S2DR (CS 6238) Report

Authors: Vojtech Miksu, Brian Lebiednik

Date: 12/05/2015

```
Our program accomplishes all of the the implementation instructions
as described.
```

For the details about installation, running and testing see README.pdf.

Protocol Details

The server part runs an HTTPS server with the Node.js. When the client initiates the session with the server, the server challenges the client with a certificate request during the TLS handshake. Both the client and the server have 4096 bit RSA keys that they request the CA to sign as part of their initialization. The client part uses openssl for the key generation and the request for the server to sign the certificate. The certificates are stored locally. After the client have a trusted connection, they share a socket for further communication. This connection is persistent until the client terminates the connection or it is timed out due to inactivity.

From the socket the client can call all of the required functions. The messages appear to the server with the hostname, socket, and requested action. If the hostname and key of the client do not match during initialization (or any other request), the server will reject the socket and deny access. The server knows based off the username in the client certificate which active workspace to access as long as the socket is still valid. From this, the server can access the active workspace and store or retrieve files for the client.

The original owner or delegated user has several options during the delegation call. The owner is allowed to set the propagation flag of the file for the other users that the owner delegates access. The owner or delegated user places a max time on the delegation the is_allowed algorithm checks to ensure the delegation remains valid. The communication of the file, client, and permission then are passed to the server. The server checks that file and client exist and that they request is valid.

The server checks all requests against its access control list for the file requested. The rules for delegation first check the owner of the document and then the delegation rules set on the file by the owner or delegated user. The further algorithm can be found in '/server/is_allowed.js'. The server checks the permissions of the file to see if the userId is in any of them or 'ALL' then checks if 'check-in', 'check-out', or 'both' are present and finally checks the time to ensure that the permission is still valid.

The program also uses openssl for the encryption/decryption and integrity/verification of the the documents. When a user checks in a document, he/she can set the flag on the document to 'CONFIDENTIALITY', 'INTEGRITY', 'BOTH' or 'NONE'. When the client sets the flag to 'CONFIDENTIALITY' the server encrypts the file using the openssl library and AES256. When the client sets the 'INTEGRITY' flag, the server captures the SHA256 hash of the file, signs the hash, and stores it. Likewise, it conducts the reverse operations to either decrypt the file or check the integrity of the file to ensure the file has not been modified. The flag 'BOTH' runs both encryption and signing.

Security Analysis

Communication security

The security of the communication between the client and server is based on the security of a 4096 bit RSA key and the use of OpenSSL. OpenSSL conducts the key generation and the encryption, decryption, hashing and signing within the program. The socket communication in the client server communication remains secure with the signing and encrypting of the messages. There are no known attacks to this type of communication that can be accomplished in less than polynomial time. We use build-in Node.js HTTPS server with client authentication that is widely deployed in a real world.

Code security and typos

Throughout the project, the team checked the code with ESLint. ESLint is a powerful linter that inspects JavaScript code and is able to catch many different errors through advanced static analysis. We are using almost 90 different rules that are defined in '.eslintrc'. There are real-time integrations for all major text editors and IDEs.

Unit testing

The main function isAllowed that checks delegations and gives permissions is tested by 17 unit tests.

User inputs and errors

All user inputs are checked in the client app and most importantly on the server (RESTful API) site. The server provides various error codes and messages if anything goes wrong.

Contributions

Vojtech Miksu conducted the coding and optimization of the project. His work was instrumental to accomplishing all of the goals set forth in the requirements document. He provided the configurations for the server and client as well as securing the communication between them.

Brian Lebiednik provided code review and documentation. He worked the product to ensure that it accomplished all of the security reviews. He also authored the report for the project and provided proofing.