

# WM3B3-24 Low Level Tools and Techniques for Cyber Security

21/22

**Department**

WMG

**Level**

Undergraduate Level 3

**Module leader**

Amila Kotte Liyanage

**Credit value**

24

**Module duration**

30 weeks

**Assessment**

100% coursework

**Study location**

University of Warwick main campus, Coventry

---

## Description

### Introductory description

Modern programming approaches typically abstract the things the developer needs to create away from the instructions that will execute on the machine. These high levels of abstraction use code generation programs such as compilers and assemblers to take the human author's input, and produce code that will execute as output. The modern programmer rarely needs to consider the underlying architecture of the machine that will execute the code.

There are situations where, rather than creating an executable from source, you need to go in the opposite direction; you need to infer what the source code might look like by analysing the executable. Maybe you have some potential malware; maybe you have an executable for which you no longer have the source. Either way, you want to know what the program will do, were it to run on your system.

In order to reverse back from the executable to the original, you need to understand the typical idioms that an operating system, architecture and code generation programs will adopt to convert high level constructs into low level executables.

If the executable is malware, then it is likely the authors will have strewn this road you wish to reverse with obfuscating hazards. Under these circumstances you need to understand the typical idioms of obfuscation.

## Module aims

- 1 - Identify common idioms and patterns used during code transformation and so explain the origin and organisation of arbitrary code and / or data fragments within an executable program.
- 2 - Apply tools and techniques as appropriate to infer the overall high level function of executable, potentially obfuscated, potentially malicious code.

## Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

The content of this module will be taught from a cyber security perspective.

- executable code from a variety of perspectives
- assembly language programming
- machine-level instruction set and organisation
- code generation
- reverse engineering techniques
- de-obfuscation
- common tools for reverse engineering
- anti-debugging mechanisms
- fuzzing

## Learning outcomes

By the end of the module, students should be able to:

- 1 - Identify common idioms and patterns used during code transformation and so explain the origin and organisation of arbitrary code and / or data fragments within an executable program.
- 2 - Apply tools and techniques as appropriate to infer the overall high level function of executable, potentially obfuscated, potentially malicious code.

## Indicative reading list

Aho, A. V., Lam, Monica S., Sethi, R. and Ullman, Jeffrey D., "Compilers: Principles, Techniques, and Tools", 2 Ed, Pearson (2013)  
Sikorski, Michael and Honig, Andrew "Practical Malware Analysis", No Starch Press (2012)  
Szor, Peter, "The Art of Computer Virus Research and Defense", Addison-Wesley (2005)

## Subject specific skills

- 1 - Identify common idioms and patterns used during code transformation and so explain the origin and organisation of arbitrary code and / or data fragments within an executable program.

2 - Apply tools and techniques as appropriate to infer the overall high level function of executable, potentially obfuscated, potentially malicious code.

## Transferable skills

Critical thinking, problem solving

---

## Study

### Study time

Type	Required
Supervised practical classes	6 sessions of 6 hours (15%)
Private study	68 hours (28%)
Assessment	136 hours (57%)
Total	240 hours

### Private study description

One third of independent study time will not directly contribute to assessment

Two thirds of independent study time will contribute to assessment

Lecture time falls within workshop time

## Costs

No further costs have been identified for this module.

---

## Assessment

You do not need to pass all assessment components to pass the module.

### Assessment group A1

	Weighting	Study time	Eligible for self-certification
Assessment component			
Coursework	100%	136 hours	Yes (extension)

Reassessment component is the same

## **Feedback on assessment**

Written feedback for each assignment  
Verbal feedback during tutorial sessions  
Solutions provided to tutorial questions  
Summative feedback on assignments

---

## **Availability**

## **Courses**

This module is Core for:

- Year 3 of UWMA-H651 Undergraduate Cyber Security