Міністерство освіти і науки України

Національний технічний університет «ХПІ»

Навчально-науковий інститут комп’ютерних наук та інформаційних технологій

Кафедра комп’ютерної інженерії та програмування

**ЗВІТ**

з лабораторної роботи № 5

з дисципліни «Сучасні технології безпечного програмування»

**«Дослідження динамічних бібліотек»**

Виконав:

студент гр. КН-Н922б

Зозуля В.Д.

Перевірив:

доцент кафедри

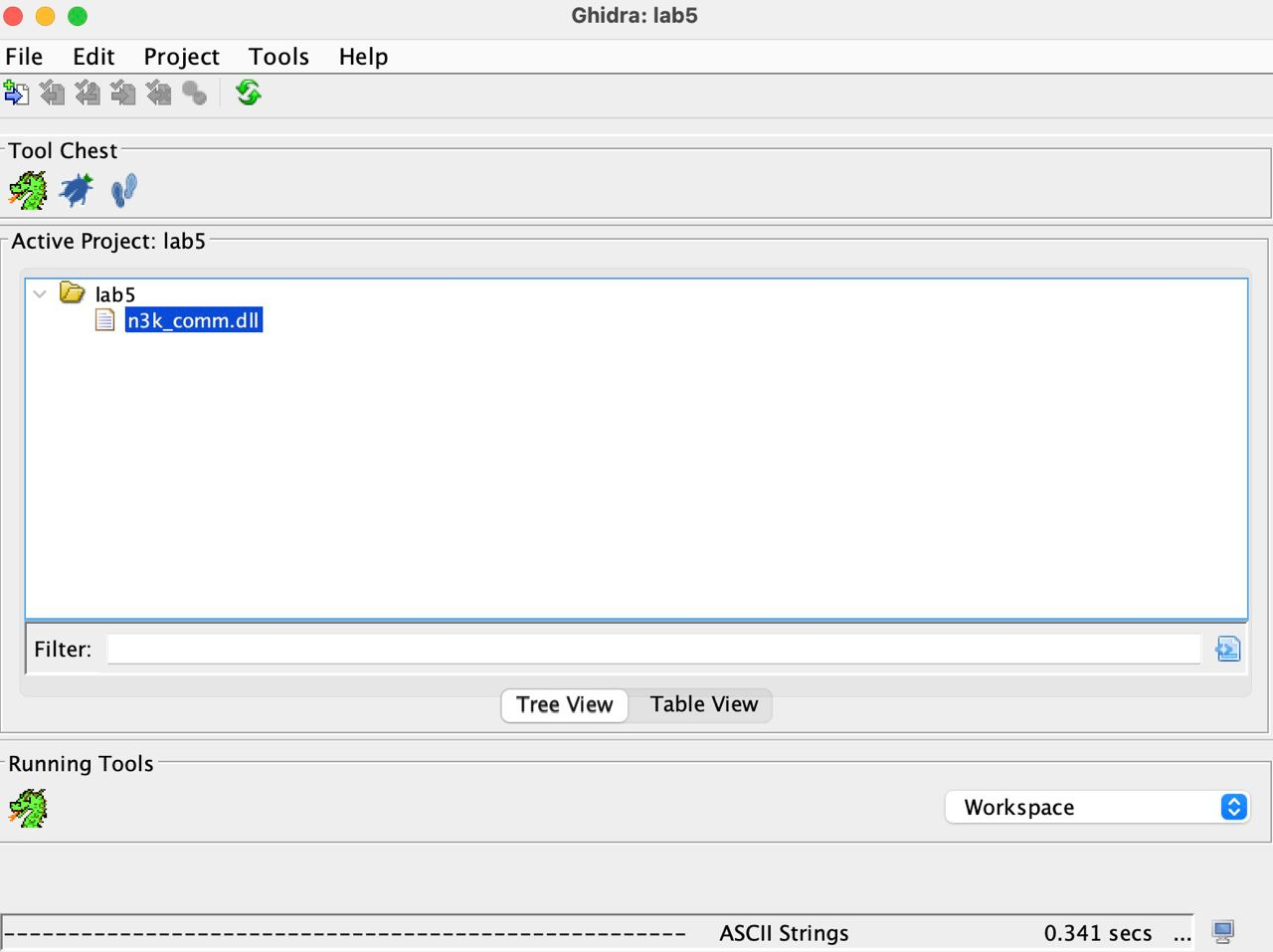
Бульба С.С.

Харків – 2022

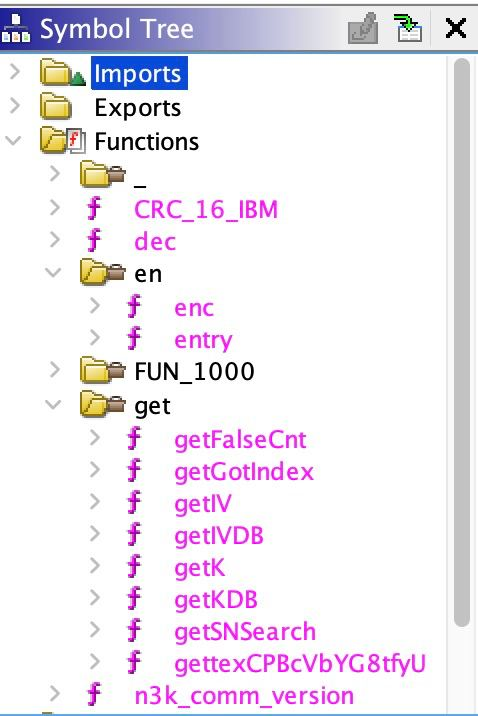
**Мета роботи**: Отримати навички основ реверс-інжинірингу.

**Хід роботи**

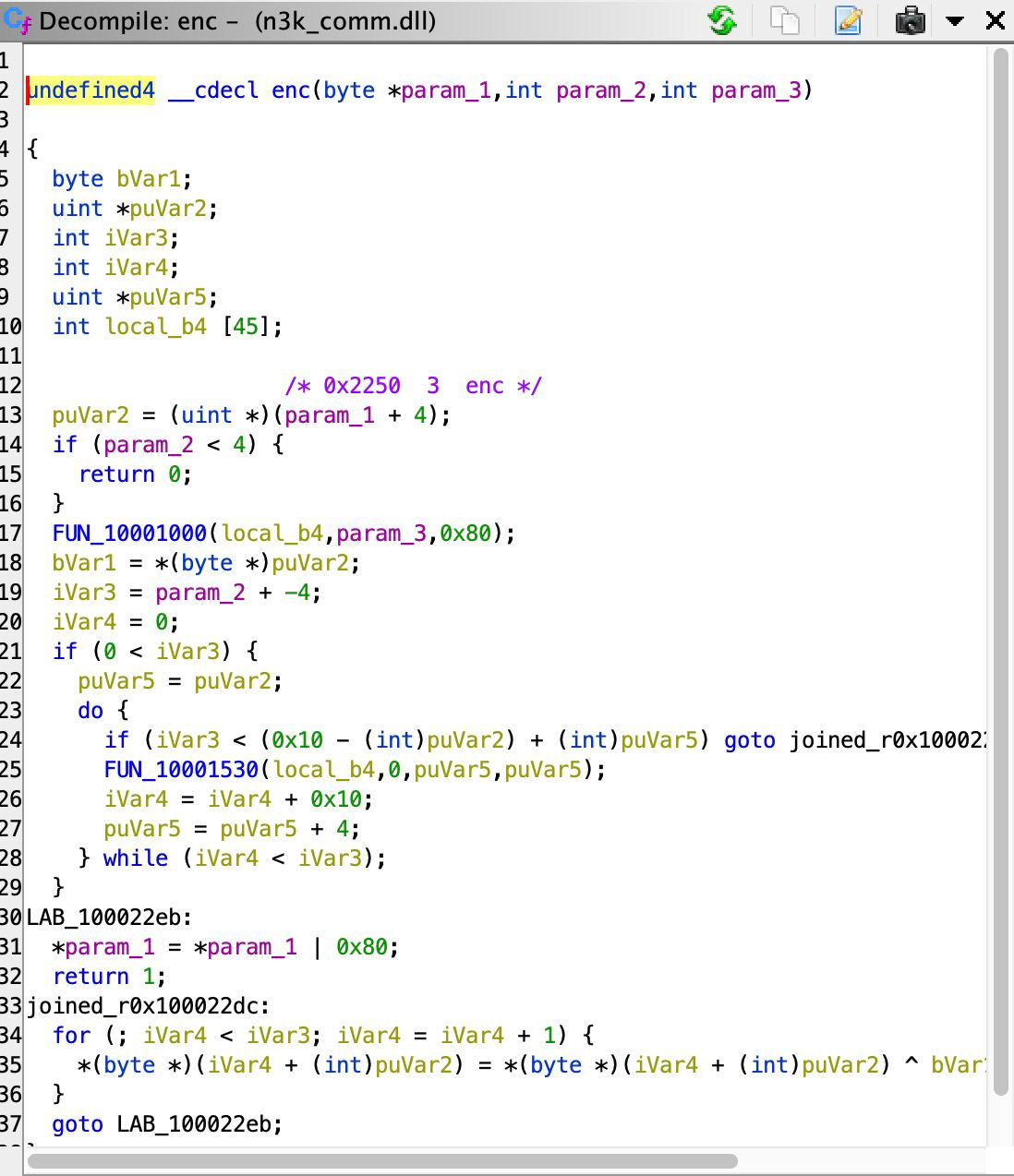
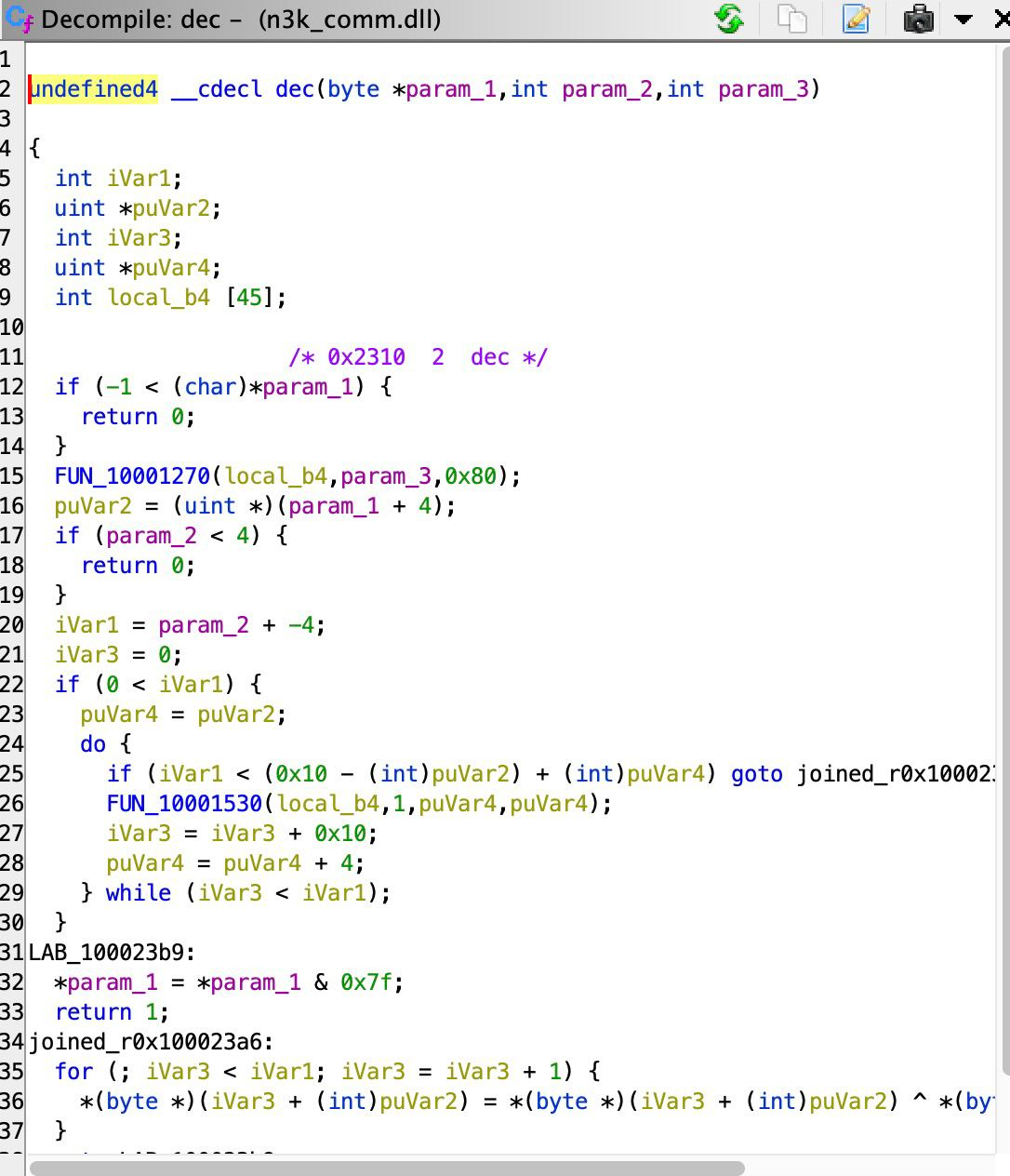
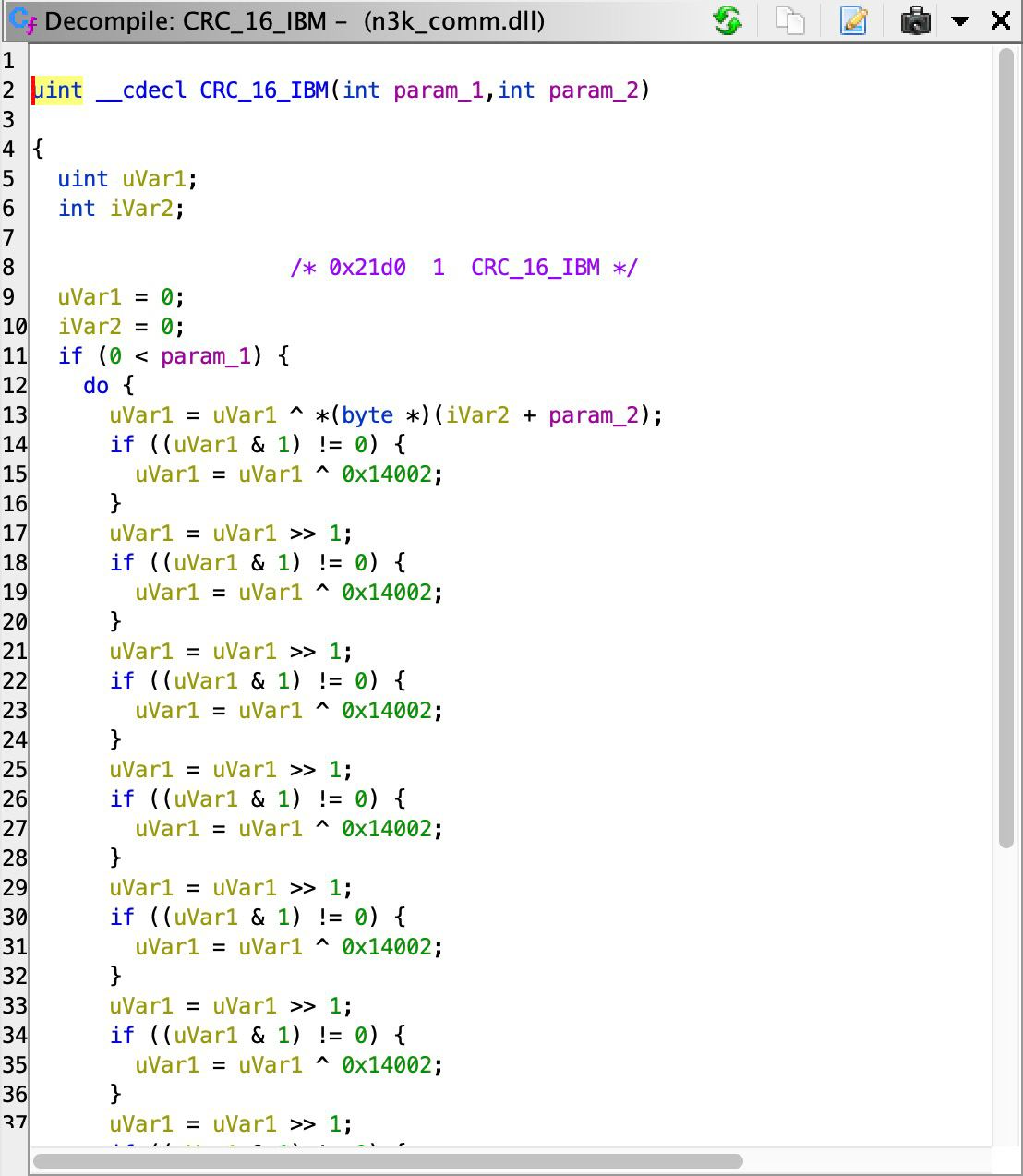
1. Відкриття бібліотеки **n3k\_comm.dll**.



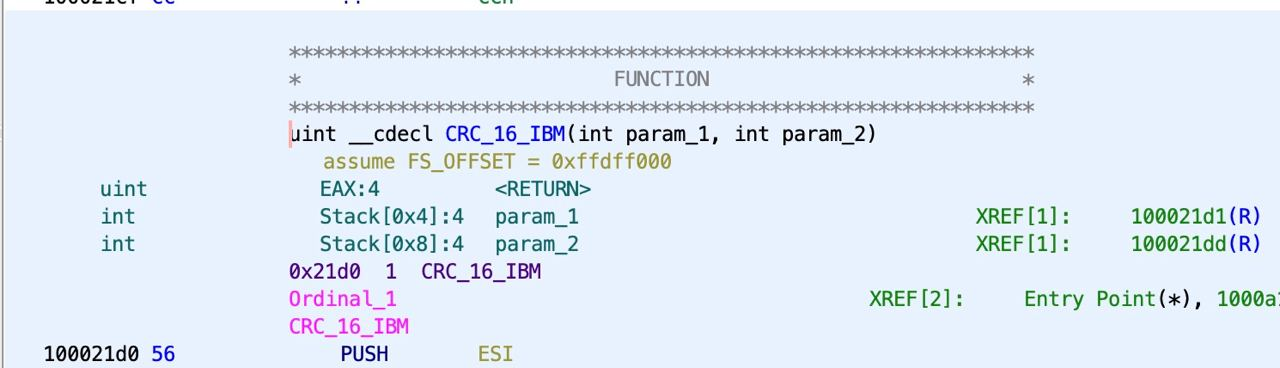
1. Список методів.



1. Код із бібліотеки.



1. Функція **CRC\_16\_IBM**.



**Код програми**

#include <windows.h>

#include <iostream>

#include <processthreadsapi.h>

#include <libloaderapi.h>

extern "C" {

typedef unsigned int (\*CRC\_16\_IBM)(int len, int param\_2);

typedef int (\*getIV)(char\* param\_1);

typedef int (\*getK)(char\* param\_1);

typedef int (\*enc)(byte\* param\_1, int param\_3, int param\_2);

typedef int (\*dec)(byte\* param\_1, int param\_2, int param\_3);

}

unsigned int MY\_CRC\_16\_IBM(int, int);

int My\_getIV(char\*);

int main()

{

auto hdl = LoadLibraryA("n3k\_comm.dll");

if (hdl)

{

auto crc16Func = reinterpret\_cast<CRC\_16\_IBM>(GetProcAddress(hdl, "CRC\_16\_IBM"));

if (crc16Func) {

auto dllResult = crc16Func(16, 0x1000b040);

auto myResult = MY\_CRC\_16\_IBM(16, 0x1000b040);

std::cout << "DLL CRC\_16\_IBM result = " << dllResult << std::endl;

std::cout << "My CRC\_16\_IBM result = " << myResult << std::endl;

}

else {

std::cout << "Function not found!" << std::endl;

}

auto getIVFunc = reinterpret\_cast<getIV>(GetProcAddress(hdl, "getIV"));

if (getIVFunc) {

char ivStr[17] = { 0 };

auto r = getIVFunc(ivStr);

char ivStr2[17] = { 0 };

auto r2 = My\_getIV(ivStr2);

std::cout << "DLL getIV result = " << ivStr << std::endl;

std::cout << "My getIV result = " << ivStr2 << std::endl;

}

auto getKFunc = reinterpret\_cast<getK>(GetProcAddress(hdl, "getK"));

if (getKFunc) {

char kStr[17] = { 0 };

auto r = getKFunc(kStr);

std::cout << "Dll getK result = " << kStr << std::endl;

}

auto encFunc = reinterpret\_cast<enc>(GetProcAddress(hdl, "enc"));

auto decFunc = reinterpret\_cast<dec>(GetProcAddress(hdl, "dec"));

if (encFunc && decFunc) {

byte str[16] = { 'V', 'L', 'A', 'D', '\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0','\0' };

int key = 0x1000b040;

auto r = encFunc(str, 16, key);

std::cout << "DLL enc result = " << str << std::endl;

auto r2 = decFunc(str, 16, key);

std::cout << "DLL dec result = " << str << std::endl;

}

FreeLibrary(hdl);

}

else {

std::cout << "Library not found!" << std::endl;

}

}

unsigned int MY\_CRC\_16\_IBM(int len, int address)

{

unsigned int result = 0;

for (int i = 0; i < len; i++)

{

result ^= \*(byte\*)(i + address);

for (int j = 0; j < 8; j++)

{

if ((result & 1) != 0)

result ^= 0x14002;

result >>= 1;

}

}

return result;

}

int My\_getIV(char\* param\_1)

{

char\* dest;

for (int i = 0; i < 16; i++)

{

dest = (char\*)(i + param\_1);

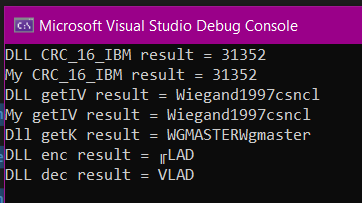
\*dest = (dest[(int)0x1000b040 - (int)param\_1] - i) - 1;

}

return 1;

}

**Результат роботи програми**



**Висновок**

Отримав навички основ реверс-інжинірингу.