HTTPS with Nginx

The Problem

- We want to use HTTPS
  - Encrypts all web traffic
- Prefer not to modify your app code
  - Makes our code more modular and portable
  - Your HTTPS setup can change without changing your app

- Use a reverse proxy

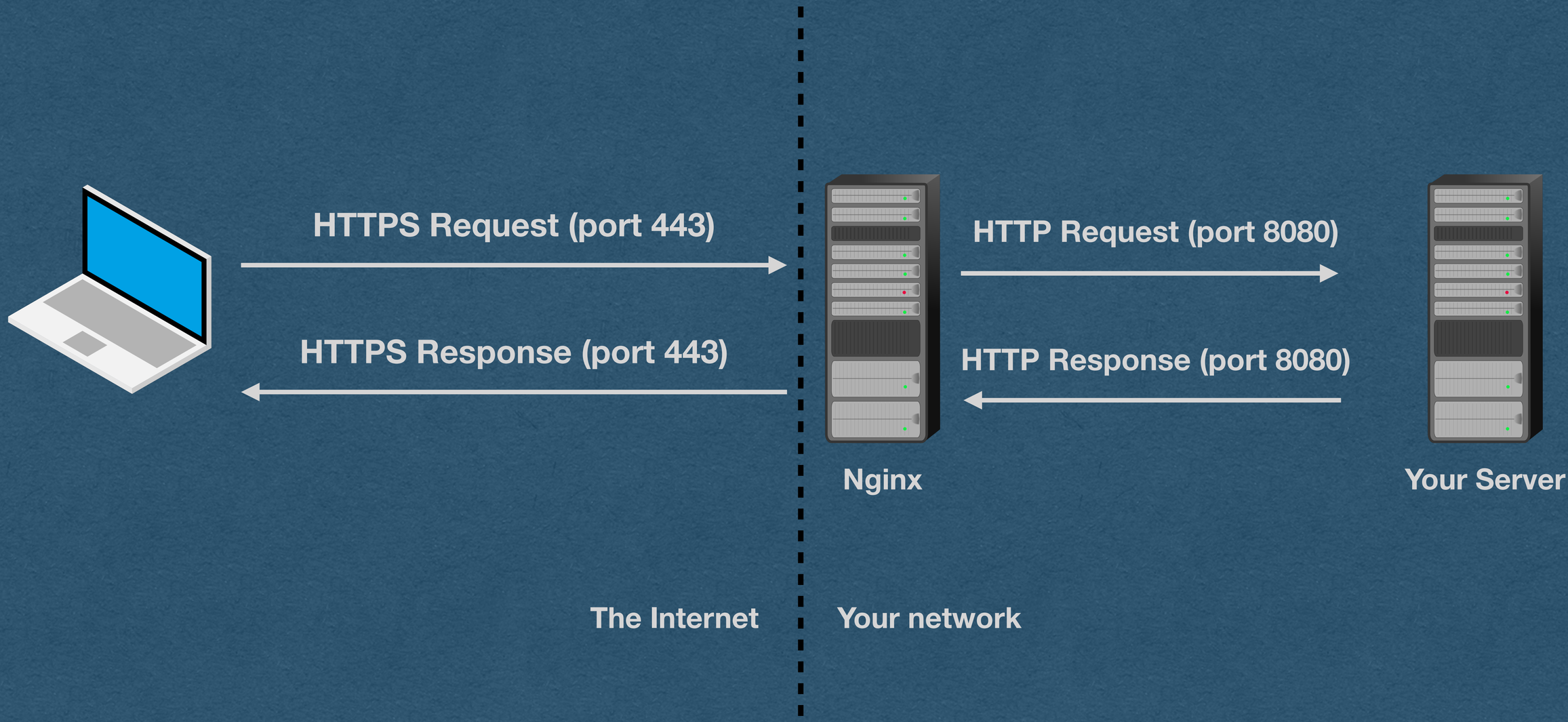
Nginx

- Nginx has many uses and is very powerful software
- We'll use it a reverse proxy with TLS to enable HTTPS

Nginx Strategy

- Use Nginx to listen for HTTPS requests
  - Nginx needs a cert and private key
- Nginx will decrypt requests and forward them to your server as HTTP requests
  - These unencrypted requests are sent over a local network and never touch the Internet
- Your server sends HTTP responses to Nginx
  - Nginx encrypts and sends the HTTPS response over the Internet
- If Nginx gets an HTTP request (port 80) redirect the user to HTTPS (port 443)

Nginx Strategy



OpenSSL

- OpenSSL is a very common SSL/TLS library
  - Written in C
  - Wrappers exist for many languages

- Can be used for many encryption needs
  - Generating keys
  - Signing certs
  - Validating certs

- We'll use OpenSSL in the command line to generate self-signed certificates

Self-Signed Certificate

openssl req -x509 -newkey rsa:4096 -keyout private.key -out cert.pem -days 365 -sha256 -nodes -subj “/C=US"

- This command will generate a self-signed certificate and a private key
  - Private key saves as private.key
  - Certificate saved as cert.pem

Nginx Strategy

Demo