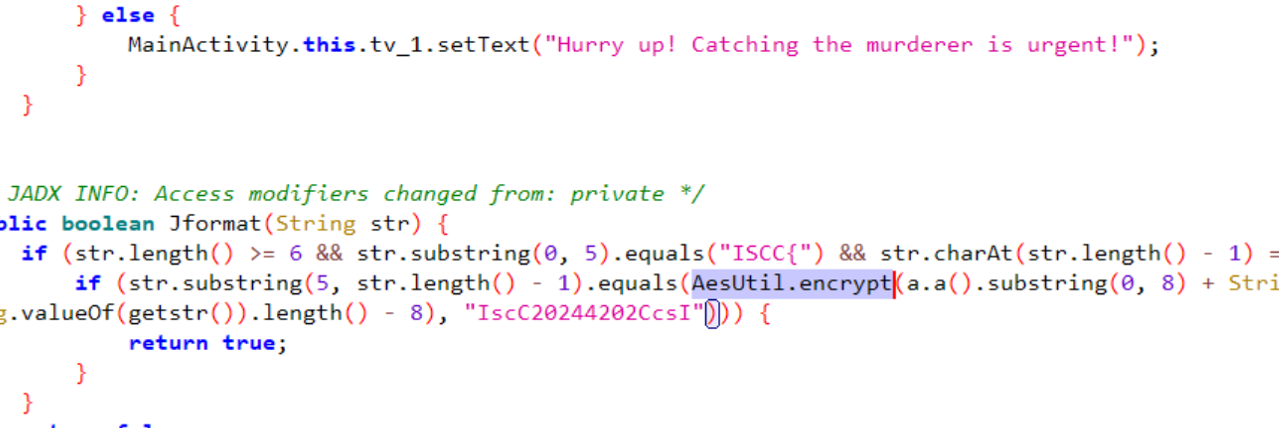
# ISCC2024 WriteUp

Whooops+李卓航+925798691@qq.com

### Mobile+ohHELP



使用Frida设置GetKey返回值：

let Myjni = Java.use("com.example.ohhelp.MyJNI.Myjni");

Myjni["GetKey"].implementation = function () {

console.log('GetKey is called');

let ret = this.GetKey();

console.log('GetKey ret value is ' + ret);

return "PUDzbflthjqxlJVW";

};

反射调用com.example.ohhelp.getstr.generateRandomString时发现校验property会导致程序退出。使用Frida设置System.getProperty返回值：

let System = Java.use('java.lang.System');

System.getProperty.overload('java.lang.String').implementation = function (propertyName) {

var returnValue = this.getProperty(propertyName);

console.log("System.getProperty called with propertyName: " + propertyName + ", returned: " + returnValue);

if (propertyName === "java.vm.vendor") {

return "";

}

return returnValue;

};

之后hook AesUtil.encrypt就不会退出了。

GetTime函数返回当前时间戳，根据题目描述找到正确的时间戳：

使用Frida设置GetTime返回值：

let Myjni = Java.use("com.example.ohhelp.MyJNI.Myjni");

Myjni["GetTime"].overload().implementation = function () {

console.log('GetTime is called');

let ret = this.GetTime();

console.log('GetTime ret value is ' + ret);

return "1055853128000";

};

之后hook AesUtil.encrypt得到的返回值即为flag。

最终的Frida hook脚本：

function hook() {

Java.perform(function () {

let Myjni = Java.use("com.example.ohhelp.MyJNI.Myjni");

Myjni["GetTime"].overload().implementation = function () {

console.log('GetTime is called');

let ret = this.GetTime();

console.log('GetTime ret value is ' + ret);

return "1055853128000";

};

Myjni["GetKey"].implementation = function () {

console.log('GetKey is called');

let ret = this.GetKey();

console.log('GetKey ret value is ' + ret);

return "PUDzbflthjqxlJVW";

};

let AesUtil = Java.use("com.example.ohhelp.AesUtil");

AesUtil["encrypt"].overload('java.lang.String', 'java.lang.String').implementation = function (str, str2) {

console.log('encrypt is called, str: ' + str + ', str2: ' + str2);

let ret = this.encrypt(str, str2);

console.log('encrypt ret value is ' + ret);

return ret;

};

let System = Java.use('java.lang.System');

System.getProperty.overload('java.lang.String').implementation = function (propertyName) {

var returnValue = this.getProperty(propertyName);

console.log("System.getProperty called with propertyName: " + propertyName + ", returned: " + returnValue);

if (propertyName === "java.vm.vendor") {

return "";

}

return returnValue;

};

});

}

hook();