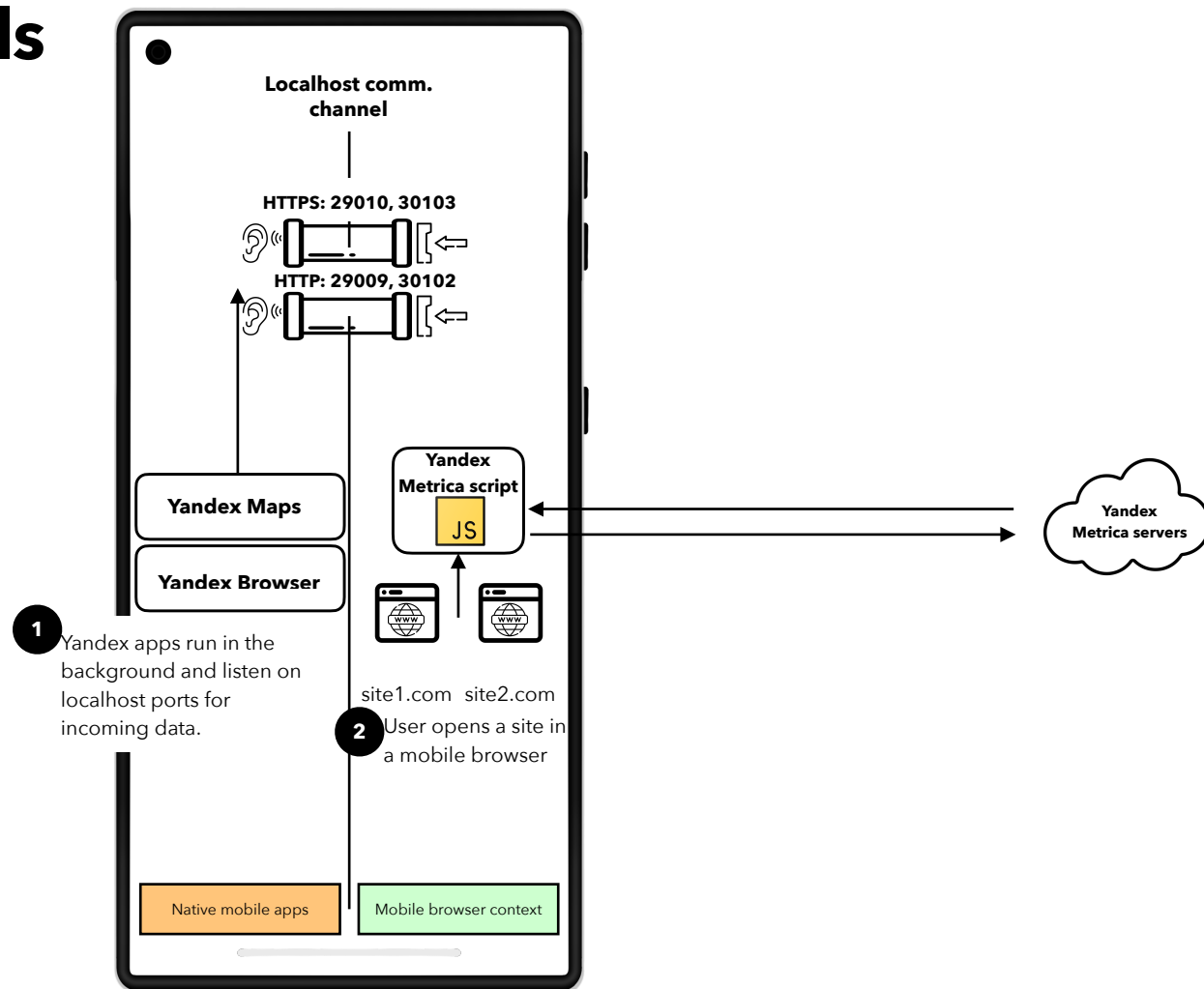
Technical Details



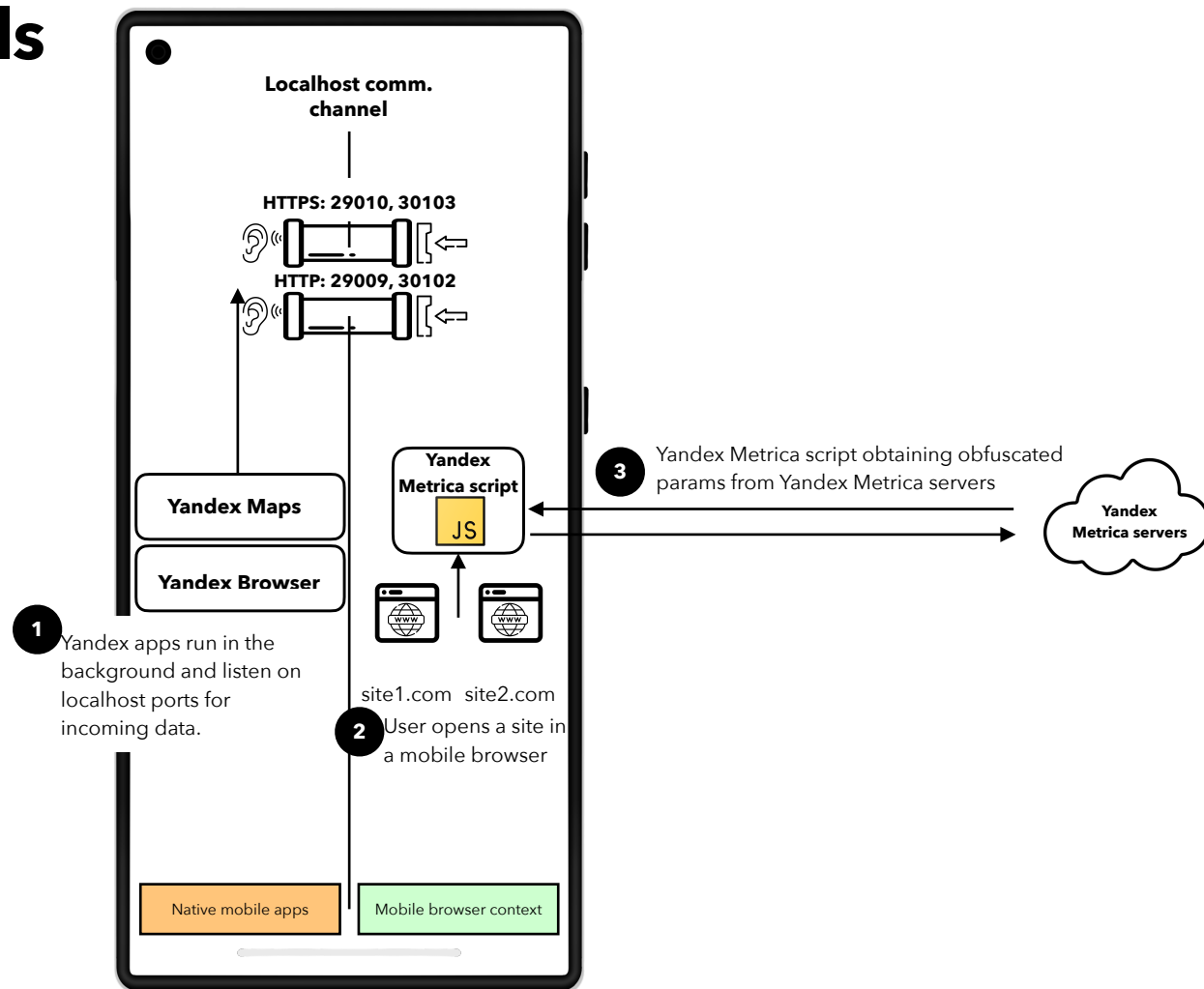
Technical Details



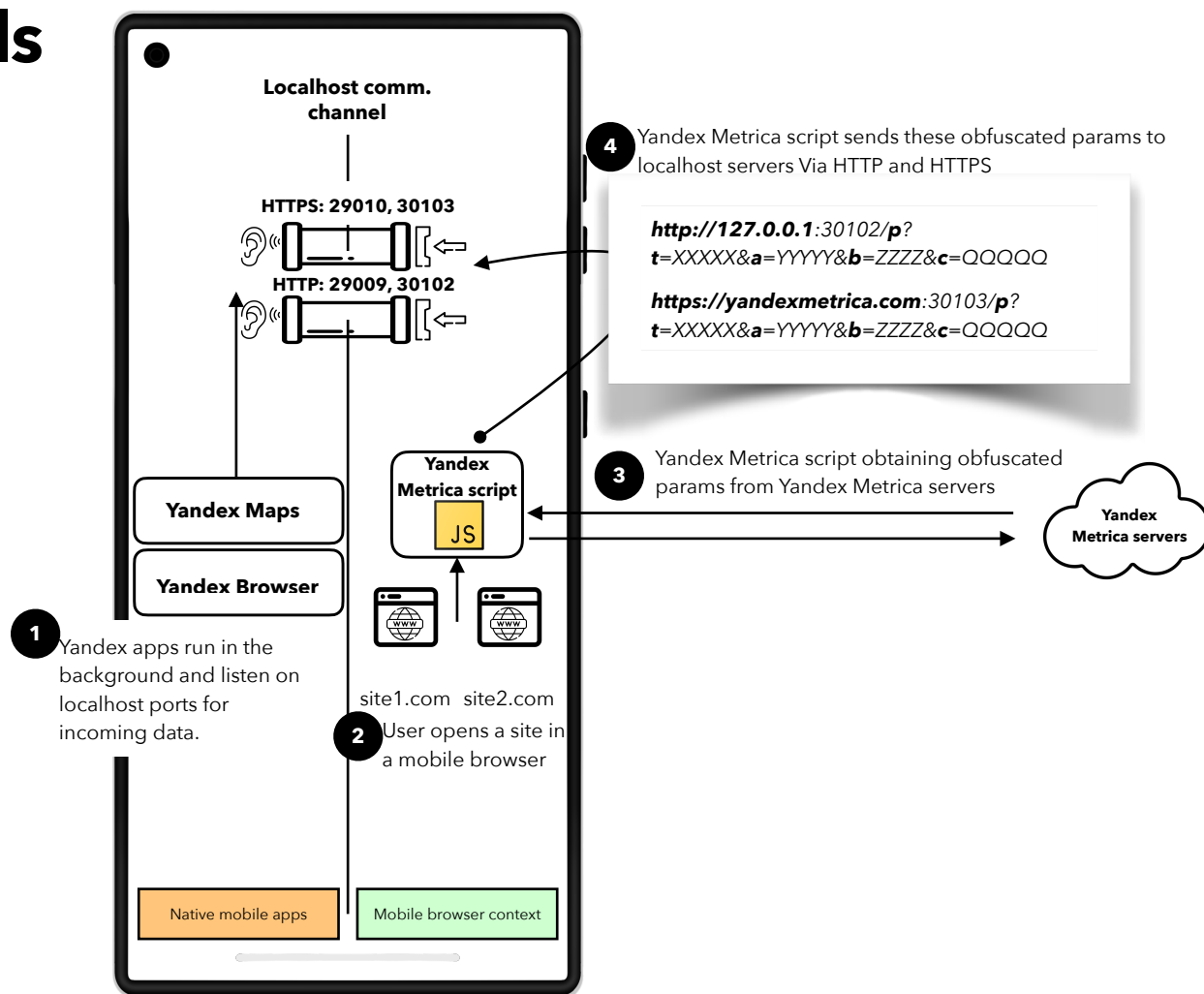
Technical Details



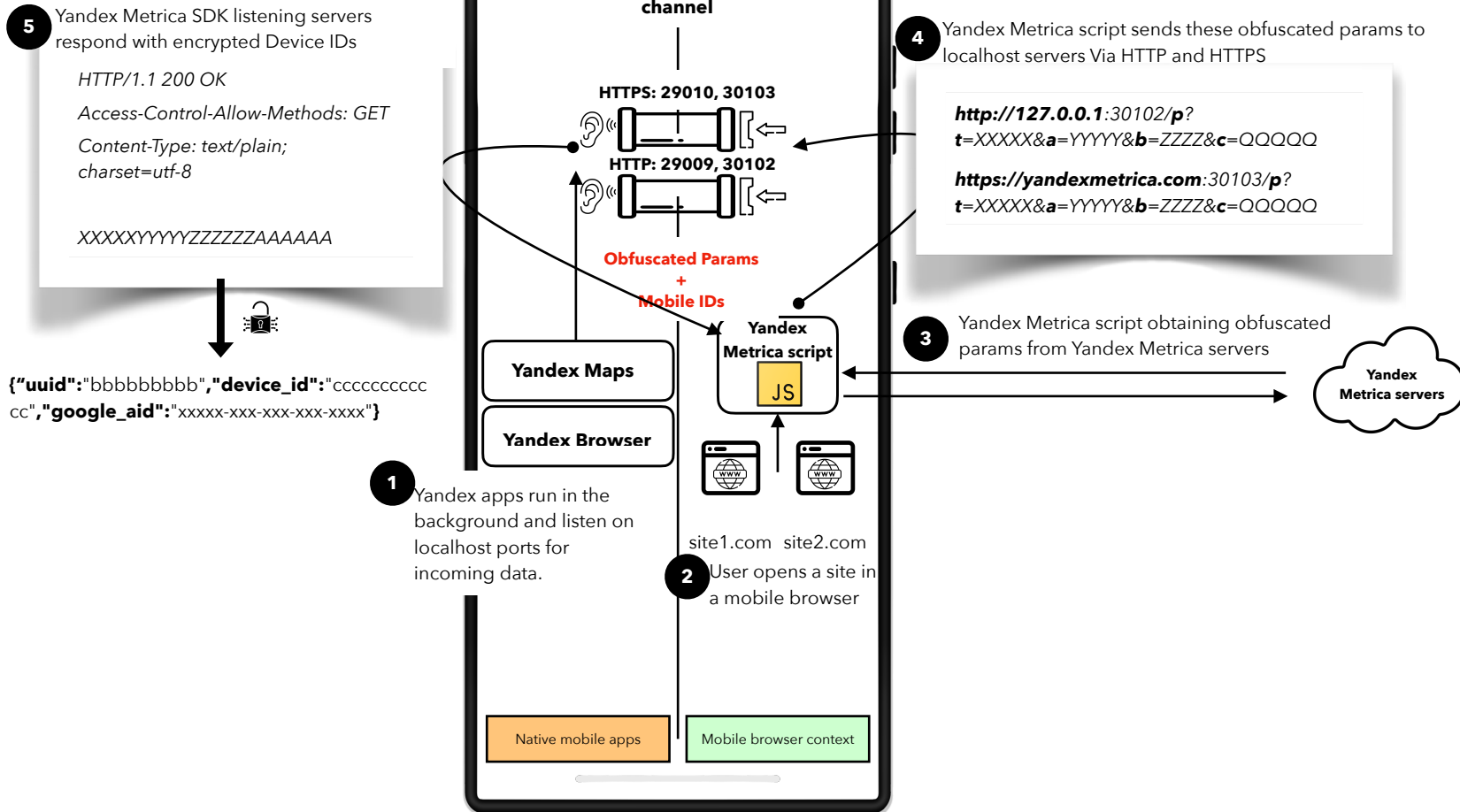
Technical Details



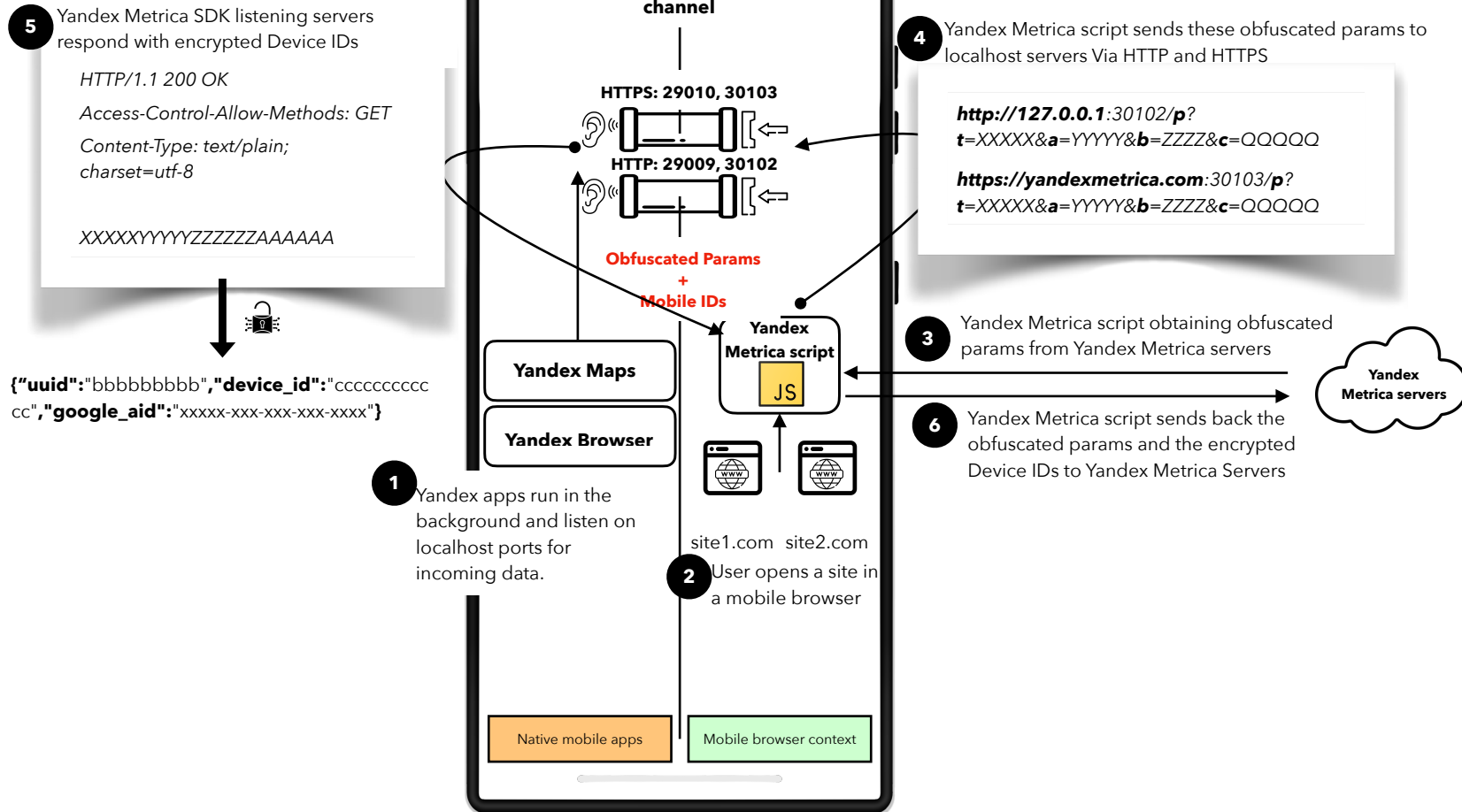
Technical Details



Technical Details



Technical Details



Technical Details: Domain Name Points to Localhost

www.milliyet.com.tr outbound to 127.0.0.1:30102 not encrypted

```
GET /p?t=UV|L7,! "T[rwe&D_>ZIb\aw#98Y.PC6k&a=08ej58gw+
XCJg [REDACTED]XBQKn8FjQeBC8YuC&b=YT+
Cyq2SzbgtX0dxeohWGr [REDACTED]VbPC/Nj&c=7 [REDACTED] HTTP/1.1^M
```

www.milliyet.com.tr outbound to 127.0.0.1:29009 not encrypted

```
GET /p?t=UV|L7,! "T[rwe&D_>ZIb\aw#98Y.PC6k&a=08ej58gw+
XCJg [REDACTED]XBQKn8FjQeBC8YuC&b=YT+
Cyq2SzbgtX0dxeohWGr [REDACTED]VbPC/Nj&c=7 [REDACTED] HTTP/1.1^M
```

www.milliyet.com.tr outbound to yandexmetrica.com:30103 encrypted

```
GET /p?t=UV|L7,! "T[rwe&D_>ZIb\aw#98Y.PC6k&a=08ej58gw+
XCJg [REDACTED]XBQKn8FjQeBC8YuC&b=YT+
Cyq2SzbgtX0dxeohWGr [REDACTED]VbPC/Nj&c=7 [REDACTED] HTTP/1.1^M
```

www.milliyet.com.tr outbound to yandexmetrica.com:29010 encrypted

```
GET /p?t=UV|L7,! "T[rwe&D_>ZIb\aw#98Y.PC6k&a=08ej58gw+
XCJg [REDACTED]XBQKn8FjQeBC8YuC&b=YT+
Cyq2SzbgtX0dxeohWGr [REDACTED]VbPC/Nj&c=7 [REDACTED] HTTP/1.1^M
```

yandexmetrica.com		
	A	Search
 Culiacan, Mexico Megacable	127.0.0.1	✓
 Santa Cruz do Sul, Brazil Claro	127.0.0.1	✓
 Paterna de Rivera, Spain ServiHosting	127.0.0.1	✓
 Manchester, United Kingdom Ancar B	127.0.0.1	✓
 Lille, France Completel SAS	127.0.0.1	✓
 Diemen, Netherlands Tele2 Nederland	127.0.0.1	✓
 Leipzig, Germany Universitaet Leipzig	127.0.0.1	✓
 Cullinan, South Africa Liquid	127.0.0.1	✓
 Vladivostok, Russia Rostelecom	127.0.0.1	✓
 Rawalpindi, Pakistan CMPak	127.0.0.1	✓
 Coimbatore, India		

Technical Details: Delayed Activation

```
ru.yandex.metro inbound to startup.mobile.yandex.net:443
```

```
HTTP/1.1 200 Ok^M
```

```
GET /analytics/startup?deviceid=75[REDACTED]de&adv_id=d01143[REDACTED]
e7a61a34bf4>&oaaid=&yandex_adv_id=&app_set_id=753[REDACTED]52ede&app_set_id_scope=
app&app_platform=android&protocol_version=2&analytics_sdk_version_name=5.2.1&model=AOSP on sargo&
manufacturer=Google&os_version=12&screen_width=2220&screen_height=1080&screen_dpi=440&scalefactor
=2.75&locale=en_US&device_type=phone&queries=1&query_hosts=2&features=ec,eg,pi,s,pc,fc,lc,lbs,g,h,tht
,wa,wc,ca,si,sl,ilc,gplc,up,ucfb,ues,ures,cai,caico,sp&s=1&app_id=ru.yandex.metro&lc=1&app_debuggable
=0&sl=1&country_init=es&uuiid=e6[REDACTED]5e&time=1&requests=1&stat_sending=1&
permissions=1&ilc=1&up=1&ucfb=1&ues=1&ures=1&rp=1&tht=1&cc=1&mak=1&d=1&pc=1&app_system_flag=0&aic=1&
at=1&ec=1&eg=1&su=1 HTTP/1.1^M
```

```
User-Agent: com.yandex.mobile.metrica.sdk/5.2.1.45002274 (Google AOSP on sargo; Android 12)^M
```

```
Accept-Encoding: encrypted^M
```

```
Accept: application/json^M
```

```
Host: startup.mobile.yandex.net^M
```

```
Connection: Keep-Alive^M
```

```
HTTP/1.1 200 Ok^M
Content-Encoding: encrypted^M
Content-Length: 1312^M
Content-Type: application/octet-stream^M
```

```
"device_id": {
  "hash": "18[REDACTED]"
},
"stat_sending": {
  "disabled_reporting_interval_seconds": 86400
},
...
"socket": {
  "ports_http": [
    30102,
    29009
  ],
  "launch_delay_seconds": 0,
  "min_successful_request_interval_seconds": 86400,
  "min_failed_request_interval_seconds": 3600,
  "token": "UV|L7,!\"T[rwe&D_>Zib\\aW#98Y.PC6k\"",
  "first_delay_seconds": 259200,
  "ports_https": [
    30103,
    29010
  ],
  "seconds_to_live": 86400,
  "ports": [
    30102,
    29009
  ]
},
```

Technical Details: Delayed Activation

```
ru.yandex.metro inbound to startup.
```

```
HTTP/1.1 200 OK^M
```

```
GET /analytics/startup?deviceid=75
```

```
e7a61a34bf4>&oaaid=&yandex_adv_i
```

```
app&app_platform=android&protoc
```

```
manufacturer=Google&os_version=
```

```
=2.75&locale=en_US&device_type=phone&queries=1&query_hosts=2&features=ec,eg,pi,s,pc,fc,lc,lbs,g,h,tht
```

```
User
```

```
Accep
```

```
Acce
```

```
Host
```

```
Conn
```

```
$ adb shell su -c netstat -untap | grep -wE "29009|29010|30102|30103"
```

tcp6	0	0	[::]:29009	[::]:*	LISTEN	11204/com.yandex.browser:AppMetrica
tcp6	0	0	[::]:29010	[::]:*	LISTEN	11204/com.yandex.browser:AppMetrica
tcp6	0	0	[::]:30102	[::]:*	LISTEN	10787/ru.yandex.yandexnavi:AppMetrica
tcp6	0	0	[::]:30103	[::]:*	LISTEN	10787/ru.yandex.yandexnavi:AppMetrica

```
HTTP/1.1 200 OK^M
Content-Encoding: encrypted^M
Content-Length: 1312^M
Content-Type: application/octet-stream^M
```

```
"device_id": {
```

```
": {
```

```
reporting_interval_seconds": 86400
```

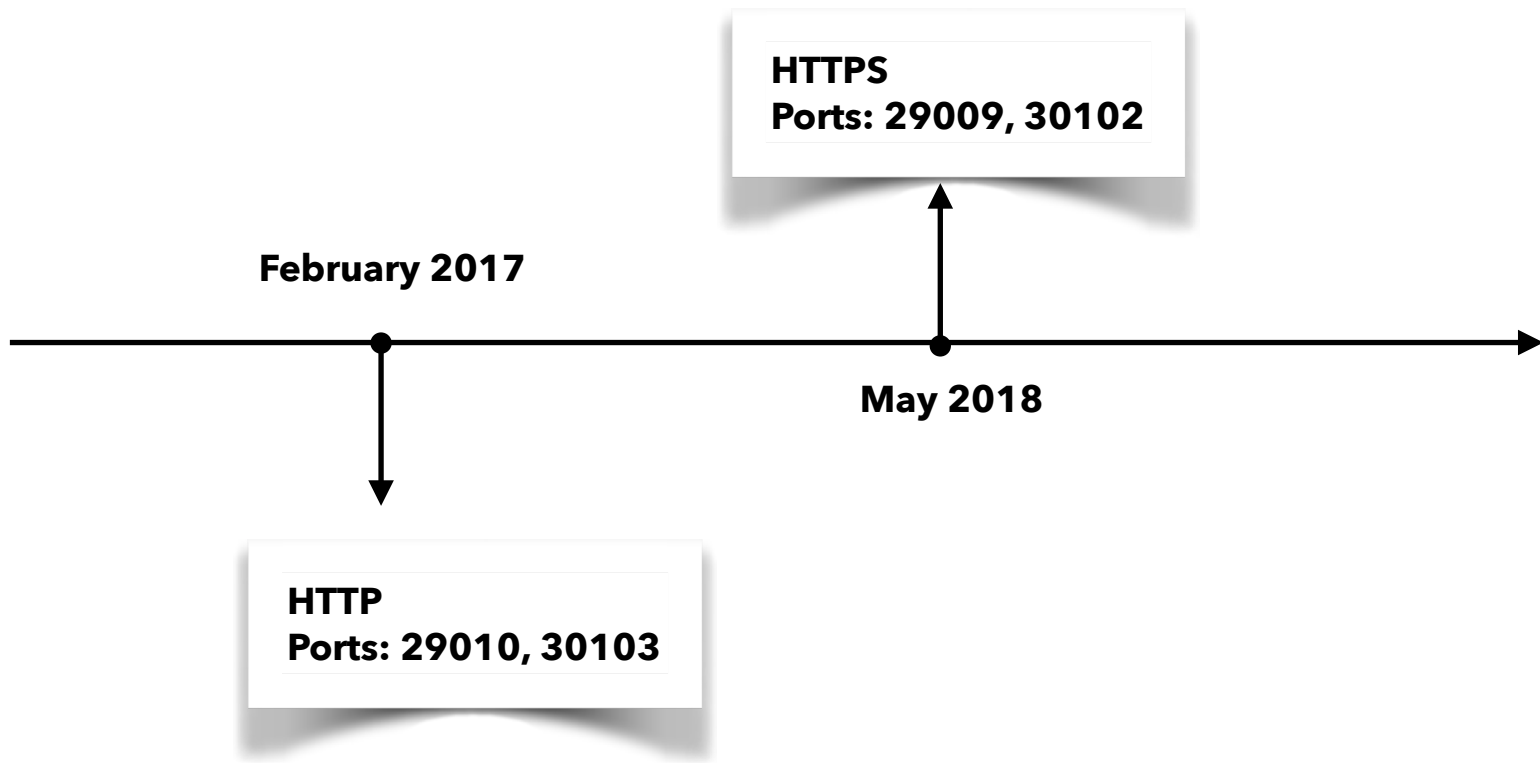
```
"ports_http": [
```

```
30102,
29009
```

```
]
},
```

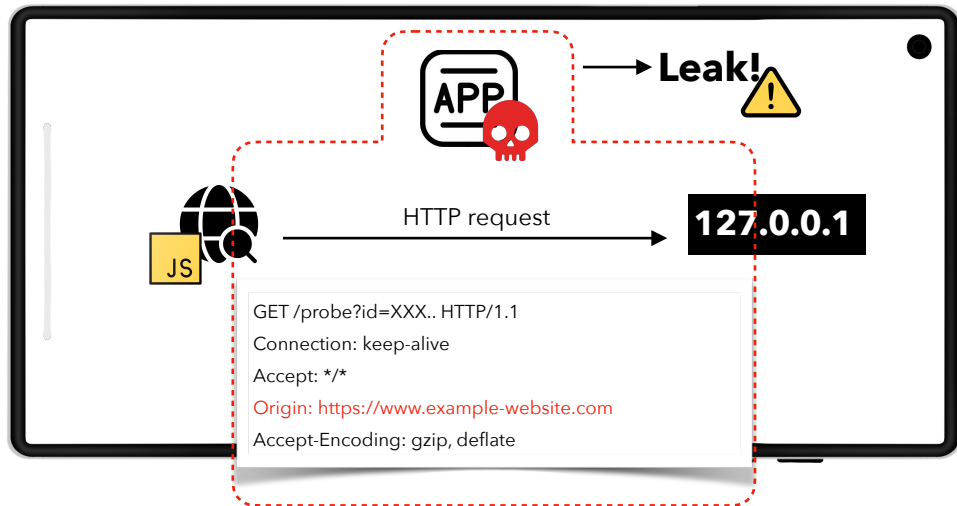
.....Wait for 3 days!

Yandex's Side Channel: Behaviour Across Time



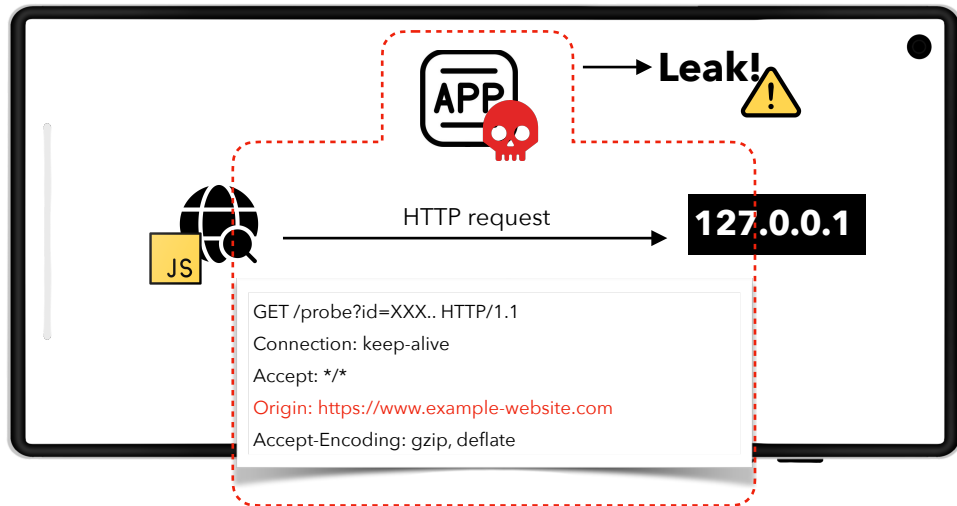
Browsing History Leak

- Malicious apps can intercept HTTP/ WebSocket based localhost requests
- Origin headers reveal web browsing activity



Browsing History Leak

- Malicious apps can intercept HTTP/WebSocket based localhost requests
- Origin headers reveal web browsing activity



Affected Sites – Real-World Scale of Abuse

 Meta → 11 Million

 Yandex → 1 Million

Affected Sites – Real-World Scale of Abuse



 United States (New York vantage point)

- Meta Pixel:
 - 17,223 sites with tracker
 - 13,468 sites (78.2%) initiate localhost **before consent**
- Yandex Metrica:
 - 1,312 sites with tracker
 - 1,095 sites (83.5%) initiate localhost **before consent**

Affected Sites – Real-World Scale of Abuse



United States (New York vantage point)

- Meta Pixel:
 - 17,223 sites with tracker
 - 13,468 sites (78.2%) initiate localhost **before consent**
- Yandex Metrica:
 - 1,312 sites with tracker
 - 1,095 sites (83.5%) initiate localhost **before consent**



Europe (Frankfurt vantage point)

- Meta Pixel:
 - 15,677 sites with tracker
 - 11,890 sites (75.8%) initiate localhost **before consent**
- Yandex Metrica:
 - 1,260 sites with tracker
 - 1,064 sites (84.4%) initiate localhost **before consent**

Affected Sites – Real-World Scale of Abuse



These results suggest that localhost abuse is not only real—it's global, widespread, and potentially without meaningful user awareness or control.



Un

- 17,223 sites with tracker
- 13,468 sites (78.2%) initiate localhost **before consent**
- Yandex Metrica:
 - 1,312 sites with tracker
 - 1,095 sites (83.5%) initiate localhost **before consent**

- 15,677 sites with tracker
- 11,890 sites (75.8%) initiate localhost **before consent**
- Yandex Metrica:
 - 1,260 sites with tracker
 - 1,064 sites (84.4%) initiate localhost **before consent**

Responsible Disclosure & Platform Mitigations

- Coordinated disclosure led to rapid mitigations by Chrome, Firefox, DuckDuckGo, Brave

Browser	Yandex Affected	FB Affected	Mitigations
 Chrome 136			Chrome 137 blocks abused ports + SDP munging / Local Network Access (LNA)
 Firefox 138			Port blocking in version 139
 DuckDuckGo	 Minimal		Blocklist amended
 Brave			Blocklist + localhost user prompts since 2022

Response to a Request for a Network

- Coordinating

Browse



Chrome



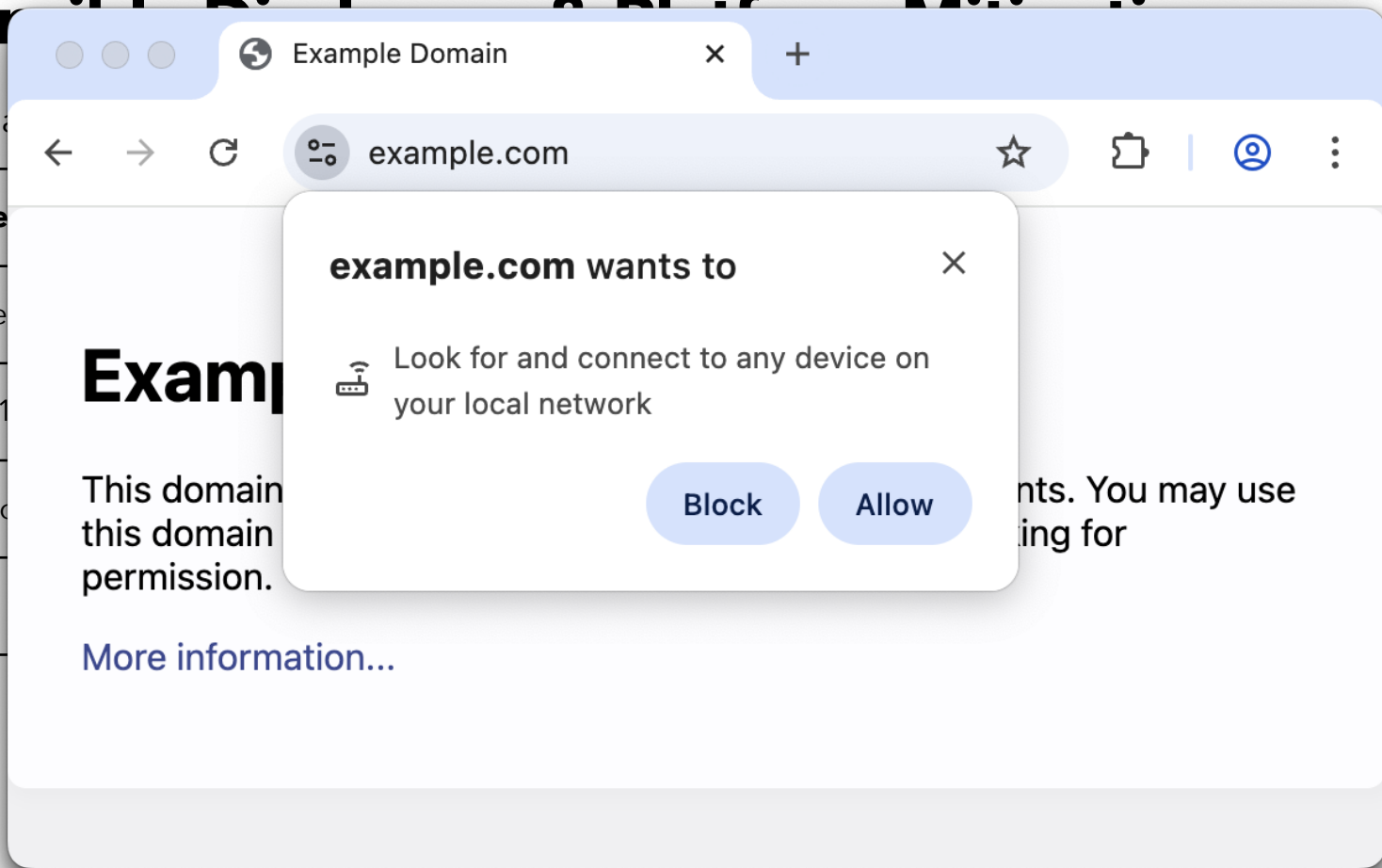
Firefox 1



DuckDuckGo



Brave



Responsible Disclosure & Platform Mitigations

- Coordinated disclosure led to rapid mitigations by Chrome, Firefox, DuckDuckGo, Brave

Browser	Yandex Affected	FB Affected	Mitigations
 Chrome 136			Chrome 137 blocks abused ports + SDP munging / Local Network Access (LNA)
 Firefox 138			Port blocking in version 139
 DuckDuckGo	 Minimal		Blocklist amended
 Brave			Blocklist + localhost user prompts since 2022

Responsible Disclosure & Platform Mitigations

- Coordinated disclosure led to rapid mitigations by Chrome, Firefox, DuckDuckGo, Brave

Browser	Yandex Affected	FB Affected	Mitigations
 Chrome 136			Chrome 137 blocks abused ports + SDP munging / Local Network Access (LNA)
 Firefox 138			Port blocking in version 139
 DuckDuckGo	 Minimal		Blocklist amended
 Brave			Blocklist + localhost user prompts since 2022

**Platform-level issues remain unresolved.
No user prompts, weak IPC policies.**



Platform Scope: Android Only?



Platform Scope: Android Only?



Confirmed: Only Android was affected in our real-world tests

→ Meta and Yandex scripts targeted Android apps and browsers

Platform Scope: Android Only?



Confirmed: Only Android was affected in our real-world tests

→ Meta and Yandex scripts targeted Android apps and browsers

No abuse detected on iOS

→ Despite similar localhost access being technically possible

Platform Scope: Android Only?



Confirmed: Only Android was affected in our real-world tests

→ Meta and Yandex scripts targeted Android apps and browsers

No abuse detected on iOS

→ Despite similar localhost access being technically possible

iOS differences:

→ Background execution + stricter platform controls

Platform Scope: Android Only?



Confirmed: Only Android was affected in our real-world tests

→ Meta and Yandex scripts targeted Android apps and browsers

No abuse detected on iOS

→ Despite similar localhost access being technically possible

iOS differences:

→ Background execution + stricter platform controls

Other surfaces like desktop browsers or smart TVs may also be at risk

→ But *not yet investigated*

Media Impact

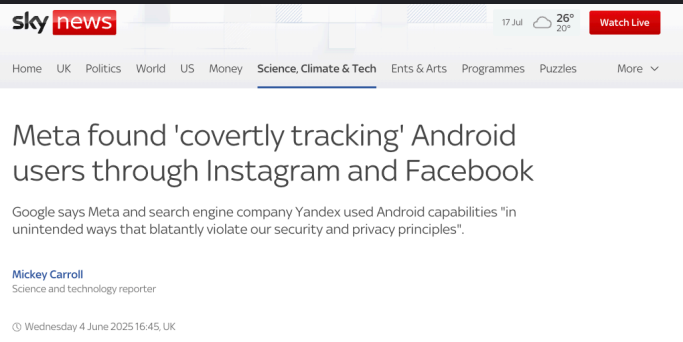
ars TECHNICA

AI BIZ & IT CARS CULTURE GAMING HEALTH POLICY SCIENCE SECURITY SPACE TECH **FORUM** [SUBSCRIBE](#) 🔍 🔔 [SIGN IN](#)

🗯️ THE SPY WHO TRACKED ME

Meta and Yandex are de-anonymizing Android users' web browsing identifiers

Abuse allows M
DAN GOODIN · 3 JUN 2025



The Washington Post
Democracy Dies in Darkness

Help Desk [Tech in Your Life](#) [Your Data and Privacy](#) [Internet Access](#) [What's New](#) [Ethical Issues](#) [Ask a Question](#)

Meta found a new way to violate your privacy. Here's what you can do.

Even hardened digital privacy veterans said they were stunned by Meta's tactics.

June 6, 2025

🕒 5 min 📄 Summary 📌 📖 465



EL PAÍS

Tecnología

[TU TECNOLOGÍA](#) · [CIBERSEGURIDAD](#) · [PRIVACIDAD](#) · [INTELIGENCIA ARTIFICIAL](#) · [INTERNET](#) · [GRANDES TECNOLÓGICAS](#) · [ÚLTIMAS NOTICIAS](#)

[PRIVACIDAD EN INTERNET](#) >

Este es el método oculto con el que Meta rastrea sin permiso la navegación en móviles (también en modo incógnito o con VPN)

Un grupo de investigadores descubre el sistema que usan las apps de Instagram y Facebook desde septiembre de 2024 para reunir el historial web de cada usuario en sus dispositivos Android



Media Impact

ars TECHNICA

AI BIZ & IT CARS CULTURE GAMING HEALTH POLICY SCIENCE SECURITY SPACE TECH **FORUM** [SUBSCRIBE](#) 🔍 [SIGN IN](#)

THE SPY WHO TRACKED ME

Meta and Yandex are de-anonymizing Android users' web browsing identifiers

Abuse allows M

sky news

17 Jul 26° 20°

Watch Live

"The developers [Meta & Yandex] in this report are using capabilities present in many browsers across iOS and Android in unintended ways that blatantly violate our security and privacy principles" – Google Representative

The Washington Post
Democracy Dies in Darkness

Help Desk Tech in Your Life Your Data and Privacy Internet Access What's New Ethical Issues Ask a Question

Meta found a new way to violate your privacy. Here's what you can do.

Even hardened digital privacy veterans said they were stunned by Meta's tactics.

June 6, 2025

🕒 5 min 📄 Summary 📌 📄 465



EL PAÍS

Tecnología

[TU TECNOLOGÍA](#) [CIBERSEGURIDAD](#) [PRIVACIDAD](#) [INTELIGENCIA ARTIFICIAL](#) [INTERNET](#) [GRANDES TECNOLÓGICAS](#) [ÚLTIMAS NOTICIAS](#)

PRIVACIDAD EN INTERNET >

Este es el método oculto con el que Meta rastrea sin permiso la navegación en móviles (también en modo incógnito o con VPN)

Un grupo de investigadores descubre el sistema que usan las apps de Instagram y



Conclusion

Android's isolation guarantees are incomplete

→ Localhost allows covert cross-process communication with no user visibility.

We reveal the first real-world deployment of covert web & app identity linkage

→ Meta & Yandex bridged cookies to AIDs and logged-in identities silently—impacting billions.

Platform assumptions must be re-evaluated

→ “No permission = no access” breaks when localhost is in the middle.

Existing defenses are ineffective

→ Sandboxing, incognito mode, and permission models offer no meaningful protection

Fixes are partial—attack surface remains

→ Addressing this requires platform-centric solutions: deeper sandbox enforcement, consent-aware networking, and web-to-app auditing.

Thank you!

Aniketh Girish
aniketh.girish@networks.imdea.org



SCAN ME

Disclosure website