



LINUX REVERSE ENGINEERING

چکیده

در نظر است بتوان ورودی، خروجی و کارکرد برنامه های Native
- کتابخانه ها و فایل های اجرایی - را در زمان اجرا و بدون تغییر
در فایل های آنها دستکاری و مورد پایش قرار داد



فهرست

2	تحویل دادنی های پروژه.....
2	دو برنامه Native – بنام های APP-N1 و APP-N2.....
2	یک کتابخانه Native (.so) – بنام Lib-N.....
3	یک برنامه Java Console Application – بنام App-J.....
4	یک برنامه C Console Application بنام App-C.....
4	دو برنامه نفوذی (بخش اصلی پروژه).....
5	کلیه ابزارها، روش ها، مستندات، Source Code ها و منابع جهت تولید موارد قبل.....
5	مستند حاوی توضیح و راهنمای استفاده از موارد ذکر شده در مورد قبل.....
5	نکات مهم.....
5	مشخصات محیط اجرای پروژه.....
5	سیستم عامل.....
5	Java.....
5	نکته مهم.....
5	نحوه همکاری.....



تحويل دادنی های پروژه

دو برنامه Native – بنام های APP-N1 و App-N2

زبان برنامه نویسی: C

دارای یک تابع به نام Fun-A که پردازش تمام آرگومان های ورودی و کارکرد اصلی برنامه در آن انجام میشود:

پارامترهای ورودی برنامه ها حداقل شامل نوع داده زیر:

- a. Param-N1: عدد
- b. Param-N2: کارکتر
- c. Param-N3: رشته (string)

کارکرد برنامه ها:

- a. App-N1
 - i. Console Output: یک رشته (string) شامل عبارت Hello World و تمام آرگومان های ورودی.
 - ii. کارکرد: ایجاد یک فایل XML استاندارد به نام N1.xml حاوی داده ایی از نوع عدد (int) به نام N1INT با مقدار پارامتر Param-N1.

- b. App-N2
 - i. Console Output: یک عدد ساده حاصل جمع پارامتر Param-N1 و مقدار عددی پارامتر Param-N2 و طول رشته پارامتر Param-N3.
 - ii. کارکرد: ایجاد یک فایل XML استاندارد به نام N2.xml حاوی داده ایی از نوع رشته (string) به نام N2STRING با مقداری شامل عبارت Thank You پیوسته شده (Concat) به پارامتر Param-N3.

Exit Code ها:

- a. App-N1: 0
- b. App-N1: 1

یک کتابخانه Native (.so) – بنام Lib-N

زبان برنامه نویسی: C

دارای 3 تابع با نام های زیر:

- a. Fun-L1
- b. Fun-L2
- c. Fun-L3

با مشخصات زیر:

a. ورودی های شامل حداقل نوع داده های زیر:

- i. ساده مانند int و char
- ii. رشته متنی (char *)



iii. Struct با ساختاری مانند struct B بصورت زیر:

```
struct A
{
    int id;
    char name[20];
    int lenght;
};

struct B
{
    char name[25];
    char number[10];
    struct A a;
};

struct C
{
    struct B b;
    struct A a;
};
```

b. خروجی با نوع داده های زیر:

- i. Fun-L1: مقدار عددی B.a.id
- ii. Fun-L2: یک رشته حاوی مقدار تمام اعضای Struct های A و B
- iii. Fun-L3: یک Struct با ساختاری مانند struct C بصورت بالا با محتویات Struct های ورودی B و A

c. کارکرد: تولید یک فایل XML استاندارد با محتویاتی شامل موارد زیر:

- i. Fun-L1: مقدار عددی B.a.id
- ii. Fun-L2: یک رشته حاوی مقدار تمام اعضای Struct های A و B
- iii. Fun-L3: یک Struct با ساختاری مانند struct C بصورت بالا با محتویات Struct های ورودی B و A

یک برنامه Java Console Application – بنام App-J

پارامترهای ورودی: Param-J1 و Param-J2 از نوع عدد با عملکرد های زیر است:

a. Param-J1 = 1: اجرای برنامه App-N1 بصورت زیر:

param3 A 123 ./App-N

b. Param-J1 = 2: اجرای برنامه App-N2 مانند مورد 1

c. Param-J1 = 3: فراخوانی توابع کتابخانه Lib-N.so بصورت زیر:

i. Param-J2 = 1: تابع Fun-L1

ii. Param-J2 = 2: تابع Fun-L2

iii. Param-J2 = 3: تابع Fun-L3



یک برنامه Console Application بنام App-C

کارکرد: دقیقاً مانند برنامه App-I.

دو برنامه نفوذی (بخش اصلی پروژه)

1. دو برنامه **App-R1** و **App-R2** در حوزه (Dynamic instrumentation for reverse-engineers, and security researchs) در رابطه با عملکرد Runtime برنامه ها و کتابخانه Native بالا با مشخصات زیر:
 - a. برنامه ها و کتابخانه های اشاره شده در بخش های قبلی صرفاً جهت تسریع در تولید بخش جاری (دو برنامه نفوذی) می باشد و معیار پذیرش پروژه و تسویه حساب منوط به فرآورده های همین بخش و کار کردن آن هم با فرآورده های بخش های قبلی است و هم برنامه ها (Native و Java) و کتابخانه هایی (Native) است که کارفرما ارایه میدهد و ورودی و خروجی های متفاوتی دارند.
 - b. هیچ گونه تغییر در فایل های Source Code و یا Binary برنامه های App-N1 و App-N2 مجاز نمی باشد.
 - c. کنترلی بر نحوه اجرای برنامه های Native و Java نداریم.
 - d. بر اساس هر گونه آگاهی از Source Code برنامه های App-N1 و App-N2 نباشند.
 - e. قابلیت اجرا بصورت دستی قبل یا پس از اجرای برنامه های Native و Java این پروژه.
 - f. بدون اخلال در پایداری محیط اجرا و سایر برنامه ها.
2. برنامه **App-R1**: دستکاری (Manipulation) در عملکرد Runtime برنامه ها و کتابخانه Native بالا بصورت زیر:
 - a. برنامه های App-N1 و App-N2:
 - i. تغییر مقادیر آرگومان ها ورودی.
 - ii. تغییر مقادیری که قرار است در فایل XML نوشته شوند.
 - iii. تغییر Exit Code ها.
 - b. توابع کتابخانه Lib-N:
 - i. تغییر پارامتر های ورودی.
 - ii. تغییر مقدار خروجی توابع.
 - iii. فایل XML:
 1. تغییر مقادیری قبل از نوشته شدن در فایل.
 2. افزودن element به XML.
 3. حذف element از XML.
 - c. Struct ها (در کلیه موارد قبلی):
 - i. تغییر اعضای با نوع ساده.
 - ii. تغییر اعضای با نوع داده Struct (تو در تو تا هر چند Level).
3. برنامه **App-R2**: پایش (Monitoring) عملکرد برنامه ها و کتابخانه بالا در Runtime بصورت زیر:
 - a. محل ارایه خروجی:
 - i. نمایش اطلاعات بر روی Console.
 - ii. ثبت در لحظه اطلاعات در فایل Log: به ازای هر بار اجرا، یک فایل Log با نام حاوی تاریخ و زمان تقویمی با دقت ثانیه ایجاد گردد.
 - b. اطلاعات مورد نیاز پایش:
 - i. عمومی



1. هر رکورد اطلاعاتی در یک خط باشد.
2. تاریخ و زمان تقویمی با دقت ثانیه به ازای هر ردیف اطلاعات.
- ii. برنامه های App-N1 و App-N2
 1. مسیر برنامه اجرایی استفاده کننده از آنها.
 2. مقادیر پارامترهای ورودی.
 3. مسیر فایل XML.
 4. Exit Code ها.
- iii. توابع کتابخانه Lib-N :
 1. مسیر برنامه اجرایی استفاده کننده از Lib-N.
 2. نام تابع.
 3. پارامترهای ورودی.
 4. نام و مقدار خروجی توابع.
 5. مسیر فایل XML.

کلیه ابزارها، روش ها، مستندات، Source Code ها و منابع جهت تولید موارد قبل
مستند حاوی توضیح و راهنمای استفاده از موارد ذکر شده در مورد قبل
نکات مهم

امکان استفاده از محصول ها / روش های Open Source: الزامی

امکان استفاده از نرم افزارهای تجاری: خیر

مشخصات محیط اجرای پروژه

سیستم عامل

Distribution: CentOS 7.3 x86_64

Kernel: 3.10.x86_64

Compiler: gcc 4.8

Java

Version: Java 1.8 x86_64

نکته مهم

امکان Upgrade / Downgrade موارد بالا: خیر

نحوه همکاری

- عقد قرارداد و امضای NDA.
- در صورت افزایش نیازمندی ها تا 25٪ امکان افزایش زمان و هزینه به تناسب وجود دارد.