طبق بررسی که انجام دادم اخیرا برای پیاده‌سازی برنامه‌های موبایل برای افزایش portability از روش پیاده‌سازی Hybrid یا Html-based استفاده می‌کنند. مزیتی که این روش‌ها دارد این است که روی پلتفرم‌های اندروید، iOS و ویندوز قابل اجرا هستند. پیاده‌سازی آنها هم با زبان‌های برنامه‌نویسی HTML5، CSS، JS[1] است.

http://www.androidauthority.com/html-5-vs-native-android-app-607214/

مقاله‌های زیر حمله‌های صورت گرفته روی این نوع پیاده‌سازی‌ها را بررسی کرده اند:

- Attacks onWebView in the Android system, 2011

در این مقاله انواع حملاتی که امکان دارد روی نرم‌افزارهایی که از API خاص WebView در اندروید یا iOS استفاده می کنند، بررسی شده است و با جزئیات کامل همراه با کد نحوه استفاده از این API بیان شده است. WebView این امکان را به توسعه دهنده می دهد تا یک مرورگر کوچک در برنامه خود داشته باشد و بتواند صفحات وب همراه با کدهای JS,CSS و HTML را نمایش دهد. علاوه بر آن این امکان را می دهد که از تابع های تعریف شده به زبان جاوا در اسکریپت های JS و هم از اسکریپت های JS در کد جاوا استفاده کرد. مثلا می توان با کد جاوا یک کلاس نوشت که نحوه دسترسی به Contactها را مدیریت کرد و از آن می توان در کد JS نیز استفاده کرد. بدون در نظر گرفتن پیاده سازی UI با WebView، همین وجود این API در appهای با native code هم می تواند یک open problem باشد چون که من مقاله ای ندیدم که آسیب پذیری‌های این حوزه را با روش concolic بررسی کرده باشد.

- Breaking and Fixing Origin-Based Access Control in Hybrid Web/Mobile Application Framework, 2014

این مقاله روی برنامه های hybrid تمرکز دارد. در این برنامه ها یک مرورگر وب داخلی مثل webView دارند که کدهای وب را اجرا می کند و شامل امکانی به نام bridge است که کد های وب می تواند از مرورگر گذر کرده و به منابع local دسترسی داشته باشد مثل دوربین. برای امکان bridging لازم است که آنها همان سطح دسترسی کل برنامه را داشته باشند و این خود عامل ایجاد حمله می شود. (حمله fracking) مهاجم با تزریق یک محتوای وب و از طریق bridging می تواند به local resourceها دسترسی پیدا کند.

- Bifocals: Analyzing WebView vulnerabilities in android application, 2013

[1] Java Script

در این مقاله دو آسیب پذیری excess authorization و file-based cross-zone scripting بررسی شده اند و ابزار Bifocals برای یافتن این آسیب پذیری ها ارائه شده است. البته روش concolic نیست.

- 2014, Code Injection Attacks on HTML5-based Mobile Apps Characterization, Detection and Mitigation

در این مقاله به دسته دیگری از حملات پرداخته شده که شبیه به حملات XSS است. با این تفاوت که در XSS مرورگر با سرور فقط تبادل اطلاعات می کرد و در محیط sandboxکدها را اجرا می کرد پس به local resourceها نمی توانست دسترسی داشته باشند. اما در حملات تزریق کد بررسی شده راه‌های دریافت کد مختلف است از جمله other apps, 2D barcode, Wi-Fi access points, other mobile devices, data sent by others or downloaded from external resources هستند علاوه بر آن به دلیل امکان bridging که در بالا گفته شد می توانند به local resourceها دسترسی داشته باشند. در این مقاله یک برنامه ساده که با خواندن بارکد در آن کد تزریق می شود هم بررسی شده است. همچنین کانالهای مختلفی که از آنها می توان کد تزریق کرد را بررسی کرده است. در نهایت هم ابزاری ارائه شده که به واسطه آن این آسیب پذیری کشف می شود که تا کید بر تحلیل ایستا است و از روش concolicهم استفاده نمی شود.

- ❖ در کل به نظرم اگر تاکید را روی webView و همچنین برنامه های Hybrid یا Html-based بگذاریم بهتر باشد چون هم مبحث جدید است و هم من مقاله ای که آسیب پذیری‌های این حوزه را که با روش concolicبررسی کند نیافتم. همچنین به نظرم کار را محدود به آسیب‌پذیری تزریق کد کنیم که در مقاله آخر بررسی شده بود.