Project review

Reviewed team: C

Ondrej Mosnáček, Manoja Kumar Das,
Mmabatho Idah Masemene

PV204 - Security Technologies

May 19, 2015

The reviewed project

- *Tomcat / Web Browser Based Authentication using JavaCard*
- **Original goal:**
  - to provide web-based authentication using JavaCard
- **What was actually implemented:**
  - a server application that checks if the client sent the string “Auth” and sends back the string “Authorized”
  - a client Java browser applet that sends the string “Auth” to the server
  - a JavaCard applet for password-based authentication with a very crude PC application
  - the browser applet does not communicate with the card at all

Design

- server and card share a master password and counter
- client downloads a Java applet from the server
- the browser applet retrieves SHA-1(Password XOR Counter) from the card (and the counter is incremented in the card)
- the browser applet sends the hash to the server
- the server computes the expected hash and increments the counter
- the server checks if the recieved hash matches the expected one

Design flaws I

1. **the crypto**
   - SHA-1(Password XOR Counter)
   - non-standard construct (possibly prone to cryptanalysis)
   - “Never design your own crypto!”
   - SHA-1 is not considered secure nowadays (but this is addressed in the documentation)
   - a proper password-based KDF should be used instead (or at least HMAC)
2. **no secure channel between client and server**
   - design does not mandate the use of TLS for client-server communication
   - an MITM attacker could hijack the authenticated session after the hash has been sent

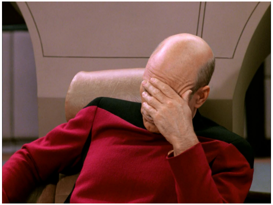
Design flaws II

3. **using a Java browser applet**
   - Java browser plugins are now deprecated[1]
   - historically suffered from many vulnerabilites
   - in this case, the applet requires unlimited permissions – security problem if server gets compromised

[1] https://blogs.oracle.com/java-platform-group/entry/moving_to_a_plugin_free

Implementation flaws

- **JavaCard applet vulnerability**
  - access to JavaCard is protected by user PIN
  - the PIN is set using `INS_SETPIN` APDU instruction
  - this instruction is not authenticated (the old PIN is not required)
  - attacker can just set the PIN to a new value and authenticate to the card
  - after authentication, the attacker can pre-generate a sequence of valid hashes



Summary

- original goals not achieved
- several flaws in the design
- one serious flaw in the implementation
- most code was copied from the Internet/study materials; only small parts were modified
- some useless functionality was left over from the original code
    - AES encryption and RSA signing in the JavaCard applet
    - a file containing arbitrary sentences to be returned by the server