`libkovanserial` Security

Braden McDorman

June 5, 2013

1 Introduction

`libkovanserial` is the communication protocol for the KIPR Link robotics controller. `libkovanserial` provides a unified protocol for both USB and network communication. `libkovanserial` is divided into three layers:

1. "Transmitters"[1] are back-ends that implement a specific communication mechanism, such as TCP/IP sockets or USB comm ports. Security is not handled on this layer.
2. The "Transport Layer"[2] handles packet creation, checksumming, and a basic ACK/resend mechanism for non-reliable protocols such as serial communication ports. Session-level security is defined on this layer.
3. The "Protocol Layer"[3] helps facilitate protocol-level communication with the KIPR Link. This is intentionally left as a somewhat leaky abstraction. User password security is implemented on this layer.

What follows is a description of the new security system for `libkovanserial` over any transmitter. The security system is designed to maximize backwards compatibility with older clients and servers.

2 User Passwords

User passwords can theoretically be any length, but for the sake of simplicity we limit users to a maximum of 10 characters. This password is never sent as plain text over a transmitter. User passwords can be entered from the KIPR Link's UI.

The user password is used to generate the two following values:

[1] `https://github.com/kipr/libkovanserial/blob/master/include/kovanserial/transmitter.hpp`

[2] `https://github.com/kipr/libkovanserial/blob/master/include/kovanserial/transport_layer.hpp`

[3] `https://github.com/kipr/libkovanserial/blob/master/include/kovanserial/kovan_serial.hpp`

- `md5(password)` – The cryptographically secure MD5 hash of the password (128 bit digest).
- `sha1(password)` – The cryptographically secure SHA1 hash of the password (128 bit digest).

3 Authentication Handshake

`libkovanserial` uses XOR encryption with a mutual shared session key that is negotiated during handshake. Since the session key is a pseudo-randomly generated 512 bits and sessions are short lived, cracking the session key is impossible. `libkovanserial` also goes a step further and fills empty space in packets with pseudo-random bytes. This prevents sniffers from detecting the key using zeroed-out sections of packets. This handshake method guards against man-in-the-middle attacks and other sniffers by XOR encrypting the session key during transmission using a private but mutual piece of information: `sha1(password)`. Since the session key is 512 pseudo-random bits and the SHA1 key is a cryptographically strong hash, decoding either piece of information is impossible.

3.1 Handshake Example

A typical handshake looks like this:

1. Ask the server if it requires authentication. If no, `finish`. If yes, `goto 2`.
2. Send the server our password's MD5 hash.
3. Check if authentication was successful. If no, prompt user for new password and `goto 2`. If yes, decrypt the session key using our password's SHA1 hash and `finish`.

Once the decrypted session key is obtained, all packets are XOR encrypted with it.

4 Backwards Compatibility

This new authentication mechanism **will break** backwards compatibility with older `libkovanserial` servers and clients. We can, however, detect some of the cases and report errors and recommended courses of actions.

4.1 Out-of-date Servers (Old KIPR Link firmwares)

KIPR Links prior to 1.9.7 have no notion of authentication, and will thus silently ignore authentication packets and accept any incoming data. Since newer clients expect the `ConfirmAuthentication` command to be sent in response to a `RequestAuthentication` command, the authentication mechanism

will patiently wait for several seconds and return a timeout error. Since we know that this timeout error probably means an old KIPR Link firmware, we can display an error message requesting them to upgrade their KIPR Link's firmware.

4.2 Out-of-date Clients (Old versions of KISS IDE)

Older versions of KISS IDE will not understand the authentication mechanism, and will thusly be unable to sign their outgoing packets with a valid session password. The recommended course of action is to not handle this case, and specify in the release notes that KISS IDE must be upgraded to work with newer KIPR Link firmwares. We could also display a warning message on the link, but this is a non-ideal solution since `kovan-serial` (the server process on the Link) currently has no notion of the UI.