Database Security & Authentication

MainePad Finder

By Ashley Pike

For database security, I have taken two major steps towards protecting the database. This includes the usage of transport layer security (TLS), as well as preventing SQL injection.

Transport Layer Security:
Through creating a local certificate authority, I have generated certificates for the frontend, backend, and database which allows the usage of TLS within all communications in this project. One of its applications is the communication between the backend and the database. This provides us with three key features for the security of our database. First, it encrypts the data ensuring that if communications were to be intercepted, the contents would be unable to be easily read by the attacker. Next, it provides cryptographic hash functions to determine if data has been deliberately tampered with. If data has been determined to be tampered with, the data is then discarded. Finally, it provides authentication by using the certificates of both communicating parties to ensure that they are communicating with the intended parties.

SQL Injection Preventation:
Within every query that the backend sends to the database, I have implemented parameterized queries. This is implemented in both standard queries, by using the placeholder specifier %s, and stored procedures, by passing in variables in a single tuple named parameters. This works by sending the user-provided variables to be inserted separately from the code that is to be executed, causing SQL to treat the variables as plaintext. This prevents malicious actors from executing code that could bypass authentication, exfiltrate sensitive data, or harm the database and its data.