kaspersky

Начинающий уровень

Реверс инжиниринг для начинающих

Сделайте первые шаги в области реверс инжиниринга и начните карьеру специалиста по поиску киберугроз или аналитика вредоносных программ

Благодаря этому тренингу вы сможете:

- Изучить основы языка ассемблера
- Освоить перевод высокоуровневых структур кода с помощью компиляторов
- Работать с контейнерами от простого пользовательского списка в C до шаблонизированных карт и векторов C++ в Rust
- Сделать первые шаги в использовании инструментов обратного инжиниринга
- Проводить анализ алгоритмов и структур с комбинацией компиляции фрагментов кода и их дизассемблирования

Язык курса - Английский

Требования

- Базовые знания в программировании
- Знакомство с языком ассемблера x86/64
- Начальный опыт работы с C++

Для кого

Отдельные специалисты, которые хотят освоить навыки реверс-инжиниринга и начать карьеру вирусного аналитика или исследователя киберугроз

Эксперты

Иван Квятковски
Исследователь безопасности

Денис Легезо
Исследователь безопасности

Программа курса

1 Theory

- The basics of the assembly language
- Working with assembly instructions
- Understanding function calls
- Recognizing argument

2 C-language “Hello World”

- reverse-engineering of programs written in C language
- Getting familiar with IDA and using the software to navigate inside assembly code

3 Simple Lambdas

- The difference between values and pointers
- Transmitting the arguments by value and by reference
- Old-style C-like memory management, where programmer is in charge of keeping proper references
- Reconstructing custom structures

4 Stuck in the heap

- The difference between main places to store the data
- Conception of stack and heapAutomatic, dynamic and static memory
- How the programmer’s decision where to store data affects the resulting binary
- Creating massive custom structures on stack and heap
- Analyzing them in resulting executable binary file to see the difference in these memory types
- Reversing the code with static variables, understand their position in executables

5 Lists and tricky pointers

- How custom C list looks on a binary level
- How to handle lists with pointers that point to the middle of next element
- Understanding custom data types, which you would met in binaries, further
- Using simple custom list to move towards real C++ STL containers
- Practice shifted pointers

6 C++ And OOP

- Recognizing C++ classes in assembly form
- Experimenting with IDA’s disassembler
- Working with code coming from the C++ STL

7 Pain In The Containers

- analysis C++ STL containers in compiled programs, creating proper structures
- Inserting and searching for operations in std::map, std::set
- The difference between map and set on binary code level
- Practicing surface std::map, set, pair analysis in binarie

8 Introduction to Golang Reverse-engineering

- The basics of the Go language
- reverse-engineering when faced with binaries generated by its compiler
- Using a debugger in order to easily obtain program arguments and return values

9 ‘Rusty’ Code

- Understanding non-stripped Rust code
- Dividing runtime and custom Rust code
- Demangling Rust function names

Связаться с нами:

support@kaspersky.happydesk.ru

www.kaspersky.com