USB key building for chipsec and secureboot checks

`A Help to build your own ChipSec and SecureBoot USB keys`

The created USB key can boot in either of the following modes:

1. a live Debian distribution to launch **ChipSec** on the computer to analyze
2. a tool to import your own trust keys and to check importation, as well as **SecureBoot** keys to import,

Linux Tools to install before generating the USB keys

```
sudo apt-get install debootstrap
sudo apt-get install sbsigntool
sudo apt-get install efitools
```

Tool to build the USB key: create-chipsec.sh

> **Note:** Some sub scripts require access to sudo commands.

The script supports writing to a block device (`/dev/sdc` for example) or a standard file, which can be later copied to a USB drive using `dd`.

Plug a new USB key (attached on /dev/sdc in this case).

```
./create-chipsec.sh /dev/sdc
```

Unplug the USB key.

Test the system

Plug the key, start the computer and pick one of the two boot modes.

Live Debian distribution

1. boot on the USB key, then at the bootloader prompt start "Debian GUN/Linux"
2. when finished booting, login as root (no password)
3. if you need an alternate keyboard, use eg. `loadkeys fr`.
4. from the root terminal, launch ChipSec with `chipsec_main.py`.
5. alternately you can run the dump__system.sh script which will also gather information about the machine (hardware present, firmware versions et.c)

EFI Tool

1. Go the BIOS/Firmware configuration and set the platform to SecureBoot enabled and reset to Setup Mode.
2. Boot on USB key, then at the bootloader prompt either:

   - start “Keytool” directly, which lets you execute binaries from its menu,
   - OR start “EFI Shell” and launch EFI binaries from EFI shell (eg. launch `EFI/keytool/KeyTool.efi` to import trust keys).
3. From the EFI shell, you can identify the USB key letter storing the binaries with commmands “fs0:” or “fs1:” or fsX: ... then “dir”. Within Keytool, the drives will have names instead. The USB key should be named “ESP”, but you can confirm it by browsing its content through the “Execute Binary” menu.
4. With Keytool, import the following files from the `EFI/keys` folder (in that order): `DBX.esl`, `DB.esl`, `KEK.esl` and `PK.auth`. Importing the PK will set the platform to User mode.
5. Restart the platform:
   - the shell should run since it’s signed with trust anchor to the PK;
   - `HelloWorld.efi` should not run since it’s unsigned.