WEB APPLICATION PENTESTING REPORT

Leader Name: priyanshu raj(2JI22CS070)

suraj(2JI22CS108)

Shashank K (2JI22CS095)

Pratik p(2JI22CS068)

Website Analyzed: https://www.shopify.com/in
Date: May 2, 2025

Table of Contents

1. Executive Summary
2. Introduction
3. Tools and Methodology
4. Technology Stack Overview
5. Security Configuration Analysis
6. Performance Audit (Mobile)
7. Recommendations
8. Conclusion
9. References

---

1. Executive Summary

This report provides a comprehensive audit of the Shopify website (https://www.shopify.com/in) based on technological stack, performance metrics, and security configuration. The analysis aims to offer insights into the current architecture, highlight areas for improvement, and propose best practices for enhancing overall performance and security.

The audit reveals that Shopify uses a sophisticated technology stack, including React, Cloudflare, and Google Tag Manager. However, critical security headers are missing, and mobile performance lags due to extensive JavaScript blocking and inefficient resource handling.

---

2. Introduction

As the digital landscape continues to evolve, website performance and security have become crucial pillars for user experience and brand credibility. This audit focuses on Shopify's Indian domain to evaluate:

- Web technologies and frameworks in use
- Frontend performance (particularly on mobile)
- HTTP security headers
- CDN, hosting, and backend infrastructure

The analysis leverages publicly available tools and browser extensions to simulate real-world diagnostics.

---

3. Tools and Methodology

To ensure an accurate and thorough review, the following tools were utilized:

- **Wappalyzer**: To detect technologies, analytics, frameworks, and platforms used
- **SecurityHeaders.io**: To assess HTTP response headers related to web security
- **Google PageSpeed Insights**: To analyze performance metrics focusing on mobile devices
- **Netcraft**: To identify hosting and network security attributes (inferred where data was unavailable)

Each tool was used to extract technical details and then cross-verified for consistency.

---

4. Technology Stack Overview

4.1 E-commerce & CMS Platforms

- Shopify (Primary e-commerce platform)
- Magento (Legacy or segment-based usage)

4.2 Programming Languages & Databases

- PHP (Backend)
- MySQL (Database)

4.3 JavaScript Frameworks

- React
- React Router (v7.5.3)
- Core-js (v3.32.2)

4.4 Analytics & Marketing Tools

- Google Analytics GA4

- Facebook Pixel
- LinkedIn Insight Tag
- DoubleClick Floodlight

4.5 CDN & Infrastructure

- Cloudflare
- HTTP/3 (for faster and reliable connections)
- Priority Hints (performance-focused)

4.6 SEO & Metadata Standards

- Open Graph (for rich link previews on social media)

4.7 Tag Management

- Google Tag Manager (centralized script deployment)

This diversified tech stack indicates that Shopify emphasizes scalability, performance, and marketing integration.

---

5. Security Configuration Analysis

5.1 Overall Rating: D (via SecurityHeaders.io)

The following HTTP response headers were analyzed:

Header	Present	Description
Strict-Transport-Security	Yes	Enforces HTTPS over time
X-Content-Type-Options	Yes	Prevents MIME-sniffing
Content-Security-Policy	No	Prevents loading malicious scripts (XSS protection)
X-Frame-Options	No	Mitigates clickjacking attacks
Referrer-Policy	No	Controls referrer information leakage
Permissions-Policy	No	Restricts access to features like camera and microphone

5.2 Vulnerabilities Identified

- Potential for **Clickjacking** due to missing X-Frame-Options
- Exposure to **Cross-Site Scripting (XSS)** from lack of Content-Security-Policy
- Data leakage via referer headers due to absent Referrer-Policy

5.3 Recommendations

- Configure all missing headers with strict policies

- Periodically review headers to stay aligned with evolving standards

---

6. Performance Audit (Mobile)

6.1 Summary Metrics (from PageSpeed Insights)

Metric	Value	Issue Level
First Contentful Paint	3.5s	Moderate
Largest Contentful Paint	3.9s	Moderate
Total Blocking Time	20,670ms	Critical
Speed Index	3.5s	Moderate
Time to Interactive	>20s	Critical

6.2 Performance Observations

- JavaScript execution time exceeds **39.8 seconds**, blocking the main thread for **41.7 seconds**.
- Large payloads of **unused JavaScript (~176 KB)** and render-blocking CSS detected.
- Poor cache policies and lack of compression for static resources.
- Diagnostic: Heavy third-party integrations contribute to delays.

6.3 Accessibility and Best Practices

- Accessibility Score: **100** (excellent)
- Best Practices Score: **100** (excellent)
- SEO: Good structure, but can benefit from faster load speeds.

---

7. Recommendations and Best Practices

7.1 Security Enhancements

- Implement Content-Security-Policy with strict directives
- Add X-Frame-Options: DENY to prevent framing
- Add Referrer-Policy: strict-origin-when-cross-origin
- Use Permissions-Policy to limit feature access

7.2 Performance Optimization

- **Minify and defer JavaScript**: Use code-splitting and async strategies
- **Lazy-load resources**: Especially for images and third-party iframes
- **Enable compression (GZIP/Brotli)**: For all text-based resources

- **Optimize caching**: Set long cache expiry for static assets
- **Remove unused CSS/JS**: Use PurgeCSS or similar tools

7.3 Monitoring & Maintenance

- Integrate continuous monitoring tools like **Lighthouse CI**
- Schedule regular header audits via tools like **SecurityHeaders.io**
- Periodically review third-party libraries and update them

---

8. Conclusion

The Shopify website showcases a strong technological foundation with modern frameworks and advanced CDN practices. However, the audit highlights a pressing need to improve HTTP security headers and mobile performance.

While marketing and analytics capabilities are robust, the frontend codebase needs streamlining, and critical headers must be configure d to safeguard user data and prevent exploitation.

Following the outlined recommendations will enhance both user experience and platform integrity.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A 172.64.145.93
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-02 06:37 EDT
Nmap scan report for 172.64.145.93
Host is up (0.0097s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE        VERSION
80/tcp   open  http           Cloudflare http proxy
|_http-title: Site doesn't have a title (text/plain; charset=UTF-8).
|_http-server-header: cloudflare
443/tcp  open  ssl/https      Cloudflare
|_http-server-header: cloudflare
|_http-title: 400 The plain HTTP request was sent to HTTPS port
8080/tcp open  http           Cloudflare http proxy
|_http-title: Site doesn't have a title (text/plain; charset=UTF-8).
|_http-server-header: cloudflare
8443/tcp open  ssl/https-alt cloudflare
|_http-server-header: cloudflare
|_http-title: 400 The plain HTTP request was sent to HTTPS port
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: broadband router|VoIP adapter|general purpose
Running (JUST GUESSING): Oracle Virtualbox (98%), Slirp (98%), AT&T embedded (95%), QEMU (94%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:danny_gasparovski:slirp cpe:/a:qemu:qemu
Aggressive OS guesses: Oracle Virtualbox Slirp NAT bridge (98%), AT&T BGW210 voice gateway (95%), QEMU user mode network
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.98 ms 172.64.145.93

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.95 seconds
```

```
1   0.98 ms 172.64.145.93

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.95 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -p- 172.64.145.93
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-02 06:38 EDT
Stats: 0:01:15 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 30.60% done; ETC: 06:42 (0:02:50 remaining)
Stats: 0:03:45 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 06:41 (0:00:00 remaining)
Nmap scan report for 172.64.145.93
Host is up (0.032s latency).
Not shown: 37912 filtered tcp ports (net-unreach), 27615 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
2052/tcp open  clearvisn
2083/tcp open  radsec
2095/tcp open  nbx-ser
8080/tcp open  http-proxy
8443/tcp open  https-alt
8880/tcp open  cddbp-alt

Nmap done: 1 IP address (1 host up) scanned in 253.15 seconds

┌──(kali㉿kali)-[~]
└─$ sudo ifconfig 172.64.145.93
[sudo] password for kali:
172.64.145.93: error fetching interface information: Device not found

┌──(kali㉿kali)-[~]
└─$
```

```
  Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports sequentially - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
           directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
           <Lua scripts> is a comma-separated list of script-files or
           script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  -T<0-5>: Set timing template (higher is faster)
  --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
  --min-parallelism/max-parallelism <numprobes>: Probe parallelization
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
```

```
  --ttl <val>: Set IP time-to-live field
  --spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
  --badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
  -oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIdd3,
     and Grepable format, respectively, to the given filename.
  -oA <basename>: Output in the three major formats at once
  -v: Increase verbosity level (use -vv or more for greater effect)
  -d: Increase debugging level (use -dd or more for greater effect)
  --reason: Display the reason a port is in a particular state
  --open: Only show open (or possibly open) ports
  --packet-trace: Show all packets sent and received
  --iflist: Print host interfaces and routes (for debugging)
  --append-output: Append to rather than clobber specified output files
  --resume <filename>: Resume an aborted scan
  --noninteractive: Disable runtime interactions via keyboard
  --stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
  --webxml: Reference stylesheet from Nmap.Org for more portable XML
  --no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
  -6: Enable IPv6 scanning
  -A: Enable OS detection, version detection, script scanning, and traceroute
  --datadir <dirname>: Specify custom Nmap data file location
  --send-eth/--send-ip: Send using raw ethernet frames or IP packets
  --privileged: Assume that the user is fully privileged
  --unprivileged: Assume the user lacks raw socket privileges
  -V: Print version number
  -h: Print this help summary page.
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -h   172.64.145.93
Nmap 7.95 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN, TCP ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
```

Ecommerce

- Shopify
- Magento

Analytics

- Linkedin Insight Tag
- Google Analytics GA4
- Facebook Pixel 2.9.199

JavaScript frameworks

- React
- React Router 7.5.3

Security

- HSTS

Miscellaneous

Programming languages

- PHP

CDN

- Cloudflare

Databases

- MySQL

Advertising

- DoubleClick Floodlight

Tag managers

- Google Tag Manager

JavaScript libraries

- core-js 3.32.2

Analytics

- Linkedin Insight Tag
- Google Analytics GA4
- Facebook Pixel 2.9.199

JavaScript frameworks

- React
- React Router 7.5.3

Security

- HSTS

Miscellaneous

- Open Graph
- HTTP/3

Databases

- MySQL

Advertising

- DoubleClick Floodlight

Tag managers

- Google Tag Manager

JavaScript libraries

- core-js 3.32.2

Performance

- Priority Hints

Mobile | Desktop

Diagnose performance issues

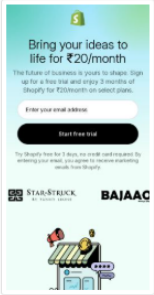
Performance	Accessibility	Best Practices	SEO
50	100	100	61

50

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100



METRICS

Expand view

▲ First Contentful Paint 3.5 s	■ Largest Contentful Paint 3.9 s
▲ Total Blocking Time 20,670 ms	● Cumulative Layout Shift 0
■ Speed Index 3.5 s	

Captured at May 2, 2025, 3:33 PM GMT+5:30
Initial page load

Emulated Moto G Power with Lighthouse 12.6.0
Slow 4G throttling

Single page session
Using HeadlessChromium 134.0.6998.165 with lr

View Treemap

Later this year, insights will replace performance audits. Learn more and provide feedback here.

Try insights

Show audits relevant to: All FCP LCP TBT CLS

DIAGNOSTICS

- ▲ Minimize main-thread work — 41.7 s
- ▲ Reduce JavaScript execution time — 39.8 s
- ▲ Largest Contentful Paint element — 3,920 ms
- ▲ Reduce unused JavaScript — Est savings of 159 KiB
- ▲ Reduce unused CSS — Est savings of 17 KiB
- ▲ Reduce the impact of third-party code — Third-party code blocked the main thread for 430 ms
- ■ Image elements do not have explicit `width` and `height`
- ■ Enable text compression — Est savings of 324 KiB
- ■ Serve static assets with an efficient cache policy — 11 resources found
- ■ Eliminate render-blocking resources — Est savings of 0 ms