Understanding

IT Security Administration

Title
Understanding IT Security Administration
SECO-Institute IT Security Foundation Courseware

Author
Hans de Vries

Editorial support
Anna Mácsai

Publisher
SECO-Institute
www.seco-institute.org
info@seco-institute.org

ISBN
9789082978124

Print
First edition, first impression, November 2019

Layout and design
Hike Helmantel

UNDERSTANDING

IT Security Administration

SECO-Institute

IT Security

Foundation Courseware

Disclaimer
This publication is educational material and does not constitute legal advice. The SECO-Institute is not liable for any advice taken from this publication. For full information and guidance, please seek professional legal advice.
Although the authors and publisher have made every effort to ensure that the information in this book was correct at press time, the authors and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

Table of Contents

PREFACE 9

DOMAIN 1: SYSTEMS

Introduction 17

1 Computing hardware 19

1.1 Computer architecture 19
1.2 Central Processing Unit 19
1.2.1 CPU privileges 21
1.2.2 Other CPU protection mechanisms 22
1.3 Storage 23
1.3.1 Primary storage 23
1.3.2 Secondary storage 24
1.3.3 Tertiary storage 24
1.3.4 Offline storage 25
1.3.5 RAID 25
1.4 Peripherals 28

2 Operating systems 29

2.1 The OS as an abstraction layer 30
2.2 The OS as a housekeeper 32
2.2.1 Process management 32
2.2.2 Memory management 33
2.2.3 Device management 34
2.3 The OS as a watchdog 36
2.3.1 System calls 37

3 Modes of cooperation 38

3.1 Terminals 39
3.2 Client-server 39
3.3 Multi-tier architecture 41
3.4 Grid computing 43
3.5 Peer-to-peer computing 44
3.6 Virtualisation 45
3.7 Cloud computing 47

SUMMARY 52
TEST QUESTIONS 56

Domain 2: Software

Introduction ... 59

4 Types of software ... 61

5 Sources of vulnerabilities in software ... 63

5.1 Input ... 63
5.2 Data processing ... 68
5.3 External dependencies ... 71
5.4 Control mechanisms ... 73
5.4.1 Access control ... 73
5.4.2 Confidentiality ... 79
5.4.3 Integrity ... 81
5.4.4 Logging ... 85
5.5 Output ... 86
5.6 Installation & configuration ... 88
5.7 A closing word on software bugs ... 90

6 Security in databases ... 91

6.1 Integrity Control ... 95
6.2 Concurrency Control ... 95
6.3 Access Control ... 96
6.4 Recovery ... 97
6.5 Database-specific vulnerabilities ... 98

Summary ... 100
Test questions ... 103

Domain 3: Networks

Introduction ... 105

7 Network devices ... 107

7.1 Definitions ... 107
7.1.1 Node ... 107
7.1.2 Link ... 107
7.1.3 Network ... 107
7.1.4 Protocol ... 107
7.1.5 Architecture ... 107
7.2 The players in this story ... 108
7.3 Direct link ... 109
7.4 Hub ... 110
7.5 Switch ... 111
7.6 Gateway ... 115
7.7 Router ... 116
7.8 Firewall ... 118
7.9 Intrusion Detection System ... 120
7.10 A closing word on network devices ... 121

8 Network connections ... 122

8.1 Wired networks ... 122
8.2 Wireless networks ... 125

9 Network models ... 128

9.1 The OSI model ... 128
9.2 Encapsulation ... 131
9.3 The TCP/IP model ... 132

10 Network architecture ... 134

10.1 Topology ... 134
10.2 Local Area Networks ... 135
10.3 Hardening hosts ... 137
10.4 Compartmentalising networks ... 138
10.5 Designing secure networks ... 139

11 Network addressing ... 140

11.1 IP addressing ... 141
11.2 Private addressing ... 142
11.3 IP security ... 144

Summary ... 146
Test questions ... 150

Domain 4: Cryptography

Introduction 153

12 Principles of cryptography 155

12.1 Relevant definitions and terms 155
12.2 One-time pads 157
12.3 Hashing 159
12.4 Encryption 161

13 Public Key Infrastructure 166

14 Important cryptographic networking protocols 170

14.1 Kerberos 170
14.2 Transport Layer Security 172
14.3 IPsec 176
14.4 Tor 178
14.5 VPN 180

15 Important cryptographic applications 181

15.1 Secure Shell 181
15.2 File(system) encryption 181
15.3 Secure email 183
15.4 Steganography 183

Summary 186
Test questions 189

Domain 5: IT Security Administration

Introduction ... 191

16 Know your enemy ... 193

16.1 Hacking as a process ... 198
16.1.1 Step 1: Preparation ... 198
16.1.2 Step 2: Passive intelligence gathering ... 199
16.1.3 Step 3: Active intelligence gathering ... 199
16.1.4 Step 4: Penetration ... 201
16.1.5 Step 5: Control ... 201
16.1.6 Step 6: Exit ... 202
16.1.7 Step 7: Conclusion ... 203
16.2 Tools of the trade ... 204
16.2.1 Gathering information ... 204
16.2.2 Deception ... 205
16.2.3 Malware ... 206
16.2.4 Breaking passwords ... 209
16.2.5 Controlling hosts ... 210
16.2.6 Test suites ... 211

17 Know yourself ... 212

17.1 IT security administration as a process ... 213
17.1.1 Step 1: Preparation ... 213
17.1.2 Step 2: Intelligence gathering ... 217
17.1.3 Step 3: Risk assessment ... 218
17.1.4 Step 4: Risk mitigation ... 221
17.1.5 Step 5: Verification ... 225
17.1.6 Step 6: Vulnerability management ... 226
17.1.7 Step 7: Incident response ... 229
17.2 Tools of the trade ... 231
17.2.1 Evolution before revolution ... 231
17.2.2 Communication ... 232
17.2.3 Challenge yourself ... 233
17.2.4 Stay up-to-date ... 234

Summary ... 236
Test questions ... 240

Annexes

Annex A: Description of a CVE bulletin 244
Annex B: Introduction to Data Flow Diagrams 247
Annex C: IT security in relation to information security 250
Annex D: Answers and explanations 252

Preface

In modern society it is almost impossible to spend a day without using any IT facilities. IT is everywhere and our dependency on it is increasing by the day. This is great, as it brings many benefits, but IT is complex and that means it can fail in many ways. Some of those failures will be caused by defects, others by manipulation. Whichever the cause, it is up to information security professionals to manage the risk of those failures.

An adagio that is often refered to in information security is "people, process, technology". This reminds us that it takes people with the right knowledge, competences and behaviour, adequate organisational processes, and proper use of technology to establish an environment wherein information can be used and managed in a secure manner.

This book focusses on the technological aspects of information security, i.e. IT security. The human and organisational aspects of information security are the focus of another book ("Understanding Information Security Management").

The goals of this book

Every highly specialised area of expertise has its own vocabulary to describe the concepts and tools that are most important to it. IT is no exception to this rule. Anyone who has ever worked in or with IT will have quickly realised just how much jargon is used in this field. The number of terms, acronyms, abbreviations, and product names can truly be overwhelming.

Most IT functionaries specialise in one or a few branches of IT, such as network or system administration, but an IT security administrator is involved in all aspects of IT. This means IT security administrators must to be able to communicate and work with all branches in IT. To do this effectively they need to master the language used by all those branches. One of the main goals of this book, therefore, is to familiarise the reader with the vocabulary used in IT security administration.

IT security administrators are responsible for designing and maintaining an environment that offers the level of security needed by their organisation. To do that they must understand what tools are available, and how they can be used to increase the security of the environment as a whole. However, in most organisations IT security administrators are not responsible for actually implementing the solutions used to create a secure environment. That most often is entrusted to network, system, database, or application administrators, which makes it necessary for an IT security administrator to understand

their way of working. Thus, another main goal of this book is to familiarise the reader on a conceptual level[1] with the functionality provided by the most common tools and protocols seen in a secure environment.

Finally, IT security administrators have their own field of expertise, which is not always familiar to other IT administrators. To make sure the environment for which they are responsible is and stays secure IT security administrators need to know about vulnerabilities, threats, and attacks. They need to understand how malicious agents operate, and what can be done to prevent malicious agents from compromising the security of networks, systems or data. Introducing the reader to how attackers and defenders operate is the final goal of this book.

This book assumes the reader has no prior knowledge of IT security, but does aspire to gain the knowledge and understanding necessary to become an IT security administrator. Although this book requires no prior knowledge, affinity with technology will definitely make it easier for the reader to grasp all the concepts introduced in this book. IT security administration is an extremely broad topic, which means this book has to cover a lot of ground. Therefore some perseverance might be needed too.

The structure of this book

In IT, people use systems to accomplish their tasks. These systems are comprised of hardware and software. Hardware, such as displays, keyboards, storage devices, network cards, and peripherals, acts like an interface between the physical world and the digital world. It enables systems to perform their tasks and deliver the results to a user or another system. Software is what makes hardware intelligent and flexible. It enables a system to execute instructions and process data that were not built in the system during manufacturing. General-purpose software can be divided into three categories: Operating Systems that make a system useful for many purposes, applications that can be used to execute specific tasks, and services that can be used to offer remote functionality to a broader public than just the people that can physically access the system. To make communication between systems possible a network is needed. And to make sure all traffic is efficiently handled the network makes use of special devices, like switches and routers. Local networks can offer services to their constituency, but can also be linked to each other to create larger networks, which together make up the Internet.

1 To make the contents of this book applicable to as wide a range of environments as possible it will limit itself to concepts and will not go into the specifics of particular products.

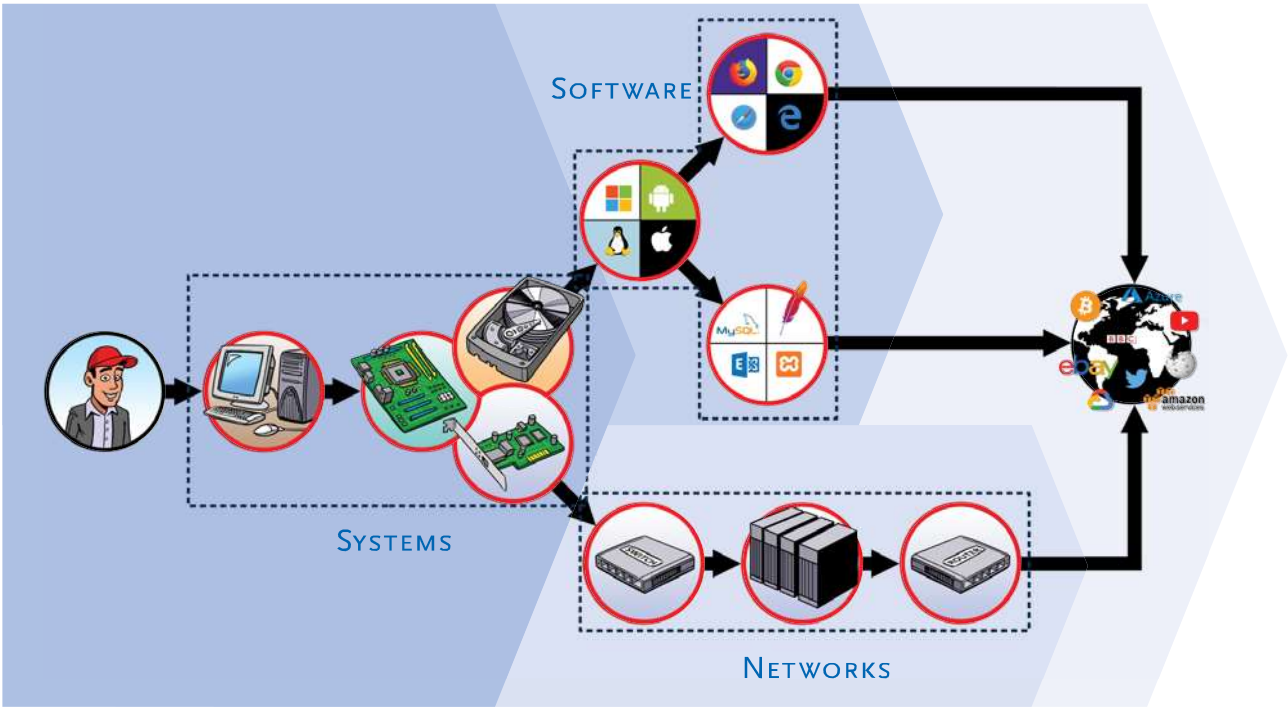

Figure 1: The scope of this course on IT security administration

This book follows the logical structure described above.

Domain 1 "Systems" describes the most important hardware components that are found in systems and explains how they function (Chapter 1). It also explains what an Operating System is and how it manages the hardware of the system it is running on (Chapter 2). This domain ends with an overview of how systems can cooperate with each other (Chapter 3).

Domain 2 "Software" presents the main types of software (Chapter 4). It analyses how software processes instructions and data, and describes the types of errors that can occur during processing (Chapter 5). Understanding this domain does not require any programming knowledge; the topics are discussed on a conceptual level, focusing on the causes and potential impact of software errors. Sufficient detail is provided to understand the security bulletins published by manufacturers of software, security researchers, and other parties. In essence databases are software, just like any other application or service, but because they are so important to processing and storing data, and because their security involves specific mechanisms they are covered in a separate chapter at the end of Domain 2 (Chapter 6).

Domain 3 "Networks" takes a look at the devices that are used to build a network and manage the traffic that is sent over it (Chapter 7). It also describes the most common types of connections between network components (Chapter 8). To understand what is involved in making networks the versatile and useful tools they are, the two most important network models are described (Chapter 9). Building on this theoretical knowledge the fundamentals of designing secure networks are covered (Chapter 10). The domain is closed with a description of how individual nodes within a network can be accessed (Chapter 11).

Domain 4 "Cryptography" explains the fundamentals of cryptography, with a focus on public key encryption (Chapter 12). This is followed by an overview of how Public Key Infrastructure can be used to manage encryption and encryption keys in an enterprise environment (Chapter 13). Domain 4 is concluded with an overview of the most important cryptographic networking protocols (Chapter 14) and applications (Chapter 15).

Domain 5 "IT security administration" brings together all the information covered in the previous domains. It shows how attackers find vulnerabilities in IT infrastructures and how they exploit them to gain unauthorised access to data and systems (Chapter 16). Subsequently, it outlines how an IT security administrator can secure IT infrastructures and minimise an attacker's chances of success. It also outlines what to do when an attack succeeds in spite of all security efforts (Chapter 17).

The annexes provide additional practical information on some of the topics covered in this book.

Notes to the reader

Some of the exercises included at the end of each domain use practical examples to illustrate how the knowledge covered in this book is applied in practice. These examples are situated in a fictitious company, called Bicsma. This organisation was created by the SECO Institute to provide context for the assignments and case studies used in SECO's educational material.

Bicsma is a Dutch beverage manufacturer. The company is managed by three brothers: Vincent, Hans, and Marcel Bicsma.

Bicsma was launched in the family's kitchen when Ma Bicsma, the family matriarch, started producing home-made fruit juices for local supermarkets. The juices were so well-received that Bicsma soon became a medium-sized company with two production facilities and 354 employees.

Bicsma is engaged in both business-to-business and business-to-consumer activities. In a somewhat unusual manner, the company sells drinks to both supermarkets and individuals.

The summaries and test questions included in this book help readers to prepare for the SECO Institute's IT Security Foundation certification exam. The test questions reflect the content and difficulty of the certification exam, but do not fully mimic the format used in the actual exam, which consist of multiple choice questions only. The open questions included in this book are intended to encourage creative and critical thinking. These questions do not always have an answer that is 100% correct or wrong. They are aimed at helping you find your own best approach to a problem.

Domains


WILL THAT WORK ON MY MS-DOS SYSTEM?

Domain 1: Systems

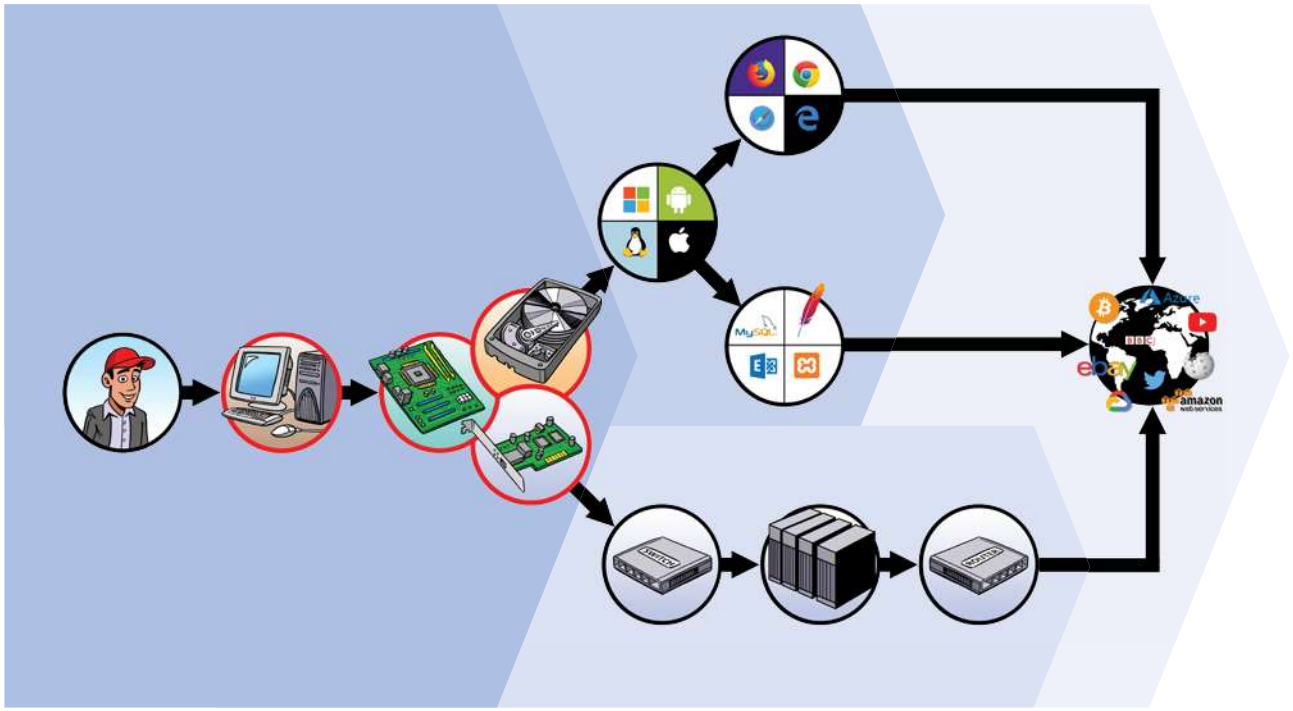

Figure 2: Systems in the scope of IT security administration

Introduction

Everything in the digital domain runs on some type of system. For that reason alone every IT security administrator needs to cooperate with system administrators. To be an effective security administrator you must understand what they do and know at least some of the vocabulary they use. You need to be able to communicate with system administrators about threats and work with them to design effective security measures that mitigate (some of) the risks those threats pose to your organisation.

The core of each system is some form of computing hardware combined with an Operating System (OS) that provides a uniform interface to the users of that hardware. This core often is complemented with peripherals and application software that enables the system to perform specific tasks.

To identify and assess possible threats to a system you need to understand on a conceptual level how each of these four components (hardware, OS, peripherals, and applications) functions and how they interact with each other. In this domain we will take a look at hardware and Operating Systems. Further on in this book we will dedicate a whole domain to software in general.

1 Computing hardware

Computing devices come in a great variety of forms. First you had large systems, like mainframes, with terminals to access them. Then the PC revolution followed, which resulted in the availability of desktop computers and laptops[2] everywhere. With the rise of the Internet, and later the Cloud, the role of servers that offer all kinds of network services increased again. And most recently mobile devices like tablets, smartphones, and wearables have added a new layer of complexity to the ecosystem of computing devices.

In essence all these devices do not differ that much. They all are made up of an ensemble of computing hardware, an OS, data storage facilities, and interfaces to enable communication (input and output). Of course, the specifics can differ immensely for each type of device, but such differences are not relevant for this chapter.

1.1 Computer architecture

A computer is a highly complex apparatus with a great many separate components that seamlessly work together. To make this possible a clear set of rules has to be agreed upon and implemented. Among other things these rules describe the device's functionality and interoperability, including both physical properties (e.g. voltages, sizes, number and type of connections) and logical properties (e.g. what operations are available, what instructions can be given, and how input and output signals should be interpreted). Collectively these rules are referred to as the **architecture** of the device.

1.2 Central Processing Unit

The heart of each computing device is the **Central Processing Unit** (**CPU**). The CPU is what makes a device "intelligent". It provides for the device's most important computing capabilities. A CPU consists of an immense number of transistors organised in integrated circuits (IC) on a chip. Each CPU is built up from three core components: an **Arithmetic Logic Unit** (**ALU**), **registers**, and a **Controller**.

The ALU can execute a predefined set of operations that are hardwired into its design. Such operations are either mathematical calculations (e.g. addition and subtraction), logical operations (AND, OR, XOR, or bit inversion), or bit-shifting operations (left shift, right shift, rotation, and a few variations on those). The more complex the operations an ALU can perform, the more impact this has on its overall speed. Because

2 Laptops are of course mobile devices, but in terms of functionality and use cases they resemble desktop PCs more than they do tablets or smartphones.

of this ALUs are designed to execute only a limited set of well-defined operations. Currently there are two schools of instruction set design: **Reduced Instruction Set Computers** (**RISC**), where the CPU contains a small set of general purpose instructions, and **Complex Instruction Set Computers** (**CISC**), where the instruction set includes more complex and specialised operations. Well-known examples of CISC and RISC architectures are the x86 and ARM family of chips, respectively.

As described above the ALU can execute operations, but that's all it can do. So, there needs to be another component that feeds the ALU with a proper sequence of instructions and the data on which to execute them. That is what the Controller does: it is the housekeeper of the CPU. The Controller makes sure that all instructions are executed in the right order, that the data on which to execute the instructions is given to the ALU at the right moment, and that the results are retrieved from the ALU before they are overwritten again. The Controller also translates the software instructions it receives to a sequence of internal hardware instructions to be executed by the ALU. Finally, it regulates when software is given processing time on the CPU.

The third core component of all CPUs are its registers. These are small storage spaces with a fixed location within the CPU that can be read from or written to extremely quickly. Registers can contain instructions or data. To further increase performance some registers are assigned specialised tasks, for example:

— The instruction register, which holds the instruction that is currently being executed.
— Address registers, that can be used to quickly look up a specific address.
— Status registers, that contain information on the current state of the CPU.
— Constant registers, that contain fixed values that are often used, such as pi.

More advanced CPUs often contain additional components, like fast memory caches that increase the CPU's performance by lowering the time it needs to access data. Another important component is the **Memory Management Unit** (**MMU**) that provides an abstraction layer between the logical memory addresses used by software and the physical addresses used by hardware.

Some examples of well-known CPU architectures are:

— The x86-family originated by Intel, mostly used in systems that focus on performance, like servers, PCs, and laptops.
— The ARM-family originated by Acorn, mostly used in mobile devices that focus on relatively low power consumption.
— The z/Architecture-family originated by IBM, mostly used in mainframes.

1.2.1 CPU privileges

To protect systems against misuse, the design of a CPU incorporates security features that enable it to guard what code is executed (or not). One of these features is the **CPU Privilege Level** (**CPL**) system built in the x86-family of CPUs. To explain how this system functions we need to take a detailed look at the inner workings of a CPU.

Every piece of code that is executed by the CPU is assigned a privilege level, ranging from 0 to 3. This privilege level is used to restrict access to three resources: the ability to execute specific instructions in the ALU, access to memory, and access to I/O ports. Code at privilege level 0 has the most privileges, and code at privilege level 3 has the least. Each privilege level is given access to the resources associated with that level *and* the resources available to the lesser levels. In other words, code running at privilege level 3 can only access the resources associated with level 3, whereas code running at privilege level 0 can access all resources. This concept is often depicted as a set of concentric rings with borders between them. When code needs to access data or an instruction in a more privileged ring, it has to use a gate that protects the integrity of the CPU by giving access only to a limited set of resources and only if certain conditions are met.

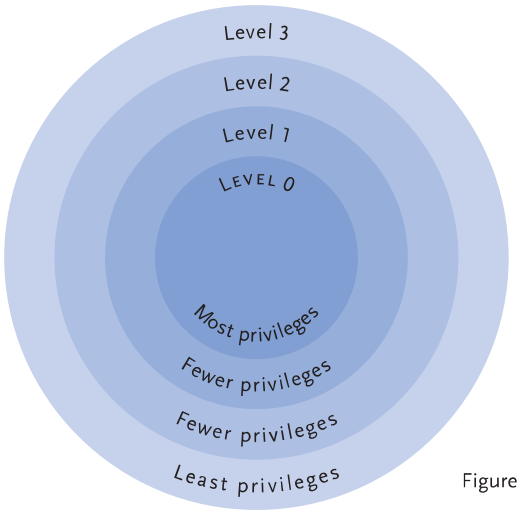

Figure 3: CPU privilege levels

Each memory access request includes a 2-bit Requested Privilege Level (RPL) field. Each memory segment contains a 2-bit field, called the Description Privilege Level (DPL), which defines the minimum privilege level needed to access that memory segment. And the CPU has a code segment register that includes a 2-bit Current Privilege Level (CPL)