



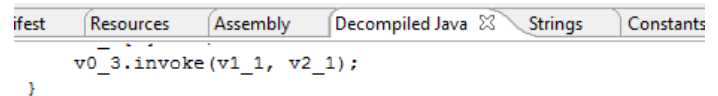
Интерактивный декомпилятор для Android

Приложение JEB - мощный декомпилятор для Android, разработанный для профессионалов по безопасности. Приложение позволяет декомпилировать и анализировать APK-файлы, тем самым серьезно экономя время ваших инженеров.

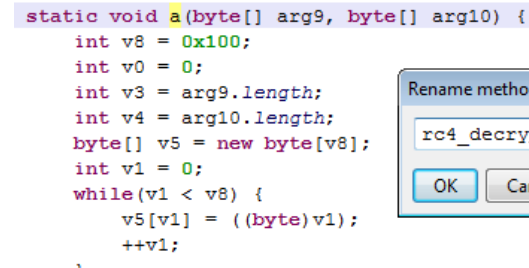
```
class TraceDisplayInformation {  
    private CoordinatesE6 center;  
    private int zoom;  
  
    public TraceDisplayInformation(CoordinatesE6 arg:  
        super();  
        int v1 = arg4.get_lng();  
        int v2 = arg4.get_lat();  
        this.center = new CoordinatesE6(v1, v2);  
        this.zoom = arg5;  
    }  
  
    public CoordinatesE6 get_center() {  
        return this.center;  
    }  
}
```

Мощный

Уникальная особенность приложения JEB - возможность декомпиляции байт-кода Dalvik в исходный код Java без дополнительной конвертации dex в jar. Наш декомпилятор принимает во внимание тонкости формата Dalvik и учитывает метаданные, содержащиеся в DEX-файле.



```
ifest Resources Assembly Decompiled Java Strings Constants  
v0_3.invoke(v1_1, v2_1);  
}
```

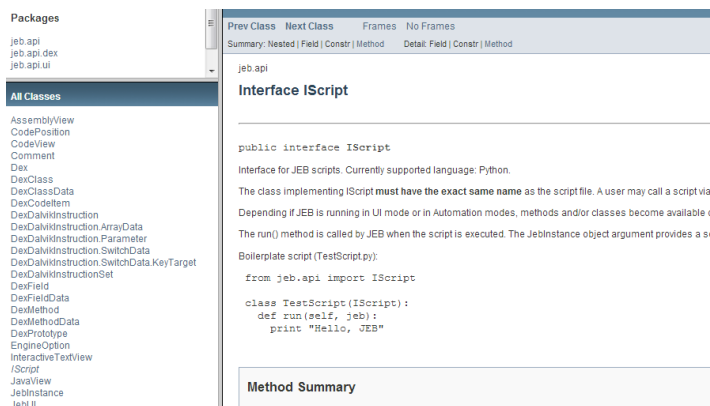


```
static void a(byte[] arg9, byte[] arg10) {  
    int v8 = 0x100;  
    int v0 = 0;  
    int v3 = arg9.length;  
    int v4 = arg10.length;  
    byte[] v5 = new byte[v8];  
    int v1 = 0;  
    while (v1 < v8) {  
        v5[v1] = ((byte)v1);  
        ++v1;  
    }  
}
```

Rename method dialog box: rc4_decrypt

Гибкий

Аналитикам необходимы гибкие инструменты, особенно при работе с запутанными или защищенными фрагментами кода. Мощный интерфейс JEB позволяет изучать перекрестные ссылки, переименовывать методы, поля, классы, перемещаться между кодом и данными, делать заметки, добавлять комментарии и т.д.



Расширяемый

Возможно подключение к JEB дополнительных плагинов и скриптов через программный интерфейс приложения (API). Пример: доступ к AST декомпилированного Java кода для удаления запутанных слоев; Использование неинтерактивного режима JEB для автоматизации серверной обработки.

API поддерживает следующие языки: Python, Java.



Интерактивный декомпилятор для Android

Два важных преимущества JEB по сравнению с аналогичными инструментами других производителей: интерактивность и гибкость, а также результат декомпиляции профессионального уровня. Все это позволит инженерам анализировать и постепенно понимать сложные фрагменты кода.

```
public class Crypto
{
    public static void rc4_crypt(byte[] paramArrayOfByte1, byte[] paramArray
    {
        int i = paramArrayOfByte1.length;
        int j = paramArrayOfByte2.length;
        byte[] arrayOfByte = new byte[256];
        int k = 0;
        int m;
        int n;
        label130: int i2;
        int i3;
        if (k >= 256)
        {
            m = 0;
            n = 0;
            if (n < 256)
                break label168;
            i2 = 0;
            i3 = 0;
        }
        for (int i4 = 0; ; i4++)
        {
            if (i4 >= j)
            {
                return;
                arrayOfByte[k] = ((byte)k);
                k++;
                break;
            }
        }
    }
}
```

Результат работы стороннего декомпилятора Java (слева)

- Статический код без интерактивных элементов
- Ошибки декомпиляции (отмечены стрелками)
- Результат неудобен для чтения и использования

Результат работы JEB (справа), после анализа кода специалистом.

Код методов аккуратно структурирован и удобочитаем.

Больше примеров на нашем сайте.

```
public static void rc4_crypt(byte[] key, byte[] data) {
    int v10 = 0x100;
    int keylen = key.length;
    int datalen = data.length;
    byte[] sbox = new byte[v10];
    int i = 0;
    while(i < v10) {
        sbox[i] = ((byte)i);
        ++i;
    }

    int k = 0;
    i = 0;
    while(i < v10) {
        k = (sbox[i] + k + key[i % keylen]) % 0x100 & 0xFF;
        byte v7 = sbox[i];
        sbox[i] = sbox[k];
        sbox[k] = v7;
        ++i;
    }

    i = 0;
    k = 0;
    int j = 0;
    while(j < datalen) {
        i = (i + 1) % 0x100 & 0xFF;
        k = (sbox[i] + k) % 0x100 & 0xFF;
        v7 = sbox[i];
        sbox[i] = sbox[k];
        sbox[k] = v7;
        data[j] = ((byte)(data[j] ^ sbox[(sbox[i] + sbox[k]) % 0x100 & 0xFF]));
        ++j;
    }
}
```

На сайте приложения вы можете заказать демо-версию и узнать стоимость лицензирования.