**MERN Stack Web Security**

**# Common website security risks**

* **DDoS Attacks**
* **Malware and viruses**
* **SQL Injection**
* **Spam**
* **Bots**
* **Broken Authentication**
* **Cross-Site Scripting (XSS)**
* **Search engine site blacklists**
* **Registering for a WHOIS domain**
* **Cross-Site Request Forgery (CSRF)**
* **Exploiting inclusion vulnerabilities – LFI and RFI**

# **Production-Ready**

* **Folder Structure**

**Project-root/**   
 ├──> **Controllers**

├──> **Helpers**

├──> **Middleware**

├──> **public**

├──> **Routes**

├──> **Services**

├──> **.Env**

├──> **Package.json**

├──> **Server.js**

* **Environment Variables**
* **Production Environment**
* **Development Environment**
* **Securing the NodeJS Application**
* **Bcrypt-js**
* **Passport-jwt**
* **Helmet**
* **Winston**
* **Express Rate Limiter**
* **Audit-Js**
* **Containerizing your MERN Application**
* **Decide the Containers**
* **Create the Docker files**
* **Create the Docker Compose file**
* **Decide the Containers**
* **Container 1 —** ReactJS
* **Container 2 —** NodeJS with Express
* **Container 3 —** MongoDB

**# Best practices for web security**

* **Maintain Security During Web App Development**
* **Require Injection & Input Validation**
* **Encrypt your data**
* **Use Exception Management**
* **Apply Authentication, Role Management & Access Control.**
* **Don't Forget Hosting/Service-Focused Measures**
* **Avoid Security Misconfigurations**
* **Implement HTTPS (and Redirect All HTTP Traffic to**

**HTTPS).**

* **Include Auditing & Logging**
* **Use Rigorous Quality Assurance and Testing**