Appendix

Raw Data Tables: Vulnerability Distributions Across Tested Websites

Table 1: Distribution of WebKit-Specific Vulnerabilities by Category

Vulnerability Category	Count	Percentage	Mean Risk Score
Unsanitized input reflected in DOM elements	58	37%	6.8
Unsafe event handler implementations	46	29%	7.2
iOS-specific touch/gesture event issues	28	18%	5.9
WebKit-specific CSS injection vulnerabilities	19	12%	4.7
Shadow DOM implementation issues	6	4%	8.1
Total vulnerabilities detected	**157**	**100%**	**6.2**

Table 2: Security Header Implementation Analysis

Security Header	Present	Missing	Percentage Present	Vulnerable to WebKit-specific Bypass
Content-Security-Policy	68	89	43%	46 (67%)
X-Frame-Options	103	54	66%	12 (12%)
X-Content-Type-Options	95	62	61%	0 (0%)
Strict-Transport-Security	110	47	70%	0 (0%)
X-XSS-Protection	79	78	50%	51 (65%)
Referrer-Policy	83	74	53%	0 (0%)

Table 3: Content Security Policy Implementation Details

CSP Implementation Issue	Count	Percentage of Sites with CSP	Risk Score Impact
Missing base-uri directive	53	78%	+1.7
'unsafe-inline' without nonces/hashes	45	66%	+2.1
Missing frame-ancestors directive	38	56%	+1.2
Missing object-src 'none'	32	47%	+0.8

CSP Implementation Issue	Count	Percentage of Sites with CSP	Risk Score Impact
Data URI allowed in script-src	29	43%	+1.9
Missing default-src	22	32%	+1.3

Table 4: Website Categories and Risk Scores

Website Category	Number of Sites	Mean Risk Score	Sites with WebKit-Specific Vulnerabilities
E-commerce	37	6.7	26 (70%)
Financial Services	22	7.8	19 (86%)
News/Media	31	5.9	18 (58%)
Government	14	4.8	6 (43%)
Social Media	18	6.9	13 (72%)
Technology	23	5.4	10 (43%)
Educational	12	4.1	4 (33%)

Table 5: Platform Comparison (Desktop vs. Mobile Versions)

Platform	Number of Sites Analyzed	Mean Vulnerabilities per Site	Mean Risk Score
Desktop (macOS)	157	2.3	5.7
Mobile (iOS)	157	2.9	6.9
Difference	-	**+0.6 (+23%)**	**+1.2 (+21%)**

Table 6: WebView Configuration Issues in iOS Applications (73 Applications Analyzed)

WebView Configuration Issue	Count	Percentage	Severity Rating
Unrestricted JavaScript interface exposure	45	62%	High
Insufficient URL validation	39	54%	High
Lack of certificate validation	27	37%	High
Insecure local storage handling	31	42%	Medium
File access from file URLs enabled	22	30%	Medium
Universal access from file URLs enabled	19	26%	Critical
Missing Content Security Policy	52	71%	Medium

Table 7: Privacy Protection Bypass Methods Detected

Privacy Bypass Method	Count	Percentage	Effectiveness Rating
Local storage partitioning bypass	28	18%	High
Canvas fingerprinting variants	35	22%	Medium
Navigation timing API abuse	19	12%	Medium
Link decoration techniques	41	26%	High
Alternative cookie mechanisms	32	20%	High
WebKit-specific storage mechanisms	17	11%	Very High
Total sites with ITP bypass	**66**	**42%**	-

Table 8: Mitigation Strategy Effectiveness by Vulnerability Type

Mitigation Strategy	DOM API Vulnerabilities	JavaScript Engine Vulnerabilities	Rendering Engine Vulnerabilities	Permission Boundary Violations	Integration Interface Weaknesses
Content Security Policy	92%	34%	76%	82%	65%
WebView Configuration Hardening	68%	31%	58%	82%	89%

Mitigation Strategy	DOM API Vulnerabilities	JavaScript Engine Vulnerabilities	Rendering Engine Vulnerabilities	Permission Boundary Violations	Integration Interface Weaknesses
Input Validation & Sanitization	79%	18%	87%	45%	51%
Origin Restriction Enforcement	65%	12%	43%	94%	72%
JavaScript Limitations	71%	86%	38%	66%	77%