HTTPS & TLS Everywhere – Secure Node.js Deployment with Let’s Encrypt

Repo link:
https://github.com/HussienShousha/todo-app

Khaled Ahmed Mohamed – 20226038
Hussien Ahmed Abdelwahab – 20226035
Andrew Makram Isaac – 20226019
Andrew Akram William – 20226018
Ahmed Hossam Abdelhameed – 20226008

INTRODUCTION

- The modern web depends on **secure communication.**
- HTTP transmits data **in plain text** — vulnerable to interception or tampering.
- **HTTPS** = HTTP + Encryption layer (TLS or SSL).
- Our goal: secure our Node.js app using **TLS certificates** from **Let's Encrypt**.

What is HTTPS?

HTTPS is an extension of HTTP that uses TLS to encrypt communication between client and server.

Key features:

- **Encryption** – Protects data from eavesdropping.
- **Integrity** – Ensures data isn't altered in transit.
- **Authentication** – Verifies the server's identity.

Protocol Flow:

- Browser requests HTTPS connection.
- Server presents its **TLS certificate**.
- Browser verifies certificate authority (CA).
- Encrypted communication begins.

What is TLS?

TLS (Transport Layer Security):

- Successor to SSL (Secure Socket Layer).
- Provides the **cryptographic backbone** for HTTPS.
- Works through **handshake process**:

  1. **Client:** proposes encryption methods.
  2. **Server:** selects method and sends certificate.
  3. **Key exchange:** both sides generate shared secret.
  4. Secure symmetric encryption starts.

Important Concepts:

Asymmetric encryption: Public/private key pair used during handshake.

Symmetric encryption: Faster; used after handshake for data transfer.

Digital certificates: Prove ownership of a domain.

Why HTTPS & TLS Are Useful?

Benefit	Explanation
Security	Prevents sniffing, MITM (man-in-the-middle) attacks
Trust	Browsers mark non-HTTPS sites as “Not Secure”
SEO Boost	Google ranks HTTPS sites higher
Data Protection	Essential for any login, form, or payment data
Modern Standards	APIs, PWAs, and service workers require HTTPS

What is Let’s Encrypt?

- **Let’s Encrypt** is a **free, automated, and open Certificate Authority (CA).**
- Managed by **Internet Security Research Group (ISRG).**
- Issues **domain-validated (DV)** certificates automatically.

Key benefits

- Free of charge
- Automated renewal (via Certbot or API)
- Short validity (90 days) - encourages automation
- Widely trusted by browsers

How Let’s Encrypt Works

- You prove domain ownership via **HTTP or DNS challenge**.
- Let’s Encrypt validates the domain.
- Issues a signed certificate.
- The certificate is installed on your web server.
- Automatic renewal every 90 days keeps it active.
  - Let’s Encrypt makes a request like: (http://yourdomain/.well-known/acme-challenge/XYZ)
  - Your app/server responds with a token proving ownership.

HTTPS & Let’s Encrypt in Our App

To-Do app:
https://todo-app-al36.vercel.app/

Deployed via **Vercel** – which automatically provides HTTPS through Let’s Encrypt.

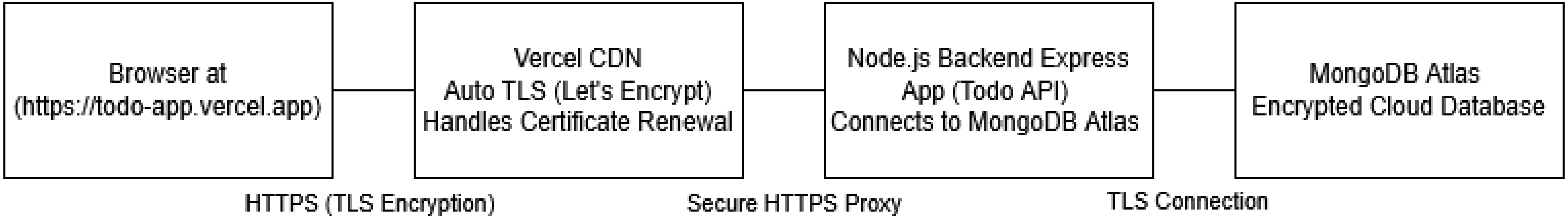
Behind the scenes:

Vercel manages TLS certificates (auto-issued & auto-renewed).

Our Node.js app serves content over HTTPS using those certificates.

Client ↔ Vercel ↔ Node server (encrypted through all layers).

System Architecture Diagram



Browser at
(https://todo-app.vercel.app)

Vercel CDN
Auto TLS (Let's Encrypt)
Handles Certificate Renewal

Node.js Backend Express
App (Todo API)
Connects to MongoDB Atlas

MongoDB Atlas
Encrypted Cloud Database

HTTPS (TLS Encryption)

Secure HTTPS Proxy

TLS Connection

How HTTPS Impacts Our App

Area	Impact
User Trust	Users see secure “lock” icon
Data Security	Todo data encrypted between browser & server
Authentication	Prevents session hijacking
API Integration	Enables secure REST API calls
Compliance	Meets modern web standards

REFERENCES

https://nodejs.org/api/https.html

https://letsencrypt.org

https://vercel.com/blog/automatic-ssl-with-vercel-lets-encrypt

Thank you