# SGX加密性测试

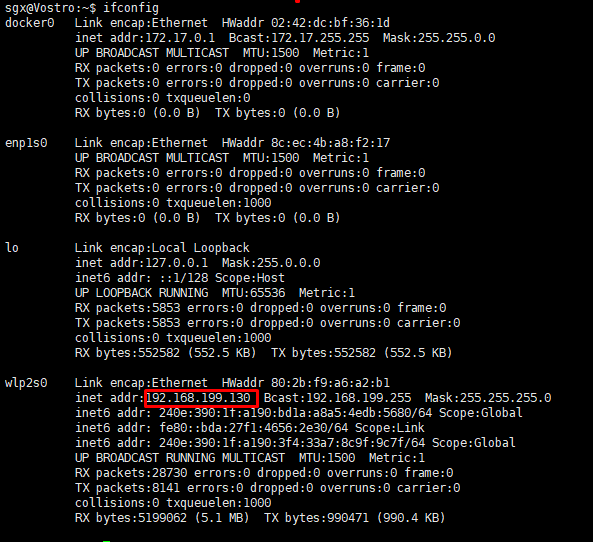
## 一、实验目的

测试SGX的加密性：确认SGX确认是可以做到加密的。

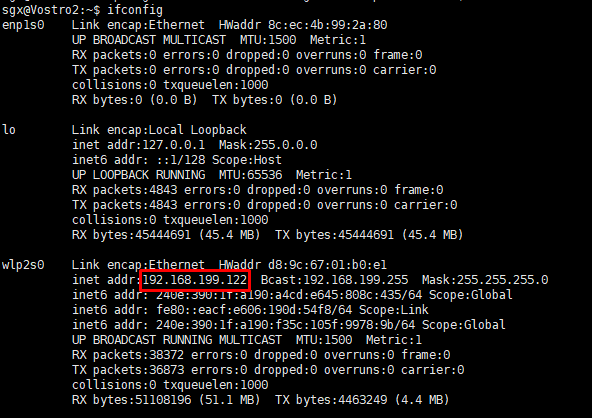
## 二、实验环境

两台相同型号的SGX机器，内存、CPU配置都一致

机器A



机器B



## 三、实验方法

1. 在一台机器上，使用SGX写一个文件，存入硬盘上

2. 使用编辑器打开文件，不可以识别里面的内容

3. 将文件拷贝到另一台机器，使用SGX打开文件，无法识别里面的内容。

## 四、实验步骤

### 1. 在机器A上，编写程序，创建文件并写入数据

#### 1）在Enclave里面定义2个函数：write\_file和 read\_file。如下。

|  |
| --- |
| #[no\_mangle] |
|  | pub extern "C" fn write\_file() -> i32 { |
|  |  |
|  | let rand = sgx\_rand::random::<RandData>(); |
|  |  |
|  | let helper = SerializeHelper::new(); |
|  | let data = match helper.encode(rand) { |
|  | Some(d) => d, |
|  | None => { |
|  | println!("encode data failed."); |
|  | return 1; |
|  | }, |
|  | }; |
|  |  |
|  | let mut file = match SgxFile::create("sgx\_file") { |
|  | Ok(f) => f, |
|  | Err(\_) => { |
|  | println!("SgxFile::create failed."); |
|  | return 2; |
|  | }, |
|  | }; |
|  |  |
|  | let write\_size = match file.write(data.as\_slice()) { |
|  | Ok(len) => len, |
|  | Err(\_) => { |
|  | println!("SgxFile::write failed."); |
|  | return 3; |
|  | }, |
|  | }; |
|  |  |
|  | println!("write file success, write size: {}, {:?}.", write\_size, rand); |
|  | 0 |
|  | } |
|  |  |
|  | #[no\_mangle] |
|  | pub extern "C" fn read\_file() -> i32 { |
|  |  |
|  | let mut data = [0\_u8; 10]; |
|  |  |
|  | let mut file = match SgxFile::open("sgx\_file") { |
|  | Ok(f) => f, |
|  | Err(\_) => { |
|  | println!("SgxFile::open failed."); |
|  | return 1; |
|  | }, |
|  | }; |
|  |  |
|  | let read\_size = match file.read(&mut data) { |
|  | Ok(len) => len, |
|  | Err(\_) => { |
|  | println!("SgxFile::read failed."); |
|  | return 2; |
|  | }, |
|  | }; |
|  |  |
|  | let helper = DeSerializeHelper::<RandData>::new(data.to\_vec()); |
|  | let rand = match helper.decode() { |
|  | Some(d) => d, |
|  | None => { |
|  | println!("decode data failed."); |
|  | return 3; |
|  | }, |
|  | }; |
|  |  |
|  | println!("read file success, read size: {}, {:?}.", read\_size, rand); |
|  | 0 |
|  | } |

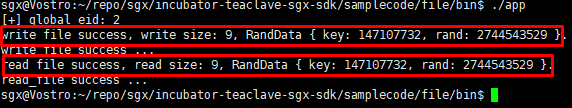
write\_file的步骤是：生成一个随机数->创建名为“sgx\_file”的文件-> 将数据写入进去 –>打印

open\_file的步骤是：打开“sgx\_file”文件->将里面的数据取出->打印

#### 2）在app中调用这两个接口，如下

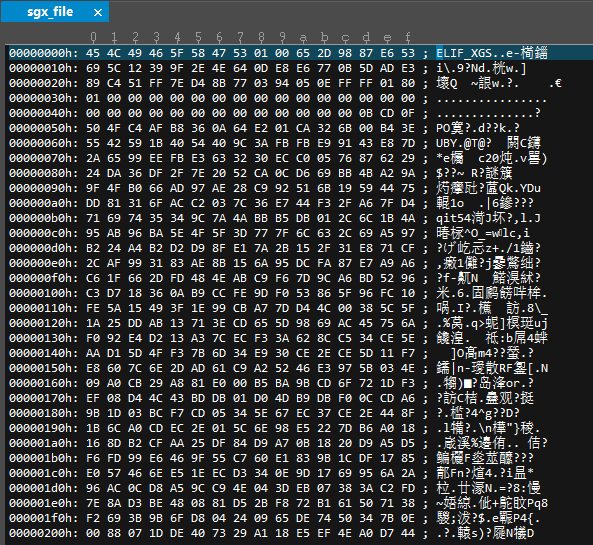
|  |
| --- |
| sgx\_ret = write\_file(global\_eid, &ret); |
|  | if(sgx\_ret != SGX\_SUCCESS) { |
|  | print\_error\_message(sgx\_ret); |
|  | return -1; |
|  | } |
|  | printf("write\_file success ...\n"); |
|  |  |
|  | sgx\_ret = read\_file(global\_eid, &ret); |
|  | if(sgx\_ret != SGX\_SUCCESS) { |
|  | print\_error\_message(sgx\_ret); |
|  | return -1; |
|  | } |

#### 3）打印结果如下



### 2. 用编辑器打开文件

用ultraedit打开，结果乱码，如下



### 3. 在机器B上尝试打开机器A的文件

1）编写程序

Enclave里面的程序不用改变，还是那2个接口；

App里面只调用read\_file接口，而不调用write\_file接口。代码如下

|  |
| --- |
| sgx\_ret=read\_file(global\_eid,&ret); |
|  | if(sgx\_ret != SGX\_SUCCESS) { |
|  | print\_error\_message(sgx\_ret); |
|  | return -1; |
|  | } |

2）打印结果如下

