

An idea for an integrated password management system.

BV

[2015-01-24 Sat 17:53]

1 What there is

There are two leading password management (PM) paradigms that I am aware of:

browser remote data stores for in-browser use. Includes LastPass and support built in to major browsers (eg Chrome and Firefox)

application local data stores for use by an OS-native application. Includes KeePassX, Revelation

2 What is the problem?

Browser based PM relies on some other server to securely store the, hopefully encrypted, data. Some systems such as Firefox's can be run on your own servers, others like LastPass rely on some degree of 3rd party trust. Browser PM is not easily used for non-browser authentication.

Application based PM usually has local stores or relies on external distribution mechanisms (btsync, dropbox, git). Besides the trust needed, they tend to have troubles synchronizing when two copies of the same store diverge. They may have browser integration or may not and require autotype or cut-buffer for communicating with other applications.

Then there are plain text password stores such as `~/.authinfo` or `~/.my.cnf` required by some "venerable" applications. These rely on file system security which can be defeated if an attacker gains access to the file system (discarded disks, backups, local exploit, etc).

3 The idea.

Enough rambling. The idea is to construct an integrated password management system which bridges multiple systems. Its design is one of many pieces working together.

3.1 The bus

The pieces (nodes) are joined by a bus which should ride on network or local sockets.

Need to think about the communication topology: hub vs. distributed and discovery.

Obviously requires secure communication protocol between nodes. Auth/auth and encrypted communications. Don't reinvent.

3.2 Nodes

These nodes are needed:

LastPass various Python interfaces are available

KeePassX mine or newer, better ones exist

text files use named pipes backed by a node (a'la `socat`)

browser plugins if we want to divest from proprietary ones