Since Facebook Connect is proprietary, a detailed protocol description has not been officially provided. To overcome this problem, we have analyzed all incoming and outgoing HTTP traffic between the browser, the Facebook login server and an application server during the authentication process.

We have translated the protocol in Alice-Bob notation into HLPSL. In particular, we have formally specified the security properties to be verified.

The replay attack can be effectively carried out since the HTTP traffic between the client and the Service Provider is not encrypted: since HTTP (without SSL) is basically a stateless, an intruder can always intercept a packet containing an HTTP request (e.g., using a packet-sniffer like Wireshark), and submit it again.

But also a more serious masquerade attack is possible: an intruder can acquire the HTTP cookies sent together with a legitimate query. Hence, he can send again them to the Service Provider to be authenticated as the Client to obtain illegitimately other resources (e.g., a user’s mailbox, even if it has not been accessed before by the owner).

Again, this protocol has been formalized in HLPSL and formal verification has been carried out using the OFMC model checker. The model checker has found no attacks.

Is Facebook Connect secure?

The authentication flow consists of six HTTP requests-response pairs, one of which uses a secure channel. The data involved in each HTTP request-response transaction are request and response headers, sent and received cookies, query string parameters, POST data and response body. Cookies play a particularly important role, since they carry the information about the login status of the user.

From HTTP traffic analysis we can define a protocol formalization in Alice-Bob notation by abstracting from implementation-level.

The replay attack can be effectively carried out since the HTTP traffic between the client and the Service Provider is not encrypted: since HTTP (without SSL) is basically a stateless, an intruder can always intercept a packet containing an HTTP request (e.g., using a packet-sniffer like Wireshark), and submit it again.

But also a more serious masquerade attack is possible: an intruder can acquire the HTTP cookies sent together with a legitimate query. Hence, he can send again them to the Service Provider to be authenticated as the Client to obtain illegitimately other resources (e.g., a user’s mailbox, even if it has not been accessed before by the owner).

Again, this protocol has been formalized in HLPSL and formal verification has been carried out using the OFMC model checker. The model checker has found no attacks.