Digital Signature Verification API

Author: Dmitry Kasatkin
Date: 06.10.2011

Introduction

Digital signature verification API provides a method to verify digital signature. Currently digital signatures are used by the IMA/EVM integrity protection subsystem.

Digital signature verification is implemented using cut-down kernel port of GnuPG multi-precision integers (MPI) library. The kernel port provides memory allocation errors handling, has been refactored according to kernel coding style, and checkpatch.pl reported errors and warnings have been fixed.

Public key and signature consist of header and MPIs:

```
struct pubkey_hdr {
        uint8_t         version;        /* key format version */
        time_t          timestamp;      /* key made, always 0 for now */
        uint8_t         algo;
        uint8_t         nmpi;
        char            mpi[0];
} __packed;

struct signature_hdr {
        uint8_t         version;        /* signature format version */
        time_t          timestamp;      /* signature made */
        uint8_t         algo;
        uint8_t         hash;
        uint8_t         keyid[8];
        uint8_t         nmpi;
        char            mpi[0];
} __packed;
```

keyid equals to SHA1[12-19] over the total key content. Signature header is used as an input to generate a signature. Such approach insures that key or signature header could not be changed. It protects timestamp from been changed and can be used for rollback protection.

API

API currently includes only 1 function:

```
digsig_verify() - digital signature verification with public key


/**
* digsig_verify() - digital signature verification with public key
* @keyring:     keyring to search key in
* @sig: digital signature
* @sigen:       length of the signature
* @data:        data
* @datalen:     length of the data
* @return:      0 on success, -EINVAL otherwise
*
* Verifies data integrity against digital signature.
* Currently only RSA is supported.
* Normally hash of the content is used as a data for this function.
*
*/
int digsig_verify(struct key *keyring, const char *sig, int siglen,
                  const char *data, int datalen);
```

User-space utilities

The signing and key management utilities evm-utils provide functionality to generate signatures, to load keys into the kernel keyring. Keys can be in PEM or converted to the kernel format. When the key is added to the kernel keyring, the keyid defines the name of the key: 5D2B05FC633EE3E8 in the example bellow.

Here is example output of the keyctl utility:

```
$ keyctl show
Session Keyring
-3 --alswrv      0     0  keyring: _ses
603976250 --alswrv      0    -1   \_ keyring: _uid.0
```

```
817777377 --alswrv      0     0       \_ user: kmk
891974900 --alswrv      0     0       \_ encrypted: evm-key
170323636 --alswrv      0     0       \_ keyring: _module
548221616 --alswrv      0     0       \_ keyring: _ima
128198054 --alswrv      0     0       \_ keyring: _evm

$ keyctl list 128198054
1 key in keyring:
620789745: --alswrv     0     0 user: 5D2B05FC633EE3E8
```