CÁTEDRA DE CIBERSEGURIDAD CIBERUGR, INCIBE-UGR

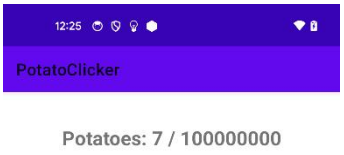
Nombre	PotatoClicker
Categoría	Reversing
Dificultad	Difícil
Puntos	500

DESCRIPCIÓN DEL RETO

Me he enganchado a un juego de darle click a una patata. Llevo ya unas horas jugando, pero no consigo ganar :(. ¿Me ayudas?

WRITEUP

Se nos da una aplicación de Android. Si la abrimos, nos encontraremos con un juego de hacer click. Cada vez que hacemos click en la patata, el contador sube uno. El problema es que, para llegar a la cantidad propuesta, se necesitarían muchos clicks.



Iniciamos JADX e importamos el archivo PotatoClicker.apk. Si nos vamos a la actividad principal “MainActivity”, se puede observar que implementación de la funcionalidad de click se realiza desde una biblioteca nativa.

```
public final class MainActivity extends AppCompatActivity {
    private String androidId;
    private ActivityMainBinding binding;
    private int count = -1;
    private int max = -1;

    public final native void clickPotato(String userId);

    public final native String getFlag(String userId);

    public final native int getMaxPotatoes(String userId);

    public final native int getPotatoCount(String userId);
```

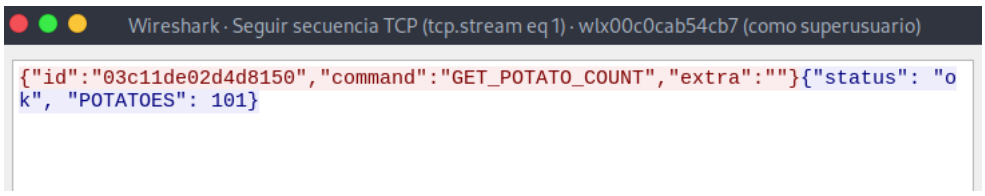
Se implementan las siguientes funciones:

- clickPotato(): Podemos suponer que es la función que le añade uno al contador de clicks.
- getMaxPotatoes(): Podemos suponer que es la cantidad máxima de patatas que aparece en la aplicación.
- getPotatoCount(): Podemos suponer que es la cantidad de clicks que llevamos.
- getFlag(): Podemos suponer que devuelve la flag cuando llegamos a la cantidad de patatas esperadas.

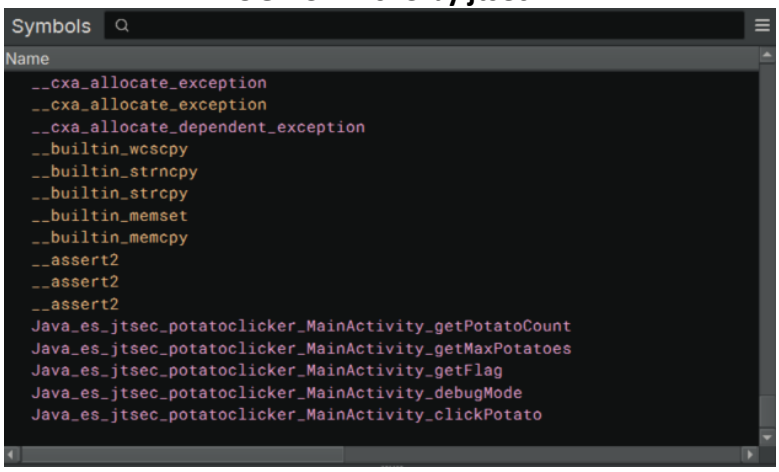
A todas estas funciones se le pasa como parámetro un “userId”. Este identificador se corresponde al identificador único de cada dispositivo Android.

```
setContentView(inflate.getRoot());
String string = Settings.Secure.getString(getContentResolver(), "android_id");
Intrinsics.checkNotNullExpressionValue(string, "getString(...)");
this.androidId = string;
updatePotatoCount();
ActivityMainBinding activityMainBinding2 = this.binding;
```

Como sabemos que las comunicaciones van por internet, podemos intentar analizar el tráfico. Para ello podemos utilizar Wireshark o un proxy como MitmProxy. En este caso usaremos Wireshark. Observamos como se crea un túnel TCP donde se manda por JSON la información sin cifrar.



Podemos ahora mirar cómo se implementa la lógica de conteo de patatas en la parte nativa. Para ello, en ese caso, usaremos Binary Ninja. Analizaremos la biblioteca “libpotatoclicker.so”. Una vez analizada por Binary Ninja, podremos observar las funciones que se implementan. Destaca la función “debugMode”.



Miraremos como ejemplo la función “getPotatoCount”. Dicha función le pasa por parámetro a “jsonRequest” la clave json “GET_POTATO_COUNT” vista anteriormente.

```
uint64_t Java_es_jtsec_potatoclicker_MainActivity_getPotatoCount(int64_t* arg1, int64_t arg2, int64_t arg3)
00024f98        if (x0_1 >= 0x17)
00024f98        {
00024fb4            int64_t x25_1 = x0_1 | 0xf;
00024fbc            void* x0_3 = operator new(x25_1 + 1);
00024fc4            x23_1 = x0_3;
00024fc8            size_t var_80_1 = x0_1;
00024fc8            var_78 = x0_3;
00024fcc            var_88 = x25_1 + 2;
00024fdc            memmove(x23_1, x0, x0_1);
00024f98        }
00024f98        else
00024f98        {
00024fa4            x23_1 = &var_88 | 1;
00024fa8            var_88 = (int8_t)(x0_1 << 1);
00024fa8
00024fac            if (x0_1)
00024fdc                memmove(x23_1, x0, x0_1);
00024f98        }
00024f98
00024ff0        *(uint8_t*)((char*)x23_1 + x0_1) = 0;
00024ff4        char var_a0;
00024ff4        __builtin_strncpy(&var_a0, " GET_POTATO_COUNT", 0x12);
00025000        int16_t var_b8 = 0;
00025014        sendJsonRequest(&var_88, &var_a0, &var_b8);
```

La función “sendJsonRequest” se estructura de la siguiente manera

```
00050350    int64_t sendJsonRequest(std::__ndk1::basic_string<char, std::__ndk1::char_traits<char>, std::__ndk1::allocator<char> > const& arg1,
00050350      std::__ndk1::basic_string<char, std::__ndk1::char_traits<char>, std::__ndk1::allocator<char> > const& arg2,
00050350      std::__ndk1::basic_string<char, std::__ndk1::char_traits<char>, std::__ndk1::allocator<char> > const& arg3)

00050350    {
00050350        /* tailcall */
00050350        return sendJsonRequest(arg1, arg2, arg3);
00050350    }
```

Donde si seguimos el flujo de ejecución averiguaremos que el primer argumento será el identificador de usuario, el segundo, el tipo de comando y el tercero, el argumento extra. Dicho argumento solo se usa en el comando DEBUG_MODE_SET_POTATOES. Este parámetro es un entero que se convierte a una string. Por lo tanto, podemos suponer que puede ser la cantidad de patatas que queremos asignar.

```
*(uint8_t*)((char*)x24_1 + x0_1) = 0;
void* x0_5;
int128_t v1_1;
x0_5 = operator new(0x20);
v1_1 = data_15250;
__builtin_strncpy(x0_5, "DEBUG_MODE_SET_POTATOES", 0x17);
int128_t var_a0 = v1_1;
*(uint8_t*)((char*)x0_5 + 0x17) = 0;
std::__ndk1::to_string(arg4);
char var_b8;
sendJsonRequest(&var_88, &var_a0, &var_b8);
```

A partir de esta información podemos implementar un script que, con el identificador de Android setee las patatas de dicho dispositivo. Para ello, simplemente crearemos un socket TCP que mande un comando “DEBUG_MODE_SET_POTATOES” al servidor junto con el número de patatas máximo que vemos en la aplicación. Después, obtendremos la respuesta y la mostraremos por terminal. En caso de que todo haya salido bien deberemos ver un mensaje de que las patatas han sido modificadas.

```python
import socket
import json

SERVER_IP = "IP"
SERVER_PORT = 6666

def send_debug_request(client_id, potato_count):
    """Sends a DEBUG_MODE command to the server to set a specific potato count."""
    request_data = {
        "id": client_id,
        "command": "DEBUG_MODE_SET_POTATOES",
        "extra": str(potato_count)
    }

    try:
        # Create socket and connect
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as client_socket:
            client_socket.connect((SERVER_IP, SERVER_PORT))

            # Send request
            request_json = json.dumps(request_data)
            client_socket.sendall(request_json.encode())

            # Receive response
            response_data = client_socket.recv(1024).decode()
            response = json.loads(response_data)

            print(f"Server response: {response}")

    except Exception as e:
        print(f"Error: {e}")

if __name__ == "__main__":
    client_id = "03c11de02d4d8150"
    potato_count = 100000000
    send_debug_request(client_id, potato_count)
```

Si ejecutamos el script y accedemos a la app, podremos ver que se muestra la flag.


2:10
PotatoClicker
Potatoes: 100000001 / 100000000
Flag: UGR_ETSIIT_CTF25{I_L1k3_POt4tO
3s_Yummy}