End-to-End Chat Application

supersecurebro.me

CECS 478 - Mehrdad Aliasgari

Super Secure Bros. :
December 11th, 2018
Chris Meyer 014520504
Bryce Moser 012446081

Asset Protection Solution

The user first registers a profile for their account. They use their email and password of their choice after selecting registration. The personalized password is then hashed and salted with bcrypt's hashing algorithm. This hashed password, username, and along with a uniquely generated user ID will be stored in stored together in the database. Upon log-in and a successful registration, a token will be generated for the user based upon their user ID, and this token will be associated with any requests done by that user to guarantee the user's authenticity.

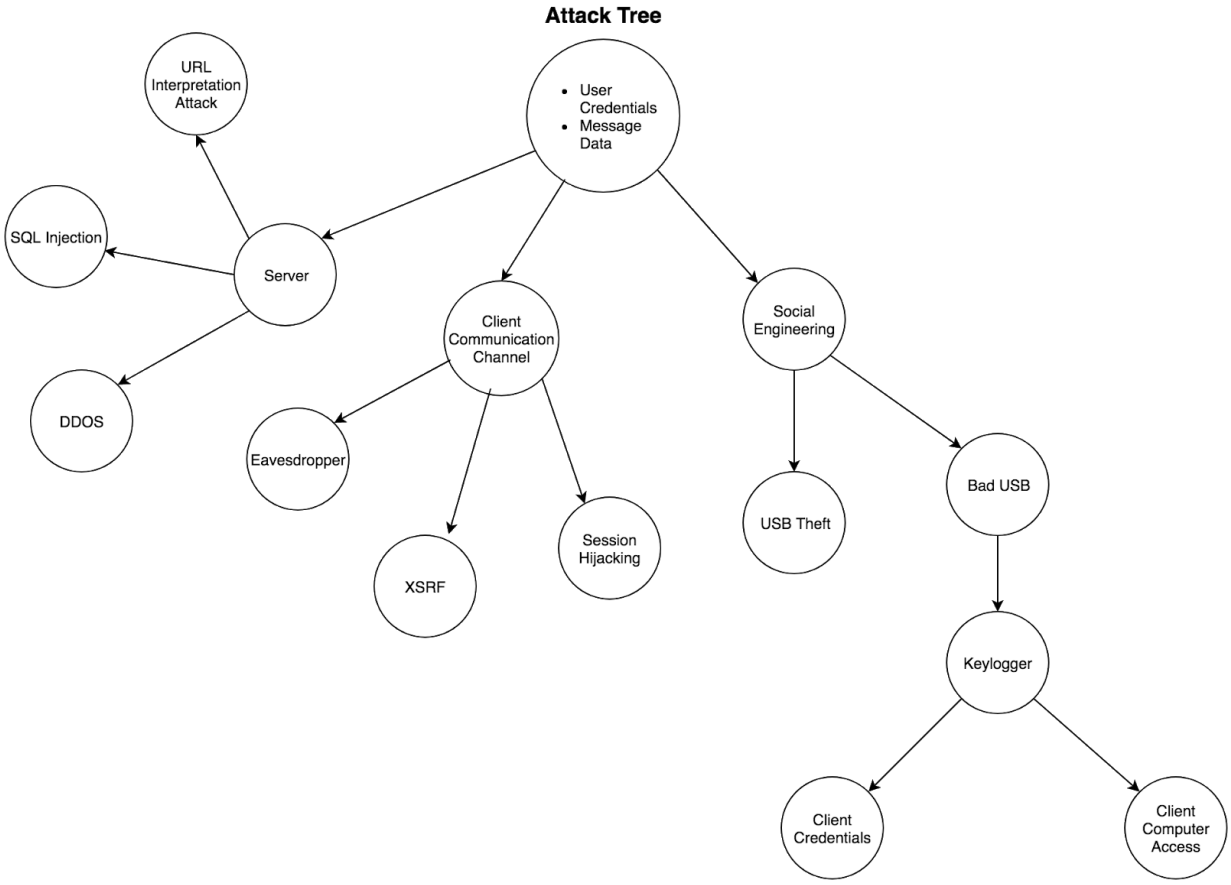
When a user sends a message they will enter both the receiver's username and the content of the message. The plaintext of the message is then immediately encrypted on the users local machine using our own encryption script, before being sent off to the server. The encryption script generates the ciphertext using a predetermined IV and key size. Next, the generated symmetric key is encrypted by an RSA public key using OAEP padding. The entire payload is then sent off to the server under an HTTPS connection.

When the receiver wants to retrieve these messages they will first log in. This will let them generate their own session token. When receiving messages, the system uses the token generated to compare the user ID and ensures the username matches the one on the messages. Any, and all messages, that match both the token and username are returned. We next decrypt the message using the data fields ciphertext, IV, encryptor tag, and the encrypted symmetric key that are stored with each message. Using the private RSA key pair we are able to decrypt the encrypted symmetric key, thus allowing us to decrypt the ciphertext of the message. All messages will then be displayed showing who the message was sent from and the content of the message. After viewing the messages once, they are erased from the database to ensure they cannot be accessed by potential adversaries.

Additional Security

We have gone through additional security measures to ensure sufficient protection of our public and private keys. Instead of keeping these keys on a server, or anywhere else online to download, these keys will be physically exchanged through a USB drive. Keeping the data offline ensures that the keys cannot be stolen from any online database or in transit between the two parties. Any information exchanged online has the potential for threats, like eavesdroppers, to steal this information passively.

Attack Tree Diagram



Attack Tree Analysis

- URL Interpretation Attack: Our system is configured against URL interpretation attacks. All requests must be routed to available API routes to be valid requests, and only specified parameters have any effect in the specified route.
- SQL Injection: Our database used in mongoDB Atlas which is a No SQL Database, so SQL injection attacks cannot occur.
- DDOS: Our webserver is hosted by AWS, who provide defences for small-scale DDOS attacks
- Eavesdropper: Our system is truly end-to-end all eavesdroppers in the communication channels to the server will gain no interpretable information.
- Cross-site Request Forgery: Our server has been setup with an appropriate certificate verifying the server's authenticity. The server is not configured to redirect the client to another server.
- Session Hijacking: Using the npm jsonwebtoken package, our sessions are authenticated with JSON web tokens that are expireable and based on the user's digital signature.
- USB Theft: We have implemented common practice to wipe our USB's upon use to avoid theft.
- Bad USB: It is common practice to only plug in USBs given directly from trusted contacts.
- Keylogger: Common practice for the user to keep their system in a secure environment and to perform common anti-malware scans with the latest software
- Client Credentials: In case a client's credentials are stolen, clients can request for a credential changes.