فصلنامهٔ امنیت پژوهی (علمی-پژوهشی)
سال بیست‌وپنجم: شماره ۹۳- بهار ۵
صص: ۸۵ - ۱۱۶
تاریخ دریافت:۱۴۰۴/۰۳/۱۰
تاریخ پذیرش:۱۴۰۵/۰۳/۱۵

Amniat Pajouhi: Security Research Quarterly
Vol.24, No.92,Winter 2025
P-P: 85 - 116

بررسی امنیت ابری در مرورگرهای سبک تلفن همراه و ارائه یک الگوی تهدید برای شناسایی و ارزیابی آسیب‌پذیری‌های احتمالی

علیرضا چمکوری[۱]
مجید حاجیانی[۲]

چکیده

مرورگرهای سبک تلفن همراه به‌دلیل سرعت بارگذاری و سهولت دسترسی به سرویس‌های ابری، به‌طور گسترده‌ای مورداستفاده قرار می‌گیرند. باوجوداین مزایا، استفاده از حافظه نهان بزرگ‌تر و کاهش وابستگی به پلاگین‌ها می‌تواند آسیب‌پذیری‌های امنیتی جدیدی ایجاد کند. در این پژوهش، یک الگوی تهدید طراحی و چهار مرورگر پرکاربرد شامل UC Browser، Dolphin ، CM Browser و مرورگر پیش‌فرض سامسونگ از منظر امنیتی ارزیابی شدند. نتایج نشان داد این مرورگرها در برخی موارد داده‌های حساس کاربران مانند تاریخچه مرور، محتوای ایمیل و اطلاعات مالی را در معرض دسترسی غیرمجاز قرار می‌دهند. همچنین مشخص شد ذخیره‌سازی نامناسب فایل‌ها می‌تواند بستر حملات فیشینگ و تزریق محتوا را فراهم کند. یافته‌های این پژوهش می‌تواند مبنای بهبود طراحی مرورگرهای سبک در آینده قرار گیرد و به توسعه‌دهندگان در ارتقای امنیت کاربران کمک نماید.

واژگان کلیدی: امنیت ابری تلفن همراه، مرورگرهای سبک تلفن همراه، الگوی تهدید، آسیب‌پذیری مرورگر، حملات فیشینگ، ذخیره‌سازی ناامن.

---

۱. نویسنده مسؤل، رشته مهندسی کامپیوتر گروه مهندسی کامپیوتر، واحد بوشهر، دانشگاه آزاد اسلامی، بوشهر، ایران chamkoori@iau.ac.ir

https://orcid.org/0000-0001-7156-0011

۲. دانشجوی دکتری پژوهشگری امنیت و مدرس دانشگاه Hajiani.114@gmail.com ،

مقدمه

در سال‌های اخیر، استفاده از دستگاه‌های تلفن همراه برای مرور اینترنت و دسترسی به سرویس‌های ابری[1] به‌طور چشمگیری افزایش یافته است. این تغییر الگو موجب شده مرورگرهای سبک به‌دلیل سرعت بالاتر و مصرف کمتر منابع، به گزینه‌ای محبوب برای کاربران تبدیل شوند. مرورگرهای سبک با بهره‌گیری از حافظه نهان بزرگ‌تر و حذف وابستگی به برخی پلاگین‌ها، تجربه کاربری روان‌تری فراهم می‌کنند. با این حال، این مزایا ممکن است به قیمت کاهش سطح امنیت داده‌های کاربران تمام شود.

مرورگرها یکی از حساس‌ترین نرم‌افزارهای کاربردی محسوب می‌شوند، زیرا به اطلاعات شخصی و مالی کاربر دسترسی مستقیم دارند. درنتیجه، وجود هرگونه ضعف امنیتی در طراحی یا نحوه ذخیره‌سازی داده‌ها می‌تواند پیامدهای جدی ازجمله دسترسی غیرمجاز به اطلاعات بانکی یا تاریخچه مرور وب داشته باشد. در این پژوهش تلاش شده است تا امنیت مرورگرهای سبک موبایل با تمرکز بر نحوه ذخیره‌سازی داده‌ها و مدیریت حافظه نهان ارزیابی شود.

بیان مسئله و ضرورت پژوهش

مسئله موردتوجه در این پژوهش این است که با وجود محبوبیت مرورگرهای سبک تلفن همراه، بررسی‌های اولیه نشان می‌دهد که این مرورگرها از نظر امنیتی در سطح مرورگرهای اصلی همچون موزیلا، فایرفاکس و گوگل کروم[2] نیستند. در بسیاری موارد، داده‌های حساس در حافظه نهان یا فضای ذخیره‌سازی خارجی دستگاه ذخیره می‌شوند که می‌تواند توسط برنامه‌های دیگر یا مهاجمان مورد سوءاستفاده قرار گیرد. در این حالت امنیت داده‌ها دچار چالش خواهد شد و کاربر تلفن همراه، همواره در دلهره‌ای از سرقت و یا دسترسی مهاجمان به داده‌هایش را دارد؛ بنابراین، پرسش اصلی این پژوهش این است که:

۱. آیا مرورگرهای سبک در ذخیره‌سازی داده‌های حساس کاربران از سازوکارهای ایمن و استاندارد بهره می‌برند؟

1. CLOUD SYSTEM
2. Mozilla Firefox & Google Chrome

۲. چه آسیب‌پذیری‌هایی در نحوه مدیریت حافظه نهان این مرورگرها وجود دارد؟
۳. آیا می‌توان الگوی تهدید ارائه کرد که آسیب‌پذیری‌های احتمالی را شناسایی و ارزیابی کند؟

پاسخ به این پرسش‌ها می‌تواند درک دقیق‌تری از سطح امنیت مرورگرهای سبک تلفن همراه ارائه دهد و مبنای بهبود طراحی آن‌ها قرار گیرد.

اهمیت و ضرورت پژوهش

گسترش روزافزون استفاده از دستگاه‌های تلفن همراه و مرورگرهای سبک، فرصت‌های جدیدی برای دسترسی سریع و آسان به سرویس‌های ابری فراهم کرده است. با این حال، این روند با چالش‌های امنیتی مهمی همراه است. مرورگرهای سبک به‌دلیل طراحی ساده‌تر و استفاده از حافظه نهان گسترده، در برابر تهدیدات امنیتی آسیب‌پذیرتر هستند. در صورت بهره‌برداری مهاجمان از این آسیب‌پذیری‌ها، اطلاعات حساسی مانند تاریخچه مرور، محتوای پیام‌های شخصی و داده‌های مالی کاربران می‌تواند در معرض افشا یا سرقت قرار گیرد.

اهمیت این پژوهش در آن است که با ارائه یک الگوی تهدید و شناسایی نقاط ضعف موجود، زمینه برای ارتقای امنیت مرورگرهای سبک فراهم می‌شود. یافته‌های این پژوهش می‌تواند مورداستفاده توسعه‌دهندگان مرورگرها، متخصصان امنیت سایبری و حتی سیاست‌گذاران حوزه فناوری اطلاعات قرار گیرد. ازآنجاکه مرورگرهای سبک نقش مهمی در تجربه روزمره کاربران تلفن همراه ایفا می‌کنند، تقویت امنیت آن‌ها نه‌تنها موجب افزایش اعتماد کاربران خواهد شد، بلکه به کاهش ریسک تهدیدات سایبری در سطح وسیع‌تر نیز کمک خواهد کرد.

پیشینه پژوهش

پژوهش‌های متعددی درزمینهٔ امنیت مرورگرها و وب در سال‌های اخیر انجام شده است. بیشتر این مطالعات تمرکز خود را بر مرورگرهای رایانه‌های شخصی یا مرورگرهای شناخته‌شده‌ای مانند Chrome و Firefox گذاشته‌اند. نتایج این مطالعات نشان داده است که ضعف‌های معماری و نحوه مدیریت داده‌ها می‌تواند زمینه‌ساز حملات مختلف

ازجمله فیشینگ، جعل درخواست میان‌وبگاهی (CSRF) و سرقت اطلاعات از طریق کلیک‌جکینگ شود.

برای نمونه:

- Wadkar و Mishra (2014) سامانه‌ای را ارائه کردند که با پایش فراخوانی‌های سامانه‌ای مرورگر، مانع نشت اطلاعات شخصی کاربر می‌شود.
- Virvilis و همکاران (2015) تأثیر روش‌های فیلترینگ فهرست سیاه در مرورگرها را بررسی کردند و نشان‌دادند که این روش‌ها محدودیت‌های جدی دارند.
- Amrutkar و Traynor (2015) الگویی از تهدید برای مرورگرهای موبایلی معرفی کردند که می‌تواند ضعف‌های معماری و آسیب‌پذیری‌های ناشی از ذخیره‌سازی داده‌ها را آشکار کند.

علاوه بر این، تحقیقات اخیر به‌طور خاص به امنیت مرورگرهای سبک پرداخته‌اند و بیان کرده‌اند که استفاده گسترده از حافظه نهان و ذخیره‌سازی داده‌ها در فضای اشتراکی دستگاه‌های اندرویدی، زمینه‌ساز حملات متنوعی ازجمله بازیابی تاریخچه مرور و دسترسی غیرمجاز به اطلاعات حساس کاربران است.

باوجوداین مطالعات، بررسی‌های جامع در مورد مرورگرهای سبک تلفن همراه محدود است. این پژوهش با تمرکز بر چهار مرورگر پرکاربرد (UC Browser، Dolphin، CM Browser و مرورگر پیش‌فرض سامسونگ) تلاش می‌کند شکاف موجود در ادبیات را پر کند و آسیب‌پذیری‌های ناشناخته این مرورگرها را شناسایی نماید.

تعریف مفاهیم و چهارچوب نظری

مرورگرهای موبایل:

مرور یک صفحه وب نیاز به بارگذاریِ مجموعه‌های متعددی از منابع دارد، مانند اچ‌تی‌ام‌ال، سی‌اس‌اس و جاوااسکریپت[1] و فایل‌های رسانه‌ای.

به‌عنوان‌مثال، مطابق با وانگ و همکارانش.، بارگذاری این منابع با توجه به تفاوت‌های معماری و محدودیت‌های محاسباتی می‌تواند بر روی دستگاه‌های تلفن همراه آهسته‌تر از رایانه‌های شخصی باشد.[2]

1. HTML، CSS، JavaScript

سرعت در طی مرور وب‌سایت یکی از نگرانی‌های اصلیِ کاربران است. به‌عنوان‌مثال، بنا به گزارش، یک ثانیه تأخیر در بارگذاری صفحه وب می‌تواند نتیجه را در دیدگاه‌های مربوط به صفحه وب ٪11 و رضایت مشتریان را ٪16 کاهش دهد مشاهدات مشابهی در مطالعات انجام شده توسط آمازون و گوگل نیز تکرار شد.[3]

مرورگرهای سبک برای بهبود سرعتِ مرور و به‌تبع آن بهبودِ کیفیت تجربه کاربران، در سمت سرویس‌گیرنده راه‌کارهای مفیدی را به کار می‌گیرند. این مسئله موجبِ ایجاد یک حافظه نهان بزرگ‌تر برای ذخیره‌سازی و اجتناب از هرگونه پلاگین[1] است که می‌تواند باعثِ تأخیر بارگذاریِ منابع وب شود. حافظه نهان یک حافظه موقت برای ذخیره‌سازیِ منابع وب دانلود شده است. اگر یک کاربر تلاش کند تا به URL یا همان صفحه وبی که قبلاً دیده، دسترسی پیدا کند، مرورگر چک می‌کند که آیا این صفحه در محتوای حافظه نهان وجود دارد یا نه! اگر محتوا یافت شود، مرورگر آن صفحه را از منابعِ حافظه نهان بارگذاری می‌کند، بنابراین، باعثِ صرفه‌جویی در زمان و منابع شبکه می‌شود.

مرورگرهای محبوب، مانند گوگل کروم، موزیلا فایرفاکس و اپرا، از استانداردهای ذخیره‌سازیِ وب برای ذخیره داده‌ها در حافظه نهان بهره می‌برند [4] ذخیره‌سازی وب را می‌توان به عنوان بخشی از HTML5 معرفی کرد که توسط اتحادیهٔ (کنسرسیوم) جهانی وب[2] استانداردسازی شده است. ذخیره‌سازی وب شامل دو بخش عمده می‌شود: ذخیره‌سازی محلی و جلسه ذخیره‌سازی که به ترتیب رفتار مشابهی مانند کوکی‌های ماندگار و جلسه کوکی‌ها از خودشان نشان می‌دهند. جلسه ذخیره‌سازی تا بازشدن صفحه وب، منابعِ وب را ذخیره می‌کند.[4] در موردِ ذخیره‌سازی محلی تا زمانی که مرورگر بسته است، حافظه نهان تولیدشده بر روی دستگاه‌ها باقی می‌ماند.

در هر دو محیطِ رایانه‌های شخصی و دستگاه‌های تلفن همراه، ذخیره‌سازی وب امنیت بیشتری نسبت به حافظه نهان مرورگر بومی دارد. با توجه به اعلام اتحادیهٔ (کنسرسیوم) جهانی، ذخیره‌سازی وب اگر به‌درستی اجرا شود، می‌تواند برای ذخیره اطلاعات حساس کاربر استفاده شود. در دستگاه‌های اندروید، ذخیره‌سازی وب معمولاً

1. PLUG-in
2. W3C

از حافظه داخلی دستگاه استفاده می‌کند (مانند داده/داده/ نام بسته/ فهرست راهنما (دایرکتوری)؛ بنابراین، این موارد ذخیره‌شده در حافظه نهان نمی‌توانند به‌غیراز برنامه مالک توسط هیچ کاربر و برنامه دیگری مورد دسترسی قرار گیرند.

با این حال، ذخیره‌سازی وب توسط حجم حافظه نهان محدود می‌شود. برای استفاده عمومی، اتحادیه (کنسرسیوم) جهانی وب توصیه می‌کند که از حجم ذخیره‌سازی پنج مگابیت در هر وب‌سایت استفاده شود، اما این حجم می‌تواند در هنگام اجرا بر روی دستگاه‌های تلفن همراه کاهش یابد.

مرورگرهای سبک معمولاً برای بهبود سرعت بارگذاریِ مرورگر بر روی حافظه‌های نهان بزرگ‌تر تکیه می‌کنند؛ بنابراین، این مرورگرها اغلب مقدار زیادی از داده‌های حافظه نهانِ خارج از ذخیره‌سازی وب را در ذخیره‌سازیِ خارجی انبار می‌کنند، مانند کارتِ SD.

در دستگاه‌های اندروید، حافظه داخلی به‌طورکلی برای داده‌های برنامه، امنیت ذخیره‌سازیِ محلی بیشتری در نظر گرفته است، چراکه به‌طور پیش‌فرض، داده‌های ذخیره‌شده تنها می‌توانند توسط برنامه خالق مورد دسترسی یا تغییر قرار گیرند.[4] در مقایسه، هرگونه منابعِ ذخیره‌شده در ذخیره‌سازیِ خارجی توسط هر برنامه‌ای که اجازه رید اکسترنال استوریج[1] را دارد، می‌توانند مورد دسترسی، تغییر یا حذف قرار گیرد.

جدول۱- مرورگرهای سبک.

S. no.	نام مرورگرها	نسخه.no	ذخیره‌شده در گوگل پلی[1] (در میلیون؛ ملاحظات از سپتامبر ۲۰۱۵)	ملاحظات
.۱	مرورگرUC	۱۰٫۶٫۲	۱۰۰-۵۰۰	
.۲	دلفین	۱۱٫۴٫۱۹	۵۰-۱۰۰	
.۳	مرورگرCM	۵٫۲۰٫۰۶	۱۰-۵۰	
.۴	مرورگرهای سهام سامسونگ	N/A	N/A	پیش نصب‌شده با تلفن همراه سامسونگ، بنابراین تعداد کل کاربران و نسخه نرم‌افزار نمی‌تواند شناسایی شود.

1. READ_EXTERNAL_STORAGE
1. Google play

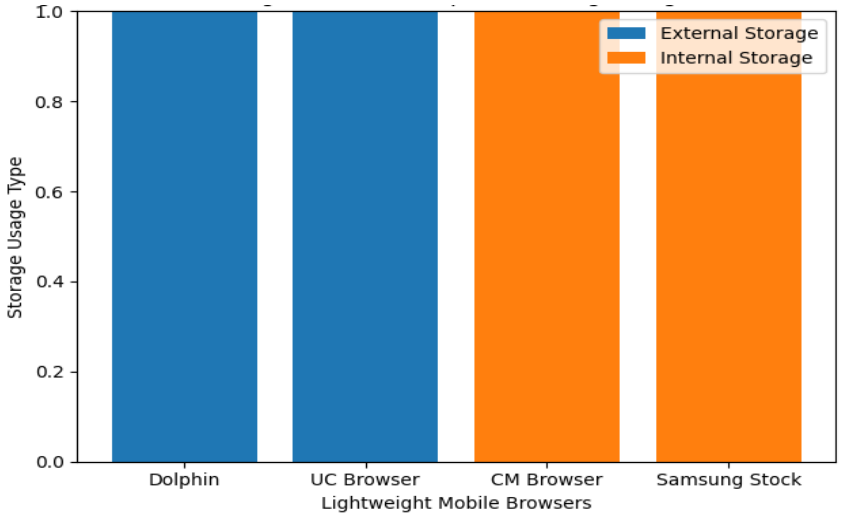
جدول۲ - حافظه نهان مورد هدف و مکان ذخیره‌سازی فایل‌های مرورگرها در این تحقیق

مرورگرها	مکان حافظه نهان مورد هدف	محتویات
دلفین	/sdcard/TunnyBrowser/cache/speeddial_covers /sdcard/TunnyBrowser/cache/tablist_cache /sdcard/TunnyBrowser/cache/webViewCache	-URLهای ذخیره‌شده به منظور دسترسی سریع -تصویربرداریِ فایل‌هایِ تصویری -تمام فایل‌های حافظه نهان (HTML, CSS, JavaScript, media)

مرورگرUC	//sdcard/UCDownloads/cache
	//sdcard/UCDownloads/config
	/sdcard/UCDownloads/offline/
		-تمام فایل‌های حافظه نهان
		-TrafficStatus.db; شامل زمان‌بندی و پاسخ ارتباط بین سرویس‌گیرنده و سرویس‌دهنده است.
		-ApplicationCache.db; شامل داده برای مدیریت بارگذاریِ حافظه نهان است.

مرورگرCM	/sdcard/CheetahBrowser/.data/	تاریخچه URL مرورگرها
مرورگرهای سهام سامسونگ	/data/data/com.sec.android.app.sbrowser/files/	تصویربرداری فایل‌هایِ تصویری

شکل شماره (۱): مقایسه محل ذخیره‌سازی حافظه نهان (Cache) در مرورگرهای سبک تلفن همراه اندرویدی.



همان‌طور که در شکل ۱ مشاهده می‌شود، مرورگرهای Dolphin وUC Browser بخش عمده‌ای از داده‌های حافظه نهان خود را در فضای ذخیره‌سازی خارجی دستگاه نگه‌داری می‌کنند. این نوع ذخیره‌سازی به‌دلیل قابلیت دسترسی سایر برنامه‌ها به حافظه خارجی، ریسک افشای اطلاعات، بازیابی داده‌های حساس و دست‌کاری محتوای کش را افزایش می‌دهد.

در مقابل، مرورگرهای CM Browser و مرورگر پیش‌فرض سامسونگ عمدتاً از حافظه داخلی برای ذخیره‌سازی داده‌های کش استفاده می‌کنند. این موضوع نشان‌دهنده سطح بالاتری از ایزوله‌سازی داده‌ها در این دو مرورگر است، هرچند نتایج سایر آزمایش‌ها نشان می‌دهد که استفاده از حافظه داخلی به‌تنهایی تضمین‌کننده امنیت کامل داده‌های کاربران نیست.

الگوی دشمنِ پیشنهادی و یک نمونه اولیه نرم‌افزار

الگوی دشمن

در یک کار اخیرِ انجام شده، مارتین و چو اولین الگوی دشمن برای خروج داده‌های پنهان اندروید را ارائه کردند. در این الگو، دشمن دارای قابلیت ره‌گیری، تزریق، ویرایش، حذف، رمزگذاری، رمزگشایی، انتقال و گوش‌دادن به ارتباطات کاربر است. در یک کار مرتبط D'Orazio و چو یک الگوی دشمن ارائه کردند که این الگوی قطاری از قابلیت‌های دنیای واقعیِ را دارد، ازجمله: مدیریت حقوق دیجیتال (DRM) مهاجم برای دستگاه‌های تلفن همراه. در این مقاله، ما یک الگوی ضعیف‌تر) و احتمالاً، واقعی‌تر(از آنچه مارتین و چو و D'Orazio و چو انجام داده‌اند، ارائه و نشان داد که دشمن به عنوان چنین الگویی نیز می‌تواند برای کشف و استفاده از آسیب‌پذیری در مرورگرهای سبک برای دستگاه‌های اندروید استفاده شود. در این راستا یک مرورگر موبایل را در صورتی ناامن در نظر می‌گیریم که هرکدام از این اهداف برآورده شود.

هدف ۱: دشمن تاریخچه URL مرورگر را می‌آموزد.

هدف ۲: دشمن عبارات جستجوی کاربر را می‌آموزد.

هدف ۳: دشمن محتوای صفحه وب (مانند محتوای ایمیل کاربر، اطلاعات حساب بانکی) را می‌آموزد.

هدف ۴: دشمن می‌تواند محتوای حافظه نهان (مانند فایل تصویری) را تغییر دهد.

نمونه اولیه نرم‌افزار

برای نشان‌دادن ابزارهای الگوی دشمنان، یک نمونه اولیه از برنامه اندروید آماده‌شده که قادر به خواندن و نوشتن فایل‌ها در حافظه به اشتراک‌گذاری شده خارجیِ دستگاه است (از طریق مجوزهایREAD_ EXTERNAL_STORAGE and WRITE_EXTERNAL_STORAGE). باملاحظه اینکه اگر به این برنامه دسترسی به مجوزِ WRITE_EXTERNAL_STORAGE داده شود، مجوزِ READ_ EXTERNAL_STORAGEبه‌طور خودکار واگذار شده است. به منظور انتقال داده از دستگاه کاربر، برنامه نیازمند مجوزِ اینترنت خواهد بود. با این حال، برنامه‌های اندروید نیازی به اطلاع و یا درخواست برای تصویب

کاربر برای این مجوز ندارند، زیرا این مجوز اگر در فایل بیانیه اظهار شود، به‌طور پیش‌فرض به تمام برنامه‌ها اعطا می‌شود. مجوز دیگری که این برنامه نیاز دارد ACCESS_NETWORK_STATEاست که به دشمن اجازه می‌دهد تا نوع شبکه‌ای که دستگاه به آن وصل است را بشناسد (مانند WiFi و یا 3G/4G)؛ به‌عبارت‌دیگر برنامه مذکور نیازمند تأیید کاربر برای مجوز WRITE_EXTERNAL_STORAGE و ACCESS_NETWORK_STATEاست.

یک چالش کلیدی برای هر برنامه مخرب) ازجمله برنامه مذکور(جلوگیری از تشخیص توسط کاربران و فیلترهای نرم‌افزارهای مخرب است. الگوریتم‌های فیلتر مخرب به‌طورکلی بر اساس ارزیابی مجوز کاربران و وضعِ فعالیت است. این الگوریتم‌ها یک برنامه را در صورتی مشکوک در نظر می‌گیرند که بیش‌ازحد مجوزهایی یا مجوزهای مرتبط با فعالیت تحمیل هزینه به دست آورد، مانند ارسال پیامک یا MMS یا تماس گرفتن؛ به‌عنوان‌مثال، به شماره حق بیمه؛ بنابراین، برنامه مذکور برای استفاده تعداد کم، به‌طور مداوم و بدون هزینه تحمیل مجوز مربوطه برای جلوگیری از تشخیص طراحی شده است.

نکته قابل‌توجه در استفاده از تلفن همراه، توجه به مصرفِ زیاد پهنای باند شبکه نیز منجر به تشخیص و یا بالابردن سوءظن کاربر می‌شود؛ بنابراین، برای جلوگیری از استفاده بیش‌ازحد باتری، برنامه ما برای استفاده اندروید ساخته و طراحی شده است. به‌طور

مشابه برای جلوگیری از مصرفِ زیاد پهنای باند، برنامه ما اطلاعات کاربر را تنها زمانی که دستگاه به شبکه WiFi متصل است، به یک سرور شخص ثالث که تحت کنترل دشمن است، آپلود می‌کند. اگر این کاربر در حال مرور صفحات وب با استفاده از 3G/4Gباشد، برنامه ما داده‌های حافظه نهان مرورگر را بر روی پوشه خصوصی برنامه کپی می‌کند و منتظر می‌ماند تا دستگاه به WiFi متصل شود. به‌عبارت‌دیگر این برنامه در WiFi و شبکه 4 Gاجرا می‌شود.

فعالیت ۱:

با تعیینِ زمانی که مرورگر شروع می‌شود، اندروید روش‌های متعددی را برای بازگشتِ برنامه‌های در حال اجرا فراهم می‌کند. و یک روش مؤثر برای ایجاد یک گیرنده پخش برای برنامه مرورگرِ مورد هدف استفاده از (ActivityManager.getRunningAppProcess) است. با این حال، این ویژگی در اندروید API سطحِ ۲۱ قدیمی شده است؛ بنابراین، از ویژگیِ fileObserverبومی در اندروید استفاده خواهد شد. زمانی که در فایل مقصد یا دایرکتوری تغییراتی رخ دهد، به اطلاع برنامه می‌رساند. FileObserver از زیرسامانه inotify از هسته لینوکس استفاده می‌کند که fileSystems را برای حریق اطلاعات زمانی که یک دایرکتوریِ داده‌شده توسط این مرورگر در دسترس قرار گیرد، گسترش می‌دهد.

فعالیت۲:

کپی‌کردن از فایل‌های حافظه نهان: ازآنجاکه در اندروید فرمانِ «Cp» وجود ندارد، ما از InputStream و OutputStream بومی برای کپی‌کردن فایل‌ها از دایرکتوریِ حافظه نهان مرورگر استفاده می‌کنیم (در الگوریتم۲) سپس ما فایل‌هایی را که با استفاده از الگوریتم ۱کپی شده، شناسایی می‌کنیم. فایل‌های کپی شده در دایرکتوریِ حافظه نهانِ برنامه اولیه (در حافظه داخلی) برای جلوگیری از بالابردن سوءظن کاربر ذخیره خواهند شد. برای به کمینه رساندن استفاده از منابع، ما دایرکتوریِ حافظه نهانِ برنامه اولیه را به

5MBمحدود خواهیم کرد و سامانه به‌طور خودکار وقتی‌که این حد پر شود، فایل‌های قدیمی‌تر را پاک خواهد کرد.

الگوریتم ۱: فهرست کردن فایل‌های حافظه پنهان

```
File applicationCache=this.getCacherDir();
File browserCache=new File(Environment.getExternalStorage()get.absolutebath() +
"/cacheDirectoryName");
String[] filename = browserCacheList();
For (int i=0; i<browserCacheListFiles().length; i++){
copyFiles(new file(browserCache, filename[i]),
new File(applicationCache, filename[i]));
}
```

الگوریتم ۲: کپی‌کردن فایل‌های حافظه پنهان

```
copyFiles(File source, File dest){
InputStream toCopyFile = new FileInputStream(source);
OutputStream copiedFile = new FileOutputStream
(dest);
Byte[] buff = new byte[1024];
Int leng;
While ((leng = toCopyFile.read(buff)) > 0){
}
toCopyFile.close();
copiedFile.close();
}
```

برای بهبودِ بهره‌وری از برنامه نمونه اولیه، می‌توان توسط اجرای این دستور «این فایل از قبل موجود است» از کپی شدنِ همان فایل بیش از یک‌بار جلوگیری کرد.

فعالیت۳:

بررسی کردن اتصال شبکه و انتقال فایل: آپلود مکرر فایل‌های حافظه نهان از دستگاه‌های سرویس‌گیرنده می‌تواند سوءظن را با توجه به مصرف پهنای باند بالا ببرد

(مانند زمانی که کاربر یک برنامه داده محدود دارد)؛ بنابراین، برنامه نمونه اولیه برای بررسی اتصالِ WiFi طراحی شده که در الگوریتم ۳ نشان‌داده‌شده است.

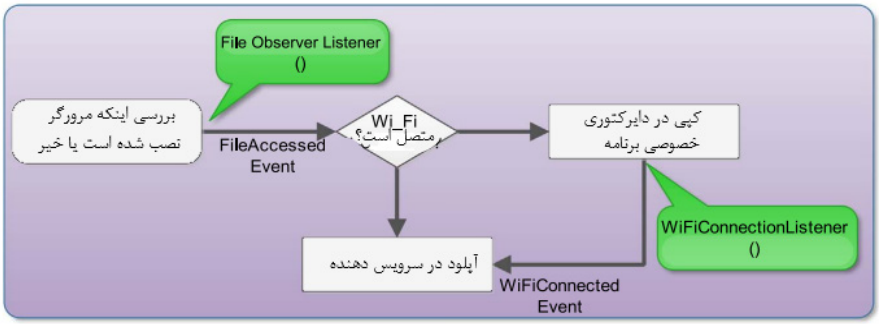
الگوریتم ۳: بررسی اتصال وای‌فای

```
ConnectivityManager manager = (ConnectivityManager) get-
SystemService (CONNECTIVITY_SERVICE);
NetworkInfo wifiConn = connManager.getNetworkInf
(ConnectivityManager.TYPE_WIFI);
return wifiConn.isAvailable();
```

اگر یک اتصالِ WiFi شناسایی شود، داده‌های ذخیره‌شده در حافظه خصوصیِ برنامه در دست اجرا بر روی سرور تعیین‌شده، آپلود خواهد شد.[9]

آپلودِ فایل‌ها بر روی سرور می‌تواند بر اساس اتصالِ بهWiFi ، یک فرآیند مصرف منابع و طولانی باشد. برنامه ما به عنوان یک سرویس در پس‌زمینه اجرا می‌شود، می‌توان برای آپلود فایل‌ها از «asyncTask» استفاده کرد. حلقه این برنامه تا زمانی که آپلود کامل شود، چندین مرتبه اجرا خواهد شد- در شکل۲. سپس فایل‌ها از حافظه خصوصی برنامه پاک خواهند شد.

شکل شماره (۲): آپلود فایل‌ها بر روی سرورِ تعیین‌شده



فعالیت۴:

درج تصویر در حافظه نهان (مسمومیت حافظه نهان): با استفاده از مجوزِ رایت ایکسترنال[1]، برنامه ما قادر به درج تصویر در دایرکتوریِ حافظه نهانِ مرورگر است- در

1. WRITE_EXTERNAL

الگوریتم ۴ و ۵. بارگذاریِ تصویر درج‌شده توسطِ مرورگر به تعدادی از شرایط بستگی دارد، مانند زمان انقضا، نوع فایل حافظه نهان و فرکانس دسترسی کاربر به URL مربوطه.

الگوریتم ۴: تعیین اینکه آیا فایل حافظه پنهان حاوی یک تصویر است یا خیر.

```
BitmapFactory.Options imageOptions = new BitmapFactory.
Option ();
imageOptions.inJustDecodeBounds = true;
BitmapFactory.decodeFile (file. getPath, imageOptions);
If (options.outWidth != -1 && options.outHeight!= -1)
return true;
```

الگوریتم ۵: جایگزینی فایل تصویر.

```
Byte [] buff = new byte [1024];
Int leng;
While (( leng = new FileInputStream(toBeCopied).read
(buff)) > 0) {
toBeReplaced.write(buff, 0, leng);
}
toBeCopied.close();
toBeReplaced.close();
```

راه‌اندازی آزمایش:

بنا به آمارِ گوگل‌پلی‌استور، مرورگرهایِ گوگل کروم، موزیلا فایرفاکس و مرورگرهای UC محبوب‌ترین مرورگرها در برنامه‌های اندروید هستند. کروم و فایرفاکس دارای سازگاریِ مرور کراس- پلتفرم و اجرای اقدامات امنیتی مختلف هستند.[10] با این حال، مرورگرهای سبک بین کاربرانی که سرعتِ بارگذاریِ منابعِ آن‌ها به‌طور زیادی کمتر است، محبوب هستند، به‌ویژه بارگذاریِ فایل‌های رسانه‌ای بزرگ و در محیط بازی؛ بنابراین، در این پژوهش، سه مرورگر سبک اندروید انتخاب شد: دلفین، مرورگرهای CM و مرورگرهای UC (از سپتامبر سال 2015، بر اساس دانلودهای گوگل‌پلی‌استور). ازآنجایی‌که سامسونگ یک الگوی محبوب اندروید است، مرورگر سهام سامسونگ نیز (نام بسته:.com.sec.android.app.sbrowser) مدنظر قرار داده‌شده که به‌طور پیش‌فرض بر روی سامسونگِ Galaxy S4 با نسخه اندروید 2.4.4 نصب‌شده است.[11]

جدول ۱ اطلاعاتی در موردِ سه مرورگرِ انتخاب شده به ما نشان می‌دهد. ما در حال حاضر به‌طور خلاصه توضیحی در موردِ وضعیتِ ذخیره‌سازیِ حافظه نهانِ مرورگرهای انتخاب شده ارائه می‌دهد.

مرورگر UC یکی از محبوب‌ترین مرورگرهای اندروید است که بیش از 100 میلیون بار دانلود شده است. پس از بازرسی اولیه، کشف شد که مرورگرِUC ، بخشِ بزرگی از فایل‌های حافظه نهان را در حافظه به اشتراک‌گذاری شده خارجی ذخیره می‌کند که شاملِ فایل‌هایHTML از صفحه‌های وبِ دیده‌شده،JavaScript ،Css، فایل‌های تصویری و طیف وسیعی از پایگاه داده‌های پرشده توسطِ فعالیت مرور کاربران است. این پایگاه داده‌ها تنها اطلاعات مربوط به صفحه‌های وبِ مرور شده توسط کاربران را نشان نمی‌دهند، بلکه تعداد دفعاتی که دیده‌شده‌اند و زمان دسترسی و اینکه هر بار چه پاسخی از سرویس‌دهنده دریافت کرده‌اند را هم نشان می‌دهند. این مسئله که چه مقدار داده (پهنای باند) را هنگام دسترسی به هر صفحه به‌طور جداگانه استفاده کرده‌اند هم نشان می‌دهند.[12]

مرورگر دلفین یکی از قدیمی‌ترین مرورگرهای تعریف شده برای اندروید است. بازرسی اولیه حافظه نهان آن نشان می‌دهد که مرورگر دلفین اکثر منابع حافظه نهان را در حافظه به اشتراک‌گذاری شده خارجی درdirectory /sdcard/TunnyBrowser/cache/ ذخیره می‌کند. این مرورگر سه زیر-دایرکتوری دارد:speeddial_covers ، tablist_cache و webviewCache. همان‌طور که از نام آن مشخص است Speed dial دسترسیِ سریع (به کاربر اجازه می‌دهد تا یوآرال‌های موردعلاقه و یوآرال‌هایی که بیش‌تر استفاده می‌کند را بر روی عکس‌های ریزِ ((thumbnails جلوی مرورگر ذخیره کند. دایرکتوریِ Speeddial_covers، URL و فایل‌های مربوط به آن وب‌سایت را ذخیره می‌کند.[13]

در ادامه بررسی مشخص شد که دلفین، تصاویرِ محتوای وبِ نمایش داده‌شده را ذخیره کرده است. به‌عنوان‌مثال، هنگامی‌که یک کاربر به‌طور هم‌زمان بیش از یک صفحه را مرور می‌کند، محتوای آن صفحه به عنوان یک تصویر قابل‌مشاهده خواهد بود که حجم آن در حدود پنجاه کیلوبایت خواهد بود و در دایرکتوریِ tablist_cache ذخیره خواهد شد.

تمامِ دیگر فایل‌های حافظه نهان که شاملِ HTML ، Css و فایل‌های رسانه‌ای می‌شود در دایرکتوریِ webViewCache ذخیره می‌شود. [14]

بازرسی اولیه نشان داده که مرورگرCM ، اکثر فایل‌های حافظه نهان را در حافظه داخلی ذخیره می‌کند و مشخص شد که ذخیره‌سازیِ محلی در دایرکتوریِapp_webview اجرا می‌شود، مشابهِ ذخیره‌سازیِ حافظه نهانِ استانداردِ ارائه شده در اچ‌تی‌ام‌ال5. با این وجود، یوآرال‌های دیده‌شده در فایلی در حافظه به اشتراک‌گذاری شده، ذخیره و فهرست می‌شوند.[15]

در موردِ مرورگر سهام سامسونگ، فایل‌های حافظه نهان، شاملِ HTML، javaScript و فایل‌های رسانه‌ای می‌شود که در حافظه داخلی ذخیره‌شده‌اند. مشابهِ مرورگرِ دلفین، همچنین هر زمان که محتوای جدید در مرورگر بارگذاری شود، تصاویر را به عنوانِ فایلِ bitmapذخیره می‌کند. این بیت‌مپ[1] زمانی که کاربر به کلید سوئیچ برنامه سریع در دستگاه دسترسی دارد یا زمانی که کاربر اقدام به تغییرِ صفحه در مرورگر می‌کند، ممکن است برای نمایشِ پیش‌نمایش‌های برنامه‌های کاربردیِ در حال اجرا استفاده شود.

به‌طور پیش‌فرض، فایل‌های ذخیره‌شده در حافظه داخلی، خصوصی و تنها قابل‌دسترسی توسط صاحب برنامه در نظر گرفته می‌شوند. با این وجود، این مجوز می‌تواند در هنگام ایجاد یک فایل با استفاده از پرچم‌هایِMODE_APPEND ،MODE_ WORLD_READABLEو یا MODE_WORLD_WRITABLE توسط خالقِ برنامه تغییر کند. این مسئله به عنوانِ «نقص امنیتی جدی» برای اندروید در نظر گرفته می‌شود و از اندروید API در سطحِ ۱۷ حذف شده است. با این حال، یک نقص در مرورگر سهام سامسونگ یافت شده است که توسط این تحقیق نشان داده می‌شود. به‌طور خاص، مشخص شد که تصاویر ذخیره‌شده توسط مرورگر سهام سامسونگ در حافظه نهانِ آن با سطحِ دسترسیِ ۶۴۴ (rw-r-r-) مجوز کاربر اختصاص داده‌شده است.، بدین معنی است که می‌تواند توسط هر برنامه دیگری مورد دسترسی قرار بگیرد.

در این پژوهش، محققان از سامسونگِ Galaxy S4 (روت شده) در حال اجرا برای اندروید با نسخه 2.4.4 به عنوان دستگاهِ اصلیِ آزمایش استفاده شد. سپس برنامه نمونه

1. Bitmap

در بخش بعدی بر روی دستگاه نصب که در فرایند پس‌زمینه اجرا می‌شود. همچنین هر مرورگر برای پنج دقیقه در یوآرال‌های بازشده در چند صفحه اجرا شد. برای بررسی اثرِ الگوی دشمنِ ما، همین آزمایش روی سامسونگِ Galaxy S5 (روت شده) با اندروید نسخه .0.5 نیز اجرا شده است.[16]

یافته‌ها

در این بخش یافته‌های تحقیق توضیح داده می‌شود که شامل:

دلفین: محققین در این زمینه موفق به کپی‌کردن تمام فایل‌ها در دایرکتوریِ ریشه حافظه نهان (/sdcard/TunnyBrowser/Cache) و در زیرشاخه‌های آن شده‌اند. همچنین فایل‌ها با موفقیت در سرویس‌گیرنده آپلود شده‌اند.

در بررسیِ فایل‌های آپلود شده، با مشاهده محتوای فایل‌های رسانه‌ای؛ صفحه‌های وب دیده‌شده و غیره، قادر به بازیابیِ تاریخچه یوآرال مرورگر شد. همچنین با دسترسی به لینک در فایلِ اچ‌تی‌ام‌ال قادر به بازیابیِ محتوای خصوصیِ صفحه وب شد، مانند تصاویر و پست‌های دیوارِ صفحه فیس‌بوک کاربر. از تصاویر آپلود شده، ایمیل‌ها، جزئیات حساب کاربری و دیگر اطلاعات حساس کاربر به دست آمد. همچنین مشخص شد که دلفین، تصاویرِ حافظه نهان را بدون تغییرِ هدر ذخیره می‌کند. به‌طورکلی، زمانی که کاربر تلاش می‌کند تا به همان یوآرال دسترسی پیدا کند، مرورگر توسط حافظه نهانِ سرویس‌دهنده بررسی می‌شود که مشخص شود آیا حافظه نهانِ ذخیره‌شده تغییر یافته است یا خیر! با این وجود، دلفین قادر به شناساییِ حافظه نهانِ تغییر یافته در این پژوهش نیست. به‌طور خاص، صفحه اصلیِ آمازون (https://www. amazon.com/) مرور شد و مشخص گردید که تمام تصاویر محصولات در حافظه نهان ذخیره می‌شوند. [16]. پس‌ازآن تعدادی از این تصاویر با تصاویر دیگر با همان نامِ فایل جایگزین شد. بلافاصله همان صفحه دوباره مرور شد و با تصاویر جایگزین شده (به‌جای تصاویر اصلی) ارائه شد. همچنین به این مسئله که زمان بین مرور اول و دوم کمتر از دو دقیقه بود توجه شد و یافته‌های به‌دست‌آمده برای دو دستگاهِ دیگر هم صادق بود- سامسونگِ Galaxy S4 و سامسونگِ Galaxy S5 .

مرورگرِ UC:

مشابهِ یافته‌های توضیح داده‌شده در بخشِ قبل، موفق به کپی‌کردن و آپلودکردنِ داده‌های ذخیره‌شده در دایرکتوریِ ریشه حافظه نهان شده است. کاربر قادر به شناسایی تصاویر نیست، اگرچه بر اساس اطلاعات جمع‌آوری‌شده از دیگر فایل‌های حافظه نهان و پایگاه داده، در دوره نقاهت تاریخچه مرورگرِ وب و شناسایی وضعیت مرور کاربر موفق بوده است. برای مثال، قادر به تعیین این بوده که صفحه وب در چه زمانی و توسط چند نفر دیده‌شده و قادر به بازیابیِ محتوای صفحه وبِ دیده‌شده هست. نتایجِ مشابهی برای هر دو دستگاه به دست آمد .[17]

مرورگر سهام سامسونگ:

همان‌طور که قبلاً توضیح داده شد، مرورگر سهام سامسونگ تمام منابعِ حافظه نهان را در حافظه داخلی دستگاه ذخیره می‌کند. با این وجود، در سامسونگِ Galaxy S4 ، قادر به کپی‌کردن و آپلودکردن تصاویر از دستگاه شده است، یک‌بار هم محتوای جدیدی در مرورگر بارگذاری شد. از طریق این تصاویر، کاربر قادر به شناساییِ فعالیت‌های محتوای وب‌گاه‌ها هست. همچنین قادر به بازیابی اطلاعات دیگری مانندِ پیام‌های رایانامه و جزئیات حساب کاربری هست. محتوای ذخیره‌شده با استفاده از مجوزِ دسترسیِ ۶۴۴، ما را از هرگونه اصلاح بازمی‌دارد.

با این وجود دریافت شد که این مرورگر بر روی سامسونگِ galaxy S5 با اندروید نسخه 0.5 به‌صورت پیش‌فرض نصب‌شده است. با این وجود، بازرسی ما از دیگر موبایل‌ها و تبلت‌های سامسونگ نشان‌دهنده این است که این مرورگر بر روی دستگاه‌های اندروید با نسخه2.4.4 نصب شده است. [18]

مرورگرِ سی‌ام[1]:

با کمال تعجب، مرورگر سی‌ام تمام فایل‌های حافظه نهان، ازجمله تصاویر برای جهت‌یابیِ چندصفحه‌ای، پایگاه داده، فایل‌های اچ‌تی‌ام‌ال و فایل‌های رسانه‌ای بر روی حافظه داخلی دستگاه در حالت خصوصی ذخیره می‌کند. تنها اطلاعاتی که می‌توان بازیابی کرد، فهرست یوآرال‌های دیده‌شده توسط کاربر است که به عنوان یک فایل در/

1. CM

/sdcard/CheeahBrowser/.data موجود است. مشابهِ یافته‌های گزارش‌شده در موردِ دو دستگاه دیگر.

روش تحقیق و تحلیل داده‌ها

جامعه آماری این پژوهش شامل مرورگرهای سبک تلفن همراه مبتنی بر سامانه عامل اندروید است که به‌طور گسترده برای دسترسی به سرویس‌های وب و ابری مورداستفاده قرار می‌گیرند. با توجه به هدف پژوهش که بررسی امنیت ذخیره‌سازی داده‌ها در مرورگرهای سبک است، از روش نمونه‌گیری هدفمند (Purposeful Sampling) استفاده شد.

بر این اساس، چهار مرورگر UC Browser، Dolphin Browser، CM Browser و مرورگر پیش‌فرض سامسونگ انتخاب شدند. معیارهای انتخاب شامل:

1. تعداد بالای کاربران و دانلود در Google Play،
2. استفاده گسترده از حافظه نهان برای افزایش سرعت،
3. تفاوت در معماری ذخیره‌سازی داده‌ها،
4. دسترس‌پذیری برای تحلیل آزمایشگاهی بودند.

آزمایش‌ها بر روی دو دستگاه Samsung Galaxy S4 (Android 4.4.2) و Samsung Galaxy S5 (Android 5.0.2) انجام شد تا اثر نسخه سامانه عامل بر رفتار ذخیره‌سازی مرورگرها نیز بررسی گردد.

روش تحلیل داده‌ها در این پژوهش تحلیل کیفی-تجربی[1] است. داده‌های استخراج‌شده از حافظه نهان مرورگرها شامل فایل‌های HTML، CSS، JavaScript، تصاویر و پایگاه‌های داده موردبررسی قرار گرفتند.

تحلیل داده‌ها در سه سطح انجام شد:

۱. تحلیل محل ذخیره‌سازی داده‌ها (حافظه داخلی یا خارجی،
۲. تحلیل سطح دسترسی فایل‌ها و امکان دسترسی غیرمجاز،
۳. تحلیل قابلیت بازیابی و دست‌کاری داده‌ها توسط مهاجم.

1. Experimental Qualitative Analysis

نتایج به‌صورت مقایسه‌ای میان مرورگرها تحلیل و در قالب جدول‌ها و نمودارها ارائه شد.

این پژوهش به روش آزمایشگاهی و تحلیلی انجام شده است. برای بررسی امنیت مرورگرهای سبک تلفن همراه، چهار مرورگر پرکاربرد شامل UC Browser، Dolphin، CM Browser و مرورگر پیش‌فرض سامسونگ انتخاب شده است. مرورگرها بر روی دو دستگاه تلفن همراه سامسونگ Galaxy S4 و Galaxy S5 با نسخه‌های مختلف سامانه عامل اندروید نصب شدند.

برای شناسایی آسیب‌پذیری‌ها، یک برنامه نمونه اندروید طراحی شد که با استفاده از مجوزهای READ_EXTERNAL_STORAGE و WRITE_EXTERNAL_STORAGE قادر به خواندن و نوشتن فایل‌های ذخیره‌شده در کش مرورگرها بود. این برنامه همچنین برای انتقال داده‌ها به یک سرور آزمایشی از مجوز اینترنت بهره گرفت.

فرآیند تحقیق شامل مراحل زیر بود:

۱. اجرای مرورگرها و مرور وب‌سایت‌های مختلف برای تولید داده در کش.
۲. شناسایی مسیرهای ذخیره‌سازی کش در هر مرورگر (حافظه داخلی یا خارجی).
۳. کپی‌برداری از فایل‌های کش شامل HTML، CSS، JavaScript، تصاویر و پایگاه‌های داده.
۴. تحلیل داده‌های استخراج‌شده به‌منظور بازیابی تاریخچه مرورگر، محتوای صفحات وب و اطلاعات حساس کاربر.
۵. مقایسه مرورگرها بر اساس میزان آسیب‌پذیری، سطح دسترسی فایل‌ها و امکان سوءاستفاده مهاجم.

فرضیه اصلی مقاله هم عبارت است از: مرورگرهای سبک تلفن همراه به‌دلیل ذخیره‌سازی ناامن داده‌ها در حافظه نهان، در معرض تهدیدات امنیتی جدی قرار دارند.

برای آزمون این فرضیه، رفتار ذخیره‌سازی داده‌ها در چهار مرورگر منتخب موردبررسی قرار گرفت. نتایج نشان داد:

- در مرورگرهای Dolphin و UC Browser داده‌های حساس در حافظه خارجی و بدون رمزنگاری ذخیره می‌شوند که امکان بازیابی تاریخچه مرور، محتوای صفحات وب و تغییر فایل‌های کش را فراهم می‌کند.
- در مرورگرSamsung Stock Browser ، اگرچه داده‌ها در حافظه داخلی ذخیره می‌شوند، اما تنظیمات نادرست سطح دسترسی فایل‌ها (Permission 644) منجر به دسترسی غیرمجاز می‌شود.
- مرورگرCM Browser نسبت به سایر مرورگرها امنیت بالاتری دارد، اما همچنان امکان افشای تاریخچه مرور وجود دارد.

مبانی نظری و چارچوب مفهومی

مرورگرهای سبک تلفن همراه برای بهبود سرعت بارگذاری صفحات وب، از روش‌هایی مانند افزایش اندازه حافظه نهان[1] و کاهش وابستگی به افزونه‌ها استفاده می‌کنند. هرچند این اقدامات تجربه کاربری را بهبود می‌بخشد، اما از نظر امنیتی می‌تواند آسیب‌پذیری‌هایی ایجاد نماید. در این بخش به برخی مفاهیم کلیدی مرتبط با پژوهش پرداخته می‌شود:

ذخیره‌سازی وب[2]: ذخیره‌سازی وب به‌عنوان بخشی از استاندارد HTML5 معرفی شد و شامل دو نوع ذخیره‌سازی محلی[3] و ذخیره‌سازی جلسه‌ای[4] است. این سازوکار درصورتی‌که به‌درستی پیاده‌سازی شود می‌تواند برای ذخیره داده‌های حساس کاربران مورداستفاده قرار گیرد. در دستگاه‌های اندرویدی معمولاً ذخیره‌سازی وب در حافظه داخلی انجام می‌شود که امنیت بیشتری دارد.

حافظه نهان مرورگر[5]: حافظه نهان، داده‌های صفحات وبHTML، CSS، JavaScript و فایل‌های رسانه‌ای را موقتاً ذخیره می‌کند تا در بازدیدهای بعدی سرعت بارگذاری افزایش یابد. با این حال، در مرورگرهای سبک به‌دلیل استفاده از حافظه خارجی، نظیر کارتSD ، این داده‌ها ممکن است در معرض دسترسی سایر برنامه‌ها قرار گیرند.

1. Cache
2. Web Storage
3. Local Storage
4. Session Storage
5. Browser Cache

الگوی تهدید[1]: الگوی تهدید چارچوبی است برای شناسایی مسیرهای حمله و آسیب‌پذیری‌ها. در پژوهش حاضر، الگوی دشمنی طراحی شد که قادر است تاریخچه مرورگر، عبارات جستجو، محتوای وب و فایل‌های تصویری ذخیره‌شده در کش را استخراج و حتی محتوای آن را تغییر دهد.

نشت داده[2]: ذخیره‌سازی ناامن داده‌ها در مکان‌های عمومی می‌تواند منجر به نشت اطلاعات حساس کاربران شود. برای مثال، ذخیره تصاویر، تاریخچه مرورگر یا فایل‌های HTMLدر حافظه خارجی باعث می‌شود مهاجمان با دسترسی ساده به مجوزهای اندروید بتوانند به این داده‌ها دسترسی پیدا کنند.

حملات مبتنی بر کش[3]: مهاجمان می‌توانند از فایل‌های کش مرورگر برای اجرای حملاتی مانند فیشینگ یا دست‌کاری محتوای صفحات وب استفاده کنند. این حملات نه‌تنها امنیت کاربران را تهدید می‌کنند، بلکه اعتبار سرویس‌های آنلاین نیز در معرض خطر قرارمی‌گیرد.

بنابراین، مرورگرهای سبک علی‌رغم کارایی بالا، به‌دلیل استفاده از روش‌های ذخیره‌سازی ناامن، در معرض تهدیدهای امنیتی متعددی قرار دارند. این چارچوب نظری مبنای طراحی الگوی تهدید در پژوهش حاضر را تشکیل می‌دهد.

تجزیه‌وتحلیل یافته‌ها در ارتباط با فرضیه پژوهش

با توجه به اینکه مقاله در نسخه اولیه فاقد فرضیه صریح بود، در نسخه اصلاح‌شده فرضیه زیر به‌عنوان مبنای تحلیل در نظر گرفته می‌شود:

فرضیه اصلی: مرورگرهای سبک تلفن همراه به‌دلیل شیوه ذخیره‌سازی ناامن داده‌ها، در معرض تهدیدات امنیتی جدی قرار دارند. بر اساس این فرضیه، داده‌های حاصل از آزمایش‌ها تحلیل شدند:

مرورگر دلفین[4]: نتایج نشان داد تمامی فایل‌های کش در حافظه خارجی ذخیره می‌شوند. این موضوع امکان بازیابی تاریخچه مرورگر، مشاهده محتوای صفحات خصوصی (مانند

1. Threat Model
2. Data Leakage
3. Cache-based Attacks
4. Dolphin

ایمیل‌ها و پیام‌های شبکه‌های اجتماعی) و حتی جایگزینی تصاویر اصلی با فایل‌های جعلی را فراهم ساخت؛ بنابراین فرضیه اصلی در مورد ناامنی این مرورگر تأیید می‌شود.

مرورگر UC Browser: بررسی‌ها نشان داد این مرورگر علاوه بر ذخیره فایل‌ها، پایگاه‌های داده‌ای شامل اطلاعات دقیق زمان‌بندی و پاسخ‌های سرور را در حافظه خارجی ذخیره می‌کند. این اطلاعات بدون رمزنگاری بوده و توسط برنامه آزمایشی قابل‌استخراج بود. یافته‌ها فرضیه پژوهش را در مورد آسیب‌پذیری این مرورگر تأیید می‌کند.

مرورگر سهام سامسونگ[1]: این مرورگر اگرچه داده‌ها را در حافظه داخلی ذخیره می‌کند، اما تصاویر با سطح دسترسی ضعیف (۶۴۴) ذخیره‌شده‌اند. این ضعف امنیتی به مهاجم اجازه دسترسی غیرمجاز به تصاویر و محتوای مرور شده کاربر را می‌دهد؛ بنابراین، این مرورگر نیز با فرضیه پژوهش همسو است.

مرورگر CM Browser: این مرورگر اکثر داده‌های کش را در حافظه داخلی و با سطح دسترسی مناسب ذخیره کرده است. تنها اطلاعات قابل بازیابی، فهرست URL های مشاهده‌شده بود؛ بنابراین، در مقایسه با مرورگرهای دیگر، امنیت نسبی بیشتری دارد، اما همچنان در برابر افشای تاریخچه مرور آسیب‌پذیر است.

جمع‌بندی:

تحلیل نتایج نشان داد مرورگرهای سبک به‌طورکلی در ذخیره‌سازی داده‌های کاربران امنیت کافی ندارند. مواردی همچون ذخیره داده‌ها در حافظه خارجی بدون رمزنگاری، مجوزهای دسترسی ضعیف و امکان دست‌کاری کش، تأییدی بر فرضیه اصلی پژوهش هستند.

در پاسخ به پرسش اصلی پژوهش، چنین تعریف شده است: آیا مرورگرهای سبک تلفن همراه قادر به ذخیره‌سازی امن داده‌های کاربران هستند؟ با توجه به آزمایش‌ها و تحلیل‌های انجام‌شده، پاسخ این پرسش به‌طور روشن منفی است. یافته‌های پژوهش نشان‌دادند:

۱. مرورگرهای دلفین و UC داده‌ها را در حافظه خارجی و بدون رمزنگاری ذخیره می‌کنند و امکان بازیابی تاریخچه مرور، محتوای صفحات خصوصی و حتی تغییر فایل‌های کش وجود دارد.

1. Samsung Stock Browser

۲. مرورگر سهام سامسونگ اگرچه از حافظه داخلی استفاده می‌کند، اما به علت تنظیمات نادرست سطح دسترسی فایل‌ها، تصاویر ذخیره‌شده در معرض دسترسی برنامه‌های غیرمجاز هستند.

۳. مرورگر CM نسبت به دیگر مرورگرها امنیت بهتری دارد، زیرا داده‌ها را عمدتاً در حافظه داخلی ذخیره می‌کند، با این حال در برابر افشای تاریخچه مرور همچنان آسیب‌پذیر است.

بنابراین، بر اساس شواهد به‌دست‌آمده، می‌توان نتیجه گرفت که مرورگرهای سبک تلفن همراه هنوز فاقد سازوکارهای کافی برای حفاظت از داده‌های حساس کاربران هستند و فرضیه پژوهش مبنی بر «وجود تهدیدات امنیتی ناشی از شیوه ذخیره‌سازی داده‌ها» به‌طور کامل تأیید می‌گردد.

خلاصه

جدول شماره (۳) یافته‌های پژوهش در مورد چهار مرورگر را به‌طور مختصر بیان کرده است.

جدول شماره (۴) پیاده‌سازیِ امنیتیِ (مورداستفاده در پژوهشِ) مرورگرهای سبکِ موردمطالعه در این مقاله و همچنین دو مرورگر محبوب دیگر را به‌طور مختصر بیان کرده است. این مسئله روشن است که دلفین، مرورگرِ یوسی[1] و مرورگرهای سهام سامسونگ فاقد سازوکارهای امنیتی اساسی هستند.

جدول شماره۳ - خلاصه از یافته‌ها.

مرورگرها	اهداف دشمن			
	تاریخچه مرورگر	شرایط جستجو	عصاره (دانش) محتوای وب	تغییر محتوای حافظه نهان
۱.مرورگر دلفین	بله	بله	بله	بله
۲.مرورگر CM	بله			
۳.مرورگر UC	بله	بله	بله	
مرورگرهای سهام سامسونگ	بله		بله	

1. UC

جدول شماره۴ - خلاصه مقایسه‌ای از پیاده‌سازیِ شاخص‌های امنیتی در مرورگرهای تلفن همراه.

سلسله مرورگر[۳]			
مرورگرها	فیلترضدفیشینگ	نمایش هویت ارائه‌دهندگان	نمایش گواهی نامه
گوگل کروم	بله	بله	بله
موزیلافایرفاکس	بله	بله	بله
مرورگرهای سبک (نتیجه ارزیابی ما)			
دلفین	نه	بله	بله
مرورگرUC	نه	نه	نه
مرورگرهای سهام سامسونگ	نه	بله	بله
مرورگرCM	بله	بله	بله

امنیت ذخیره‌سازی در اندروید نیز اخیراً موردتوجه محققین بوده است. برای مثال، در سال ۲۰۱۵، لی‌یوو همکارانش بر روی رفتارهای ذخیره‌سازی داده‌های برنامه اندروید مطالعه کرده‌اند. آن‌ها داده‌های ذخیره‌شده توسط برنامه‌های ارتباطیِ محبوب را موردبررسی قراردادند. مانند ویبو، فیسبوک، اینستاگرام، لاین، اسکایپ و وایبر. برای اینکه تعیین کنند چه مقدار از داده‌های خصوصی کاربران از این برنامه‌ها بدون آگاهیِ کاربران توسط مهاجم قابل سرقت و نفوذ است.[5] این مطالعه همچنین نشان می‌دهد که امنیت اطلاعات خصوصی کاربران وابسته به این است که آیا طراحِ برنامه این اطلاعات را حساس یا غیر حساس تعیین کرده است. داده‌هایی توسط طراحِ برنامه غیر حساس در نظر گرفته خواهد شد که برای اشتراک در حافظه ذخیره‌سازیِ خارجی ذخیره‌شده باشد که در دسترس دیگر کاربران هم قرارمی‌گیرد. با این حال، هیچ تعریف یکسانی برای داده‌های حساس و غیرحساس وجود ندارد. این مسئله نشان‌دهنده این است که در بعضی موارد، شماره تلفن، فهرست تماس و دیگر اطلاعات خصوصیِ کاربر به‌دست‌آمده از فایل‌های عمومی به اشتراک گذاشته‌شده توسط کاربر است.

مطالعات انجام شده توسط ژانگ و همکارانش نیز بقایای به‌جامانده از داده توسط برنامه‌های اندروید پس از حذف آن‌ها را گزارش می‌دهد که می‌تواند اطلاعات حساسی را به مهاجم نشان دهد. آن‌ها در آزمایش‌های خود، بقایای بهبود داده از سرویس‌های

سامانه و سرویس برنامه سامانه را موردبررسی قراردادند.[7] آن‌ها قادر به بازیابیِ اطلاعات حساسی مانند اعتبار ورودی کاربران، کلیدهای عمومی یا خصوصی، URL و محتوای قصد انتظار شدند. یافته‌های مشابهی از تجزیه‌وتحلیل برنامه‌های تلفن همراه نیز گزارش‌شده است.

لی‌یو و همکاران نشان‌دادند که چگونه یک مهاجم با قابلیت نصب یک برنامه با مجوزِ WRITE_EXTERNAL_STORAGE, READ_EXTERNAL_STORAGE and INTERNET می‌تواند دسترسی غیرمجاز به فایل‌های ذخیره‌شده عمومی در دستگاه‌های اندروید پیدا کند. [8] مشابه به روش لی‌یو و همکارانش ژوآو همکارانش از یک الگوی دشمن با قابلیت‌های یک مهاجم برای دزدیدن و ارسال داده از طریق مرورگر با استفاده از قصدِ URL ACTION-VIEW استفاده کردند. اما این روش (الگوی دشمن مجوز صفر) تنها زمانی می‌تواند داده را انتقال دهد که صفحه نمایش دستگاه در حال اجرا باشد. استفاده از الگوی دشمن در امنیت برنامه تلفن همراه نیز در این مطالعه یافت می‌شود که توسط مارتین و چو انجام شده است.

نتیجه تحقیق و پیشنهادها

الف-نتیجه تحقیق

نتایج نشان داد مرورگرهای سبک تلفن همراه عموماً از سازوکارهای ایمن و استاندارد برای ذخیره‌سازی داده‌های حساس کاربران استفاده نمی‌کنند. آسیب‌پذیری‌های شناسایی‌شده شامل ذخیره‌سازی داده‌ها در حافظه خارجی، ضعف در سطح دسترسی فایل‌ها و امکان دست‌کاری محتوای کش است. همچنین، الگوی تهدید پیشنهادی توانست این آسیب‌پذیری‌ها را به‌طور مؤثر شناسایی و ارزیابی نماید؛ بنابراین، شواهد تجربی به‌دست‌آمده فرضیه اصلی پژوهش را تأیید می‌کنند.

نتایج حاصل از این آزمایش‌ها در قالب جدول‌ها و توصیفات تفصیلی ارائه شد. همچنین در تحلیل داده‌ها نشان داده شد که برخی مرورگرها اطلاعات را در حافظه خارجی و بدون رمزنگاری ذخیره می‌کنند که زمینه نشت اطلاعات و سوءاستفاده مهاجمان را فراهم می‌سازد.

تعدادی از برنامه‌های تجاری برای پیگیری و نظارت بر فعالیت کاربران بر روی دستگاه‌های اندروید طراحی شده، مانند FlexySpy و .MobileSpy این برنامه‌ها به کاربر اجازه می‌دهند تا تاریخچه مرورگر را بخواند و بر روی دستگاه‌های بدون ریشه علامت‌گذاری کند، اما آن‌ها نیاز به مالک دستگاه تلفن همراه و اجازه دسترسی به برنامه را دارند، مانند مجوز مرورگرِ "com.android. . READ_HISTORY_BOOKMARKS"" برای دسترسی به تاریخچه مرورگر و "android.permission.ACCESS_FINE_LOCATION"یا " android.permission.ACCESS_COARSE_LOCATION "برای ردیابی محل کاربر استفاده می‌شوند.

این برنامه قادر به گرفتن تصاویری از فعالیت‌های مرورگر و یا دسترسی به محتوا نیستند (ایمیل و جزئیات حساب شخصی) مگر اینکه دستگاه روت شده باشد و به آن مجوزهای خاص اضافی اعطا شده باشد.[19]

نتایج حاکی از این است که؛ برنامه نمونه می‌تواند تاریخچه و محتوای مرورگرهای مقصد را بخواند (از طریق حافظه نهان) و همچنین از مرورگر سهام سامسونگ تصویربرداری کند. در مرورگر دلفین، برنامه همچنین می‌تواند فایل‌های تصویری در حافظه نهان را جایگزین کند و تصاویر جایگزین شده را نمایش دهد (به‌جای فایل‌های وب قانونی).

از آسیب‌پذیری‌های شناسایی‌شده در این مقاله، می‌توان آسیب‌پذیری توسط پروژه امنیتی برنامه وبِ باز[1] را به عنوان آسیب‌پذیریِ اصلی برنامه‌های تلفن همراه در نظر گرفت. شرح این مورد در جدول ۵ ارائه شده است.

جدول شماره (۵) - شناسایی آسیب‌پذیریِ امنیت ده برنامه تلفن همراه توسط OWASP

آسیب‌پذیریِ امنیت ده برنامه تلفن همراه OWASP	شناسایی آسیب‌پذیری در مرورگرها			
	دلفین	UC	CM	سهام سامسونگ
کنترل سمت سرویس دهنده ضعیف ذخیره‌سازیِ داده ناامن	بله	بله	بله	
حفاظتِ ناکافیِ لایه انتقال نشت داده به‌طور ناخواسته	بله	بله		بله

1. OWASP

آسیب‌پذیریِ امنیت ده برنامه تلفن همراه OWASP	شناسایی آسیب‌پذیری در مرورگرها			
	دلفین	UC	CM	سهام سامسونگ
احراز هویت و مجوز ضعیف				
رمزنگاری شکسته تزریق سمت سرویس‌گیرنده	بله			
تصمیم‌گیری‌های امنیتی از طریقِ ورودیِ غیرقابل اطمینان				
اداره جلسه نامناسب				
حمایتِ فاقد دودویی				

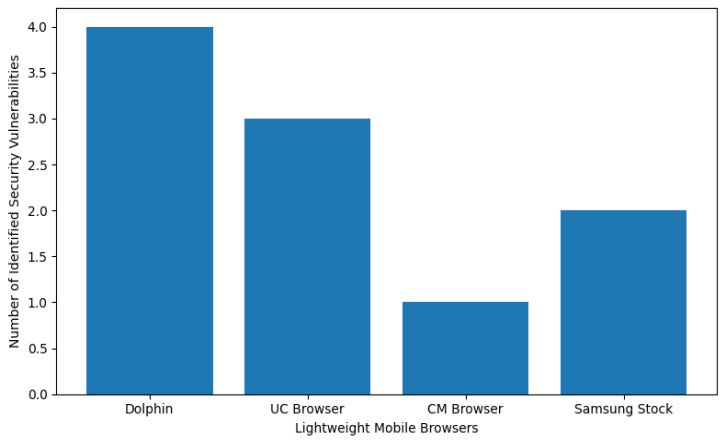
ذخیره‌سازیِ ناامن داده‌ها شاملِ عمل ذخیره‌سازی اطلاعات حساس بدون رمزنگاری و یا حفاظت از اطلاعات در مکانی ناامن است. برای مثال، در این تحقیق مشخص شد که مرورگر سهام سامسونگ تصاویر را در حافظه داخلی با یک مجوز فایلِ ضعیف ذخیره می‌کند. برنامه نمونه ما قادر به تزریقِ محتوای مخرب در حافظه نهانِ مرورگر دلفین است؛ بنابراین، این مرورگر به یک حمله تزریقی از سمت سرویس‌گیرنده آسیب‌پذیر است.

نشت اطلاعات به‌طور ناخواسته می‌تواند نتیجه ذخیره اطلاعات در مکانی ناامن باشد (به‌عنوان‌مثال داده ذخیره‌شده می‌تواند توسط شخص یا برنامه غیرمجاز مورد دسترسی قرار گیرد). برای مثال، این چهار مرورگرِ موردبررسی در این مقاله، داده را در حافظه به اشتراک‌گذاری شده خارجی ذخیره می‌کنند که می‌تواند توسط برنامه‌های دیگر مورد دسترسی قرار گیرد.

مورد پایانی؛ در این مقاله یک الگوی دشمن ارائه شد که می‌تواند در مطالعه امنیت مرورگرهای سبک استفاده شود که یک روش محبوب از دسترسی به سرویس‌های ابری است. برای نشان‌دادن عملیِ این الگوی دشمن، یک برنامه نمونه ساخته شد. با استفاده از چهار مرورگرِ سبک محبوبِ موردمطالعه، مشخص شد که دلفین، مرورگرِ CMو مرورگرِ UC اطلاعات کاربری حساس را در حافظه خارجیِ اشتراک‌گذاری شده ذخیره می‌کنند. اگرچه مرورگرِ سهام سامسونگ اطلاعات کاربری حساس را در حافظه داخلی دستگاه ذخیره می‌کند. نشان داده شد که یک مهاجم می‌تواند از مجوزهای فایل ضعیف برای دسترسی به اطلاعات ذخیره‌شده استفاده کند.[20] این آسیب‌پذیری به

علتِ یک نقص طراحی است؛ ذخیره‌سازی فایلِ نادرست توسط مرورگرها. این مسئله همچنین نیاز به اطمینان برای ذخیره‌سازی فایل‌های اندروید و سامانه مجوز فایل برای بهبود و جلوگیری از دسترسیِ افراد غیرمجاز به فایل‌های ایجاد شده توسط برنامه‌های دیگر تقویت می‌کند. از دیدگاهِ اجرایی، بارگذاری محتوا از حافظه داخلی بدون به خطر انداختن امنیت، بسیار کارآمدتر است. با این وجود، به نظر می‌رسد که مرورگرها هنوز هم برای استفاده از حافظه خارجی طراحی‌شده‌اند تا استفاده از حافظه داخلی، شاید با توجه به حجم ذخیره‌سازی‌شان این‌طور باشند؛ بنابراین، قابل‌اتکا نیست که توصیه‌ای ساده به مرورگرها برای ذخیره‌سازی فایل‌ها در حافظه داخلی، راهبرد قابل‌دوامی باشد، مگر اینکه خطر مرتبط با مجوز فایل همچنان مورد خطاب قرار بگیرد؛ بنابراین، یک راه‌حل کوتاه‌مدت برای مرورگرها می‌تواند ذخیره فایل‌های غیر حساس یا بزرگ (مانند کلیپ‌های ویدئویی) در حافظه خارجی باشد. با این حال، بهتر است به کاربران هشدار داده شود که این فایل‌ها ممکن است توسط برنامه‌های دیگر مورد دسترسی قرار بگیرند. موارد در شکل شماره ۳ ارائه شده است.

شکل شماره (۳): توزیع آسیب‌پذیری‌های امنیتی شناسایی‌شده در مرورگرهای سبک تلفن همراه اندرویدی



مطابق شکل ۳ مرورگر دلفین بیش‌ترین تعداد آسیب‌پذیری‌های امنیتی شناسایی‌شده را نشان می‌دهد که بیانگر سطح بالاتری از ریسک در ذخیره‌سازی و مدیریت داده‌های

کش است. مرورگر UC Browser نیز با وجود عملکرد بهتر نسبت به دلفین، همچنان دارای ضعف‌های امنیتی قابل‌توجهی است

در مقابل، مرورگر CM Browser کمترین میزان آسیب‌پذیری را داراست که می‌تواند ناشی از کنترل بهتر دسترسی فایل‌ها و محدودسازی ذخیره‌سازی داده‌ها باشد. مرورگر پیش‌فرض سامسونگ در وضعیت میانی قرارمی‌گیرد که نشان می‌دهد استفاده از حافظه داخلی به‌تنهایی تضمین‌کننده امنیت کامل نیست.

ب- پیشنهادها

با توجه به نتایج پژوهش و آسیب‌پذیری‌های شناسایی‌شده، پیشنهادهای زیر ارائه می‌گردد:

۱. برای طراحان مرورگرها:

- استفاده از حافظه داخلی رمزنگاری‌شده برای ذخیره فایل‌های کش و پایگاه‌های داده.
- اعمال سطح دسترسی محدود[1] بر روی فایل‌های ذخیره‌شده، به‌گونه‌ای که برنامه‌های ثالث امکان خواندن یا تغییر آن‌ها را نداشته باشند.
- پیاده‌سازی مکانیسم‌های ضد فیشینگ و اعتبارسنجی فایل‌های کش تا از دست‌کاری و جایگزینی فایل‌های ذخیره‌شده جلوگیری شود.

۲. برای کاربران:

- استفاده از مرورگرهای رسمی و معتبر (مانند Chrome یا Firefox) که از استانداردهای امنیتی بیشتری پشتیبانی می‌کنند.
- پرهیز از ذخیره‌سازی اطلاعات حساس (مانند رمز عبور یا اطلاعات بانکی) در مرورگرهای سبک.
- پاک‌سازی دوره‌ای حافظه کش مرورگرها برای کاهش احتمال سوءاستفاده.

۳. برای پژوهش‌های آتی:

- بررسی امنیت مرورگرهای جدیدتر و پرکاربرد مانند Brave و Opera Mini.
- توسعه ابزارهای نظارتی برای شناسایی آسیب‌پذیری‌های ذخیره‌سازی در مرورگرهای موبایل.

1. Permissions

- مطالعـه تطبیقی بـر روی مرورگرهای سـامانه‌عامل‌های مختلـف (iOS، HarmonyOS و ...) برای ارائه راهکارهای جامع‌تر هم پیشنهاد داده می‌شود.

منابع

[1] Adebayo, O. S., & Aziz, N. A. (2015). Static code analysis of permission-based features for Android malware classification using Apriori algorithm with particle swarm optimization. Journal of Information Assurance and Security, 10(4), 1-10.

[2] Amrutkar, C., Singh, K., Verma, A., & Traynor, P. (2012). VulnerableMe: Measuring systemic weaknesses in mobile browser security. In Information Systems Security (pp. 16-34). Springer.

[3] Amrutkar, C., Traynor, P., & van Oorschot, P. C. (2015). An empirical evaluation of security indicators in mobile web browsers. IEEE Transactions on Mobile Computing, 14(5), 889-903.

[4] Aung, Z., & Zaw, W. (2013). Permission-based Android malware detection. International Journal of Scientific and Technology Research, 2(3), 228-234.

[5] Wadkar, H., Mishra, A., & Dixit, A. (2014). Prevention of information leakages in a web browser by monitoring system calls. In Proceedings of the IEEE International Advance Computing Conference (IACC) (pp. 199-204). IEEE.

[6] Tsalis, N., Virvilis, N., Mylonas, A., Apostolopoulos, T., & Gritzalis, D. (2015). Browser blacklists: A utopia of phishing protection. Springer.

[7] Liu, X., Zhou, Z., Diao, W., Li, Z., & Zhang, K. (2015). An empirical study on Android for saving non-shared data on public storage. In ICT Systems Security and Privacy Protection (pp. 542-556). Springer.

[8] Azfar, A., Choo, K.-K. R., & Liu, L. (2016). An Android communication app forensic taxonomy. Journal of Forensic Sciences. https://doi.org/10.1111/1556-4029.13164

[9] Bansal, C., Preibusch, S., & Milic-Frayling, N. (2015). Cache timing attacks revisited: Efficient and repeatable browser history, OS and network sniffing. In ICT Systems Security and Privacy Protection (pp. 97-111). Springer.

[10] Chamkoori, A., & Katebi, S. (2023). Security and storage improvement in distributed cloud data centers by increasing reliability based on PSO and AIS algorithms. Concurrency and Computation: Practice and Experience, 35(6). https://doi.org/10.1002/cpe.7375

[11] Chamkoori, A., & Katebi, S. (2021). Robustness of the storage in cloud data centers based on simple swarm optimization algorithm. Security and Communication Networks, Article ID 6653214. https://doi.org/10.1155/2021/6653214

[12] Burguera, I., Zurutuza, U., & Nadjm-Tehrani, S. (2011). Crowdroid: Behavior-based malware detection system for Android. In Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (pp. 15-26). ACM.

[13] Chamkoori, A., & Katebi, S. (2022). Improving storage in distributed cloud data centers by increasing reliability using collective intelligence algorithms. Journal of Southern Communication Engineering, 12(47), 71-83.

[14] Tian, D., Ma, Y., Balasubramanian, A., Liu, Y., Huang, G., & Liu, X. (2022). Characterizing embedded web browsing in mobile apps. IEEE Transactions on Mobile Computing, 21(11), 3912-3925. https://doi.org/10.1109/TMC.2021.3065945

[15] D'Orazio, C., & Choo, K.-K. R. (2015). An adversary model to evaluate DRM protection of video contents on iOS devices. Computers & Security, 56, 94-110.

[16] Daryabar, F., & Dehghantanha, A. (2016). Forensic investigation of OneDrive, Box, Google Drive and Dropbox applications on Android and iOS devices. Australian Journal of Forensic Sciences, 48(6), 1-28. https://doi.org/10.1080/00450618.2015.1110620

[17] Das, M. L., & Samdaria, N. (2014). On the security of SSL/TLS-enabled applications. Applied Computing and Informatics, 10(1), 68-81.

[18] Desmet, L., & Johns, M. (2014). Real-time communications security on the web. IEEE Internet Computing, 18(6), 8-10.

[19] Do, Q., Martini, B., & Choo, K.-K. R. (2014). Enforcing file system permissions on Android external storage. In Proceedings of the 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (pp. 949-954). IEEE.

[20] Do, Q., Martini, B., & Choo, K.-K. R. (2015). Exfiltrating data from Android devices. Computers & Security, 48, 74-91.