CS 9163 Application Security Fall 2012 Team Project
Anthony Candarini, Saurabh Naik, Moran Sidi, Jeffrey Valino

Application Proposal: Patient Medical Profile System

Concept

Medical providers such as emergency medical technicians (EMTs) may be able to improve their care services if they have remote, expedient access to their patient's medical records. The records could be stored in a centralized database, and appropriate security measures may be taken to ensure that patient data is transmitted safely to the medical provider.

Component Details

Databases

There will be two main databases, one for user information and one for patient medical profiles. User information shall include username and password. For simplicity, a patient's medical profile will contain the patient's first/last name, blood type (drop-down list), allergies (free text), ICE contact (In Case of Emergency), and primary doctor (name & phone).

Back-End Authentication and Database Access

This system will authenticate the user and process database requests. Input will be validated via Python reference monitor to an existing API to prevent SQL injection and buffer overflow.

Front-End Website

The website will be publicly accessible via HTTPS. Only registered users may be allowed to login and make requests to the medical profile system. Input will be validated to prevent XSS/HTML injection and buffer overflow. Browser caching will be disabled. A Linux VM has been requested with the address "pmps.poly.edu" for hosting.

System Component	Language	Responsible Teammate*	Additional Tasks Performed
user authentication with hash system, JSON backend handler	Python	Jeffrey	Initial structuring & configuration of SQL Database, phpMyAdmin.
SQL DB request handling and validation reference monitor	Python, MySQLdb (existing open source MySQL API)	Anthony	Installation of all required packages, configuration of Apache and IPTables, VM user administration, UML Dataflow Diagram. Additional tweaking of SQL DB.
Front-End, Website	Front End Mobile Development (HTML, PHP, Javascript)	Saurabh	Architectural Diagram, Initial Implementation of CodeIgniter, Initial Implementation of ZMQ Messaging Platform.
Front-End, Website	SQL, JavaScript, HTML, CSS	Moran	Wrote initial SQL requests, refined SQL database, generated initial data-validation warnings layer, created restricted views per account type, designed client

*Teammates supported each other as needed.

Application Security Elements

- **Security Modes: EMT (read-only access), Doctor (restricted read/write), Admin (unrestricted API access)**
- Multifactor authentication
    - user knows login/password
    - password must meet specific length/difficulty requirements
    - ~~user knows one-time passcode (SecurID)~~
    - ~~device (MAC address) is registered~~
    - login credentials are not cached
    - Inactivity logout implementation
- Medical data is displayed but not stored on device (possible?).
- Code obfuscation to hide private key
- Implementation of “Secure Sessions” (Built into PHP Library)
- Secure transport layer/communication with server (SSL, TLS?)
- Sensitive data is not stored in cookies
- Explore cookie-less session variables within JavaScript