Mt Wilson Blueprint

tpm-tools

# Background

“tpm-tools” is an open source project hosted at SourceForge as part of the Trousers project and it provides a set of command line tools for Linux for interacting with the TPM.

When we started working with tpm-tools, and still as of 2015, we discovered many of the tools have limitations such as they only support the “well-known” SRK password or they accept a password on the command line (insecure) or they only accept a password via keyboard input (impedes automation). Furthermore, the toolkit is incomplete and doesn’t provide all functionality supported by the TPM. This is not derogatory - IBM did a great service to the community by sponsoring the Trousers and tpm-tools projects.

This blueprint describes how we extend tpm-tools to provide functionality that we need for our solutions and how we’re going to contribute that functionality back to the tpm-tools project to share with the community.

# Architecture

This blueprint discusses two types of contributions: patches and additions.

A patch is a change we make to an existing tpm-tool.

An addition is a new tool that we write to expand the toolkit.

An option modifier is an option that does not take any parameters itself because its purpose is to alter the interpretation of another option or another option’s parameter.

## Patches

### Hex password

This patch adds the capability to accept a password in hex format. Wherever possible, we name this option “-x”, and it is an option modifier for interpreting other options that accept a password parameter.

This feature is useful in situations where the password contains nonprintable characters that are challenging to process in an interactive or shell scripting environment.

It interprets the argument to the password parameter as a hex representation of the password instead of the password itself, so the hex representation is decoded to obtain the intended password.

The value of this feature is enabling the user to easily expand the password search space to the entire 2^20 bit space without having to manage non-printable characters in scripts.

For example, if a given tool already has an option “-p” for accepting a password, and normal usage would be “-p intel” then with the hex password patch the user could instead write “-x -p 696e74656c” and this would be equivalent.

### Password in environment

This patch adds the capability to accept a password from an environment variable instead of the command line. Wherever possible, we name this option “-t”, and it is an option modifier for interpreting other options that accept a password parameter.

This feature is useful in situations where a password must be provided to a tool but it is not desirable for that password to be exposed in the process list. Because environment variables are not exposed by the process list, they provide more confidentiality when providing passwords to programs.

The value of this feature is enabling the user to provide a password without exposing it in the command line.

For example, if a given tool already has an option “-p” for accepting a password, and normal usage would be “-p intel” then with the password in environment patch the user could instead write “-t -p PASSWORD” and the tool would then look for an environment variable named “PASSWORD” and use its value as the password. The process list therefore shows only “-t -p PASSWORD” which does not leak the password itself.

### tpm\_nvread

This patch adds two include files to the source of tpm\_nvread.c which are required in order to compile it on our systems:

#include <sys/types.h>

#include <sys/stat.h>

## Additions

### tpm\_bindaeskey

Binds an AES key using an existing TPM key by encrypting the AES key using the public key portion of the TPM key (called the binding public key).

Command line parameters:

-i <file> Specifies input file which is the AES key to wrap (16 bytes for AES-128)

-k <file> Specifies the RSA public key file to use for wrapping the AES key

-o <file> Specifies output file which is the wrapped AES key

### tpm\_createkey

Creates a new RSA, 2048-bit, volatile, non-migratable, TPM key of the specified type (binding or signing).

Command line parameters:

-b, --binding Create a binding key (private key used for decryption)

-s, --signing Create a signing key (private key used for encryption)

-k, --keyout Output filename in which to store the private key blob

-p, --pubout Output filename in which to store the public key

-q, --keypassword Password to protect the new TPM key

-Q, --keypasswordsha1 Modifier, indicates argument to -q is already a SHA-1, use as-is

-t, --env Modifier, indicates argument to -q is an environment variable name and not a literal password

-x, --hex Modifier, indicates argument to -q is a hex-encoded password which must be hex-decoded before use

### tpm\_signdata

Signs an arbitrary SHA-1 hash using a TPM signing key.

Command line parameters:

-i, --infile Input filename from which to read the SHA-1 hash to sign

-k, --keyfile Input filename from which to load the signing private key blob

-o, --outfile Output filename in which to store the RSA signature

-q, --keypassword Password to protect the new TPM key

-Q, --keypasswordsha1 Modifier, indicates argument to -q is already a SHA-1, use as-is

-t, --env Modifier, indicates argument to -q is an environment variable name and not a literal password

-x, --hex Modifier, indicates argument to -q is a hex-encoded password which must be hex-decoded before use

-N, --NIARL Load keys using NIARL style

### tpm\_unbindaeskey

Unbinds an AES key using an existing TPM key by decrypting the encrypted AES key using the private key portion of the TPM key (called the binding private key).

Command line parameters:

-i <file> Specifies input file which is the AES key to unwrap (16 bytes for AES-128)

-k <file> Specifies the RSA private key blob file to use for unwrapping the AES key

-o <file> Specifies output file which is the unwrapped AES key; if not specified the unwrapped AES key bytes will be written to stdout

-q, --keypassword Password to access the TPM binding private key

-Q, --keypasswordsha1 Modifier, indicates argument to -q is already a SHA-1, use as-is

-t, --env Modifier, indicates argument to -q is an environment variable name and not a literal password

-x, --hex Modifier, indicates argument to -q is a hex-encoded password which must be hex-decoded before use

# Contribution

The tpm-tools maintainers have requested that we submit patches for one feature at a time. So patch for supporting -x option must be submitted separately from patch supporting -t option.