Strategic Framework for Automated Data Extraction from Walmart Canada: Technical Analysis of robots.txt Compliance and Data Architecture

The systematic acquisition of large-scale retail data from a major digital infrastructure like Walmart Canada necessitates a rigorous adherence to established web protocols and an intimate understanding of the platform's architectural constraints. For data engineers and market analysts, the robots.txt file of walmart.ca serves as the primary technical manifest, outlining the boundaries between permissible discovery and prohibited interference.[1] In an era where e-commerce data serves as a real-time economic indicator, the ability to extract pricing, inventory, and catalog information with precision—while remaining within the technical and ethical guardrails set by the retailer—is a critical competency.[3] This analysis provides an exhaustive interrogation of Walmart Canada's public directives, exploring the intersection of the Robots Exclusion Protocol, search engine optimization (SEO) data structures, and the sophisticated anti-bot defenses that define the current scraping landscape.[1]

The Robots Exclusion Protocol as a Governance Mechanism

The foundational document for any automated interaction with Walmart Canada is the robots.txt file. Hosted at the root directory of the domain, this plain text file communicates the site owner's preferences regarding bot behavior.[1] While the protocol is technically advisory and relies on the compliance of the visiting agent, it represents the formal technical boundary established by Walmart's engineering team.[1] The file for walmart.ca is particularly detailed, containing versioned notes such as "# MG IG" and "# NV," which suggest that the directives are subject to frequent revision by specific developers or internal teams to address emerging crawl patterns or infrastructure shifts.[2]

Core Directives and Access Management

The Walmart Canada robots.txt utilizes the universal User-agent: * wildcard to apply its primary set of restrictions to all crawlers, supplemented by specific instructions for search engine bots like Bingbot.[2] The architecture of these rules reflects a dual priority: protecting the computational resources of the internal search engine and ensuring that sensitive user-facing paths remain shielded from automated indexing.[2]

Directive Category	Path Examples	Technical Justification
Search Infrastructure	/search/*, /recherche?*, /en/search	Prevents automated agents from "hammering" the database-intensive internal search functionality.[2]
User State & Security	/cart, /panier, /sign-in*, /account/login/*	Protects endpoints that handle sensitive user session data and prevents the indexing of transactional pages.[2]
Refined Navigation	/en/*+*, /fr/*+*, /*?f=*	Blocks "facet explosion" where combinatorial filters create millions of near-duplicate pages.[2]
Internal Assets	/webjars/*, /css-pattern-library, /assets/*	Reduces bandwidth consumption by preventing the crawling of non-content supporting files.[2]
Specialized Services	/en/kiosk/*, /en/email/gifts-voucher	Protects specific business logic and high-fraud-risk endpoints from automated discovery.[2]

The prohibition of refined browse pages (/en/*+*) is a critical technical insight. In modern e-commerce architectures, selecting multiple attributes—such as size, color, and brand—generates unique URLs that combine these facets using the + character.[2] Without this block, a crawler could easily fall into an "infinite loop" or "spider trap," attempting to index every possible combination of product attributes, which would provide no incremental value to a search index while placing immense strain on the host server.[2]

Granular Product Page Permissions

A sophisticated nuance within the Walmart robots.txt is the treatment of Item Pages (IP). The file contains a sequence of directives that demonstrate a highly granular approach to directory management [2]:

1. Disallow: /en/ip/*
2. Allow: /en/ip/*/*

This logic indicates that Walmart wishes to prevent bots from accessing the top-level directory index of the /ip/ path—which might otherwise allow a bot to list or iterate through products in a non-linear fashion—while explicitly permitting the crawling of deep-linked product detail pages (PDPs) that follow the /en/ip/[slug]/[id] format.[2] This suggests that as long as a scraper discovers a specific product URL through a sitemap or a valid category link, the retrieval of that specific page is compliant with the technical rules.[2]

Discovery Architectures: Leveraging XML Sitemaps

Rather than employing traditional "spidering"—where an agent recursively clicks every link on a page—Walmart Canada provides a comprehensive sitemap index to guide automated discovery.[2] This "inclusion protocol" is designed to be more efficient than crawling the document object model (DOM), as it provides a structured list of canonical URLs that Walmart intends to be indexed.[1]

The Hierarchy of Walmart Sitemaps

The sitemaps provided in the robots.txt are categorized by inventory source (1P vs. 3P), language (English vs. French), and content type.[2] This categorization allows for highly targeted data acquisition strategies.

Sitemap Type	URL Example Path	Data Utility for Scrapers
Category Sitemaps	sitemap-categories.xml	Essential for mapping the product taxonomy and internal hierarchy of the Walmart catalog.[2]
Brand Sitemaps	sitemap-brand-1p-en.xml	Provides direct access to brand-specific landing pages for first-party inventory.[2]

Product 1P	sitemap-product-1p-en.xml	The primary source for links to Walmart's own inventory (first-party products).[2]
Product 3P	sitemap-product-3p-en.xml	Contains URLs for the third-party marketplace, enabling monitoring of external sellers.[2]
Product Head	sitemap-product-3p-head-en.xml	Likely contains "head" or high-traffic products, useful for prioritizing trending items.[2]
Topic & CP	sitemap-topic.xml, sitemap-cp.xml	Includes thematic content pages and category-specific landing pages.[2]

Following the Sitemap protocol is considered the most "technically polite" method of scraping, as it respects the server's resource allocation and avoids the high-cost search and filter endpoints.[1] Furthermore, XML sitemaps often include metadata such as the <lastmod> tag, which indicates the last time a page was updated, allowing scrapers to implement incremental updates rather than re-crawling the entire catalog.[10]

Technical Limits of Sitemaps

Under the standards established by major search engines, a single XML sitemap cannot exceed 50,000 URLs or 50 megabytes in uncompressed size.[10] For a retailer of Walmart's scale, which likely possesses millions of individual Stock Keeping Units (SKUs), the sitemap index found in the robots.txt acts as a gateway to multiple nested sitemap files.[10] This architecture ensures that the crawl remains deterministic and that the bot does not miss deep-linked products that might be buried multiple clicks away from the homepage.[10]

Anatomy of the Walmart Product Detail Page (PDP)

A successful scraping operation requires more than just discovery; it demands a deep understanding of how data is structured within the Product Detail Page. Walmart employs specific URL patterns and metadata standards that can be leveraged to ensure data accuracy and persistence.[15]

URL Logic and Product Identifiers

The URL of a Walmart product page is typically formatted as https://www.walmart.ca/en/ip/[slug]/[item_id].[2] The "slug" is a human-readable, hyphenated version of the product title used for SEO purposes, while the "item_id" is the definitive numerical identifier for the product in Walmart's database.[15]

Scrapers should prioritize the Item ID for tracking, as the slug may change if the product is renamed or if SEO strategies shift.[16] Walmart utilizes several types of product identifiers, each with specific formatting requirements:

- **GTIN (Global Trade Item Number)**: A 14-digit number. If the provided number is shorter, leading zeros are added to reach the 14-digit standard.[19]
- **UPC (Universal Product Code)**: A 12-digit number primarily used in North America. Like GTIN, it requires leading zeros if the source code is only 8 digits (UPC-E).[19]
- **WIN (Walmart Item Number)**: An internal 9-digit identifier assigned by Walmart to link supplier configurations to sellable GTINs.[20]
- **SKU (Stock Keeping Unit)**: A supplier-defined alphanumeric code. Unlike other IDs, SKUs can include letters, hyphens, and spaces.[20]

Identifier	Digits	Format Notes
GTIN	14	Standardized with leading zeros.[19]
UPC	12	Primary identifier for US/Canada items.[19]
EAN	13	Global standard; often converted to GTIN-14.[19]
WIN	9	Internal Walmart system code.[20]
Item ID	Variable	Found at the end of the URL; primary for web scraping.[16]

SEO Meta-Data and Content Standards

Walmart provides explicit guidelines for how product content should be displayed, which serves as a blueprint for scrapers seeking to identify the most relevant data fields.[15] For

instance, a Title Tag is recommended to be between 50 and 60 characters and should follow the format: Brand + Defining Qualities + Item Name + Pack Count.[15] By understanding these rules, a scraper can implement validation logic to ensure that the extracted title matches the expected manufacturer-provided format.[15]

Furthermore, image URLs on Walmart must meet stringent technical requirements. They must end in a valid image file type (.jpg,.png,.gif) and link directly to a public file rather than an HTML wrapper.[22] A notable constraint is that image URLs must only utilize specific communication ports: 80, 443, 8080, or 8443.[22] If a scraper encounters an image URL on an unsupported port, it may indicate a redirect or a legacy asset that could be unstable.[22]

Technical Impediments and Anti-Bot Infrastructure

Extracting data from Walmart is "brutally difficult" due to the layered defense systems deployed to distinguish human traffic from automated agents.[5] Walmart leverages industry-leading anti-bot solutions, including Akamai Bot Manager and PerimeterX (now known as HUMAN).[5]

Fingerprinting and TLS Handshakes

Most basic HTTP clients, such as the standard Python requests library, are identified and blocked almost immediately because they do not exhibit the complex "fingerprint" of a modern web browser.[5] Advanced defense systems analyze the Transport Layer Security (TLS) handshake, the order of HTTP headers, and the support for modern browser features like HTTP/2.[5]

To successfully bypass these barriers, scrapers must adopt one of several advanced strategies:

1. **Headless Browser Automation**: Using Playwright, Selenium, or Puppeteer allows the scraper to execute JavaScript, handle cookies, and mimic the rendering behavior of a real user.[5]
2. **Header Optimization**: Scrapers must provide a comprehensive set of headers, including User-Agent, Accept-Language, and Sec-Ch-Ua, ensuring they match the expected profile of a legitimate browser.[3]
3. **JavaScript Rendering**: Much of Walmart's critical data—such as dynamic price updates and store-level stock status—is not present in the initial HTML but is loaded via JavaScript or embedded in a __NEXT_DATA__ script tag.[24]

The Role of Persistent Sessions and Cookies

Walmart uses cookies to "lock" a session to a specific geographical region or store.[24] For scrapers, this is a dual-edged sword. While it allows for accurate local pricing, it requires the

script to maintain state across multiple requests.[24] If a scraper fails to preserve cookies, it may find its session reset, or it may receive "Global" prices that do not reflect the actual in-store reality of a specific zip code.[24]

Anti-Bot Layer	Mechanism of Action	Mitigation Strategy
Akamai	Analyzes network signatures and IP reputation.[24]	Use of residential proxies and TLS mimicry.[3]
PerimeterX	Monitors behavioral patterns and mouse movements.[24]	Implementing natural delays and headless browser automation.[5]
CAPTCHA	Challenges users when automated behavior is detected.[5]	Automatic CAPTCHA solvers or rotating IPs before detection.[5]
Rate Limiting	Blocks IPs that exceed a certain threshold of requests per second.[3]	Distributed scraping across a large proxy pool.[3]

Advanced Extraction Methodologies

Beyond simple HTML parsing, modern Walmart scraping involves the extraction of "hidden web data" and the management of large-scale proxy rotations.[5]

Extracting Structured Data from JSON Objects

A second-order insight into Walmart's frontend architecture is the use of the Next.js framework, which embeds the page's initial data into a JSON object within the HTML source.[25] This object, typically found in a <script id="__NEXT_DATA__"> tag, contains the full product catalog for that page in a structured format.[25] Extracting data directly from this JSON is significantly more reliable than using CSS selectors or XPaths, as the JSON structure is less prone to the frequent layout changes that often break traditional scrapers.[5]

The structure of the __NEXT_DATA__ object generally includes:

- props.pageProps.initialData.searchResult.itemStacks: For search result pages.[25]
- props.pageProps.initialData.data.product: For individual product detail pages.[25]

Proxy Strategy and Performance

To avoid IP blocks, the use of residential proxies is nearly mandatory for high-volume scraping.[3] Residential proxies route traffic through home internet connections, making them indistinguishable from real customers.[3] Research indicates a wide range of success rates among proxy providers when targeting Walmart's infrastructure.[5]

Provider	Success Rate	Avg. Latency	Cost Metric
Zenscrape	100%	N/A	High Reliability [5]
ScrapingBee	98%+	7.2s	Performance Focus [5]
Scrapingant	62%	N/A	Competitive Pricing [5]

The choice of proxy provider often depends on the required latency. While residential proxies are more successful at bypassing blocks, they typically exhibit higher latency (5-10 seconds) compared to datacenter proxies (1-2 seconds) because the requests must travel through multiple hops in a peer-to-peer network.[3]

official API Integration: A Compliant Alternative

For organizations that require a high degree of stability and have a formal business relationship with Walmart, the use of official APIs is the recommended pathway.[27] Walmart offers a diverse ecosystem of APIs tailored to different partnership types.[28]

The Marketplace and Supplier APIs

The **Marketplace API** is the primary tool for third-party sellers to manage their listings, inventory, and orders.[27] This API uses REST standards and requires authentication via WM_CONSUMER.ID and a private key generated through the Walmart Developer Portal.[30] It is divided into two primary groups:

- **Item API**: For the submission, updating, and retirement of product listings.[30]
- **Transaction API**: For managing price changes, inventory levels, and order fulfillment.[30]

Similarly, the **Supplier API** is designed for first-party (1P) suppliers who sell directly to Walmart (Warehouse or Drop Ship Vendors).[27] These APIs provide the most stable data connection but are restricted to verified partners and do not allow for the broad competitive

monitoring that web scraping permits.[27]

Affiliate and Catalog APIs

The **Affiliate Catalog Product API** provides a middle ground for developers who want to access Walmart's product data for marketing purposes.[31] This API allows for the paginated retrieval of the entire product catalog, which can be filtered by category, brand, or special offers like "Rollback" or "Clearance".[31]

The response from the Affiliate API includes:

- nextPage: A URI to the subsequent set of results, facilitating systematic catalog extraction.[31]
- items: A list of product metadata, including titles, prices, and affiliate-ready links.[31]
- WM_SEC.AUTH_SIGNATURE: A security signature with a time-to-live (TTL) of 180 seconds to prevent replay attacks.[31]

Ethical and Legal Considerations in Data Mining

The intersection of technical capability and legal restriction is a primary concern for any data-driven enterprise. Walmart's position on automated data extraction is clearly defined in its Terms of Use and has been tested in several high-profile legal cases.[5]

Terms of Use and "Data Mining"

Walmart's Terms of Service explicitly prohibit the use of any "robot, spider, site search/retrieval application or other manual or automatic device to retrieve, index, 'scrape,' 'data mine' or otherwise gather any Materials" without prior written consent.[5] This prohibition is a binding agreement for any user who interacts with the site, particularly those who have created an account or engaged in transactions.[5]

However, the legal landscape regarding the scraping of *publicly available* data—such as product prices and descriptions—remains a subject of ongoing debate and litigation.[3] The 2020 lawsuit between **Walmart Inc. and Bright Data Ltd.** (formerly Luminati) highlighted the tension between a site's right to control its infrastructure and the public's right to access data that is openly displayed on the internet.[5]

Privacy and Personal Information

A significant legal risk emerges when scrapers inadvertently collect personal information from reviews or user profiles.[5] Walmart's Privacy Notice outlines the categories of personal information it collects, including "Device and Online Identifiers" and "Internet or other network activity information".[33] Under regulations like the California Consumer Privacy Act (CCPA), users have the right to request the deletion of their personal data.[32] A scraper that maintains a

database of user reviews may inadvertently become a "data processor" or "data controller," subjecting them to the same compliance requirements as the original retailer.[32]

To minimize legal exposure, ethical scraping practices suggest:

- **Avoid Restricted Areas**: Never attempt to scrape data behind a login wall or access the /account/ or /cart/ directories.[5]
- **Respect Rate Limits**: Even if not explicitly enforced, implementing a crawl delay (such as the 1-second delay requested for Bingbot) demonstrates technical politeness and reduces the risk of being accused of a "denial of service" attack.[2]
- **Identify Your Bot**: Providing a transparent User-Agent string with a contact URL allows the site owner to communicate directly with the bot operator.[5]

Best Practices for Successful Implementation

For a professional-grade scraping implementation targeting Walmart Canada, several technical best practices should be integrated into the system architecture.

Technical Politeness and Server Load

The directive Crawl-delay: 1 found in the robots.txt for Bingbot should be treated as a minimum baseline for custom bots.[2] The relationship between crawl frequency and server impact is linear. By distributing requests over a longer period, a scraper can avoid triggering anomaly detection systems that look for "bursty" traffic patterns.[1]

$$\text{Total Crawl Time} = \frac{\text{Number of URLs}}{\text{Concurrent Threads}} \times \text{Crawl Delay}$$

If a scraper aims to index 100,000 URLs from a sitemap using 10 concurrent threads and a 1-second delay, the process will take approximately 10,000 seconds (or roughly 2.8 hours).[2] Attempting to compress this timeframe by increasing concurrency without a corresponding increase in proxy diversity will likely result in a permanent IP ban.[3]

Handling Dynamic Content and Pagination

Walmart's search results and category pages often impose a pagination limit of 25 pages (roughly 1,000 products), even if more items are available in the total result set.[25] To bypass this limit and ensure a comprehensive scrape, practitioners must use "Filter Batching" or "Query Splitting".[25]

By applying additional filters—such as specific price ranges, brands, or retailers—the scraper can generate multiple unique queries that each return fewer than 1,000 items.[25] For example, instead of searching for "laptops" and hitting the 25-page limit, a scraper might search for

"laptops" in the $200-$300 range, then the $300-$400 range, and so on.[25] This strategy effectively reconstructs the full catalog without ever exceeding the technical constraints of a single search query.[25]

Localization and Store Selection

Because Walmart Canada operates as a series of regional markets, the data retrieved can vary significantly based on the assigned "Store ID".[24] A professional scraper must be able to:

1. **Set a Zip Code**: Send a request to the location-setting endpoint to establish a geographic context.[24]
2. **Extract Cookies**: Capture the membership or store cookies that the server returns.[24]
3. **Persist the Session**: Use these cookies in all subsequent requests for that region to ensure the prices and availability are accurate for the target market.[24]

Parameter Type	Example Value	Function
customer_zipcode	M5V 2H1	Sets the regional context for shipping and stock.[26]
store_id	3111	Targets a specific physical location for in-store pickup data.[24]
sort	price_low	Organizes results to ensure the most relevant items are captured first.[25]
q	search_query	The primary keyword parameter for the search endpoint.[25]

Conclusion: The Future of E-Commerce Data Acquisition

The technical interrogation of Walmart Canada’s robots.txt and its underlying data architecture reveals a platform that is both highly structured and deeply defended.[2] Success in this environment requires a transition from rudimentary "web scraping" to a more sophisticated "data engineering" approach that respects the site's rules while intelligently navigating its constraints.[1]

By leveraging XML sitemaps for discovery, extracting structured JSON from the __NEXT_DATA__ tag for accuracy, and utilizing high-quality residential proxies for resilience, practitioners can build automated systems that provide high-fidelity insights into the e-commerce landscape.[2] As retailers continue to deploy more advanced AI-driven anti-bot systems, the ability to mimic human browsing behavior and respect the technical signals provided in the robots.txt will remain the defining factors of a successful data acquisition strategy.[5] The future of this field lies in the ability to balance the intense demand for real-time information with the technical and legal requirements of the platforms that host it.[3]

Works cited

1. What is robots.txt? | Robots.txt file guide - Cloudflare, accessed on January 7, 2026, https://www.cloudflare.com/learning/bots/what-is-robots-txt/
2. robots.txt - Walmart.ca, accessed on January 7, 2026, https://www.walmart.ca/robots.txt
3. How to Scrape Walmart in 2025: A Step-by-Step Tutorial - Oxylabs, accessed on January 7, 2026, https://oxylabs.io/blog/how-to-scrape-walmart-data
4. Walmart API - Walmart Scraper - Web Scraping Services, accessed on January 7, 2026, https://www.xbyte.io/walmart-web-scraping/
5. Walmart Scraping Teardown | ScrapeOps, accessed on January 7, 2026, https://scrapeops.io/websites/walmart/
6. What Is A Robots.txt File? A Guide to Best Practices and Syntax - Moz, accessed on January 7, 2026, https://moz.com/learn/seo/robotstxt
7. robots.txt - Wikipedia, accessed on January 7, 2026, https://en.wikipedia.org/wiki/Robots.txt
8. Robots.txt File - BigCommerce Support, accessed on January 7, 2026, https://support.bigcommerce.com/s/article/Understanding-the-Robots-txt-File
9. Walmart Product API - Google Search API, accessed on January 7, 2026, https://www.searchapi.io/docs/walmart-product-api
10. How To Add Your Sitemap To Your Robots.txt File - WooRank, accessed on January 7, 2026, https://www.woorank.com/en/blog/how-to-locate-a-sitemap-in-a-robots-txt-file
11. SEO: Manage Crawling, Indexing with Robots Exclusion Protocol - Practical Ecommerce, accessed on January 7, 2026, https://www.practicalecommerce.com/seo-manage-crawling-indexing-with-robots-exclusion-protocol
12. What is the purpose of a Robots.txt file and do you have to include all sitemaps you use?, accessed on January 7, 2026, https://www.reddit.com/r/SEO/comments/ssqab3/what_is_the_purpose_of_a_robotstxt_file_and_do/
13. 8 Sitemap Examples + Types of Sitemaps and Best Practices - Semrush, accessed on January 7, 2026, https://www.semrush.com/blog/sitemap-examples/
14. Multiple Sitemap: entries in robots.txt? - Stack Overflow, accessed on January 7, 2026,

https://stackoverflow.com/questions/2594179/multiple-sitemap-entries-in-robots-txt
15. SEO Best Practices for Item Setup - Marketplace Learn - Walmart, accessed on January 7, 2026, https://marketplacelearn.walmart.com/ca/guides/Item%20setup/Item%20content,%20imagery,%20and%20media/seo-best-practices-for-item-setup
16. Walmart Product Data API - Unwrangle Documentation, accessed on January 7, 2026, https://docs.unwrangle.com/walmart-product-data-api/
17. How To Locate The Product ID Of A Prodcut On Walmart.com - Sellegr8 Knowledge Base, accessed on January 7, 2026, https://docs.sellegr8.com/article/33-how-to-locate-the-product-id-on-walmart
18. Walmart Product API - ScraperAPI Documentation, accessed on January 7, 2026, https://docs.scraperapi.com/making-requests/structured-data-collection-method/walmart-product-api
19. Item Setup Tips - Marketplace Learn - Walmart, accessed on January 7, 2026, https://marketplacelearn.walmart.com/ca/guides/Item%20setup/Item%20setup/item-setup-tips
20. Product Identifiers in the Catalog - HelpDocs, accessed on January 7, 2026, https://supplierone.helpdocs.io/article/5de7urtxsp-catalog-product-identifiers
21. Product Content Policy - Marketplace Learn - Walmart, accessed on January 7, 2026, https://marketplacelearn.walmart.com/ca/guides/Item%20setup/Item%20content,%20imagery,%20and%20media/product-content-policy
22. Item Image Guidelines - Marketplace Learn - Walmart, accessed on January 7, 2026, https://marketplacelearn.walmart.com/ca/guides/Item%20setup/Item%20content,%20imagery,%20and%20media/item-image-guidelines
23. Product detail page: Image guidelines & requirements - Guides | Marketplace Learn - Walmart, accessed on January 7, 2026, https://marketplacelearn.walmart.com/guides/Item%20setup/Item%20content,%20imagery,%20and%20media/Product-detail-page:-Image-guidelines-&-requirements
24. How to Scrape Complete Grocery Data from Walmart with Python, accessed on January 7, 2026, https://scrape.do/blog/walmart-scraping/
25. How to Scrape Walmart.com Product Data (2026 Update) - Scrapfly, accessed on January 7, 2026, https://scrapfly.io/blog/posts/how-to-scrape-walmartcom
26. How to Scrape Walmart.com - Traject Data, accessed on January 7, 2026, https://trajectdata.com/how-to-scrape-walmart-com/
27. In-Depth Guide to the Walmart API - Scrapfly, accessed on January 7, 2026, https://scrapfly.io/blog/posts/guide-to-walmart-api
28. Walmart Developer Portal, accessed on January 7, 2026, https://developer.walmart.com/
29. APIs & Services - Walmart I/O, accessed on January 7, 2026, https://walmart.io/apirefservices
30. API Integration Methods - Marketplace Learn - Walmart, accessed on January 7,

2026, https://marketplacelearn.walmart.com/ca/guides/Getting%20started/Getting%20ready%20to%20sell/api-integration-methods?locale=en-CA
31. Catalog Product - Walmart I/O, accessed on January 7, 2026, https://walmart.io/docs/affiliates/v1/catalog-product
32. Walmart California Consumer Privacy Act Notice, accessed on January 7, 2026, https://corporate.walmart.com/privacy-security/california-privacy-rights
33. Walmart Customer Privacy Notice (Online and In-Store), accessed on January 7, 2026, https://corporate.walmart.com/privacy-security/walmart-privacy-notice
34. Walmart Global Marketplace Seller Privacy Notice, accessed on January 7, 2026, https://corporate.walmart.com/privacy-security/walmart-marketplace-seller-privacy-notice