**SGX Installation Manual**

There are two ways to implement SGX.

* SGX with Graphene
* SGX without Graphene

Note: We adopt the methodology to implement SGX with Graphene.

**System specification:**

1. Ubuntu 20.04.2 LTS
2. Kernel version 5.11
3. Intel core i7-8700 CPU @ 3.20 \*12
4. SGX must Enables in BIOS

**Installation:**

There are few steps needed before implementing SGX.

1. **Install Dependencies.**

We use following commands to install all dependencies including (Bison, Gawk Meson, Python 3, JinJa 2 etc).

1. sudo apt-get install -y autoconf bison build-essential gawk meson **\** python3 python3-click python3-jinja2 wget
2. sudo apt-get install -y libunwind8 python3-pyelftools python3-pytest
3. **SGX required Packages.**

To install that we use following commands.

1. sudo apt-get install -y libcurl4-openssl-dev libprotobuf-c-dev **\**protobuf-c-compiler python3-pip python3-protobuf
2. python3 -m pip install toml>=0.10
3. **Upgrade Kernal in Ubuntu to support SGX** (if less that kernel 5.9 must upgrade while there is no need for 5.11 or higher).

For that Upgrade to the Linux kernel patched with FSGSBASE. Use following guide.

1. <https://itsfoss.com/upgrade-linux-kernel-ubuntu/>
2. **Install the Intel SGX driver**

In latest Kernel like 5.11 or higher SGX already install otherwise install using following steps for ubuntu.

1. To check if matching kernel headers are installed:
   1. $ dpkg-query -s linux-headers-$(uname -r)
2. To install matching headers:
   1. $ sudo apt-get install linux-headers-$(uname -r)
3. Build the Intel(R) SGX Driver: To build Intel(R) SGX driver, change the directory to the driver path and enter the following command:
   1. $ make
   2. $ sudo mkdir -p "/lib/modules/"`uname -r`"/kernel/drivers/intel/sgx"
   3. $ sudo cp isgx.ko "/lib/modules/"`uname -r`"/kernel/drivers/intel/sgx"
   4. $ sudo sh -c "cat /etc/modules | grep -Fxq isgx || echo isgx >> /etc/modules"
   5. $ sudo /sbin/depmod
   6. $ sudo /sbin/modprobe isgx
4. **Install Intel SGX SDK/PSW**
   1. Use the following command(s) to install the required tools to build the Intel(R) SGX SDK.
5. $ sudo apt-get install build-essential ocaml ocamlbuild automake autoconf libtool wget python-is-python3 libssl-dev git cmake perl
   1. To install the additional required tools:
      1. $ sudo apt-get install libssl-dev libcurl4-openssl-dev protobuf-compiler libprotobuf-dev debhelper cmake reprepro unzip
   2. Install SDK using following link.
      1. https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions/get-started.html
   3. Download the source code and prepare the submodules and prebuilt binaries:
      1. $ git clone https://github.com/intel/linux-sgx.git
      2. $ cd linux-sgx && make preparation
   4. Copy the mitigation tools corresponding to current OS distribution from external/toolset/{current\_distr} to /usr/local/bin and make sure they have executed permission:
      1. $ sudo cp external/toolset/{current\_distr}/{as,ld,ld.gold,objdump} /usr/local/bin
      2. $ which as ld ld.gold objdump
   5. Build the Intel(R) SGX SDK and Intel(R) SGX SDK Installer.
      1. make sdk
      2. $ make sdk USE\_OPT\_LIBS=0
      3. $ make sdk DEBUG=1
      4. $ make sdk\_install\_pkg
   6. Build the Intel(R) SGX PSW and Intel(R) SGX PSW Installer
      1. make psw
      2. $ cd psw/ae/le
      3. $ make
      4. make deb\_psw\_pkg
   7. Install the Intel(R) SGX SDK
      1. sudo apt-get install build-essential python
      2. $ cd linux/installer/bin
      3. $ ./sgx\_linux\_x64\_sdk\_${version}.bin
      4. NOTE: You need to set up the needed environment variables before compiling your code. To do so, run:

$ source ${sgx-sdk-install-path}/environment

* 1. Install the Intel(R) SGX PSW
     1. $ sudo apt-get install libssl-dev libcurl4-openssl-dev libprotobuf-dev
     2. Launch Service use this command: apt-get install libsgx-launch libsgx-urts
     3. For EPID-based attestation service use this command: apt-get install libsgx-epid libsgx-urts
     4. For algorithm agnostic attestation service use this command: apt-get install libsgx-quote-ex libsgx-urts
     5. For ECDSA-based service use this command: apt-get install libsgx-dcap-ql
  2. To start the service AESMD use following command:
     1. $ sudo service aesmd start

1. **Generate signing keys:** 
   1. You can either place the generated enclave key in the default path, Pal/src/host/Linux-SGX/signer/enclave-key.pem, or specify the key’s location through the environment variable SGX\_SIGNER\_KEY.
   2. For signing use following command: openssl genrsa -3 -out enclave-key.pem 3072
2. **Building SGX Graphene:**
   1. To build Graphene, in the root directory of Graphene repo, run the following commands:
      1. # if you build graphene-direct (note that "direct" means non-SGX version) make
      2. # if you build graphene-sgx make SGX=1 ISGX\_DRIVER\_PATH=<path-to-sgx-driver-sources>
   2. Then install Graphene (recall that “direct” means non-SGX version):
      1. meson setup build/ --buildtype=release -Ddirect=enabled -Dsgx=enabled
      2. ninja -C build/
      3. sudo ninja -C build/ install
   3. **Installation of Graphene (Note: On Ubuntu 18.04 or 20.04 (for 18.04, in intel-sgx.list, replace focal with bionic)**
   4. sudo curl -fsSLo /usr/share/keyrings/graphene-keyring.gpg https://packages.grapheneproject.io/graphene-keyring.gpg
   5. echo 'deb [arch=amd64 signed-by=/usr/share/keyrings/graphene-keyring.gpg] https://packages.grapheneproject.io/ stable main' | sudo tee /etc/apt/sources.list.d/graphene.list
   6. curl -fsSL https://download.01.org/intel-sgx/sgx\_repo/ubuntu/intel-sgx-deb.key | sudo apt-key add -

echo 'deb [arch=amd64] https://download.01.org/intel-sgx/sgx\_repo/ubuntu focal main' | sudo tee /etc/apt/sources.list.d/intel-sgx.list

* 1. # (if you're on Ubuntu 18.04, remember to write "bionic" instead of "focal")
  2. sudo apt-get update
  3. sudo apt-get install graphene # for 5.11+ upstream, in-kernel driver
  4. sudo apt-get install graphene-oot # for out-of-tree SDK driver
  5. sudo apt-get install graphene-dcap # for out-of-tree DCAP driver

1. **Prepare a signing key**

* Only for SGX, and if you haven’t already:
* graphene-sgx-gen-private-key
* This command generates an RSA 3072 key suitable for signing SGX enclaves and stores it in HOME/.config/graphene/enclave-key.pem. This key needs to be protected and should not be disclosed to anyone.

1. **Run sample application**

* Core Graphene repository contains several sample applications. Thus, to test Graphene installation, we clone the Graphene repo:

git clone --depth 1 https://github.com/grapheneproject/graphene.git v1.2

* We don’t want to build Graphene (it is already installed on the system). Instead, we want to build and run the HelloWorld example. To build the HelloWorld application, we need the gcc compiler and the make build system:
* sudo apt-get install gcc make # for Ubuntu distribution
* sudo dnf install gcc make # for RHEL-8-like distribution

1. **Go to the HelloWorld example directory:**

cd graphene/CI-Examples/helloworld

**Build and run without SGX:**

make

graphene-direct helloworld

**Build and run with SGX:**

make SGX=1

graphene-sgx helloworld