



**UNIVERSITY
OF LONDON**

Introduction to computing and the internet: Volume 2

D. Miller

CO1110

2019

Undergraduate study in
Computing and related programmes

This guide was prepared for the University of London by:

D. Miller, formerly of the Department of Computing, Goldsmiths, University of London.

This guide draws on material in the previously published edition of the guide by R. Kibble.

This is one of a series of subject guides published by the University. We regret that due to pressure of work the author is unable to enter into any correspondence relating to, or arising from, the guide. If you have any comments on this subject guide, favourable or unfavourable, please use the form at the back of this guide.

This subject guide is reproduced in colour on the VLE and you may find it easier to understand if you access the online PDF.

University of London
Publications Office
32 Russell Square
London WC1B 5DN
United Kingdom
london.ac.uk

Published by: University of London

© University of London 2019

The University of London asserts copyright over all material in this subject guide except where otherwise indicated. All rights reserved. No part of this work may be reproduced in any form, or by any means, without permission in writing from the publisher. We make every effort to respect copyright. If you think we have inadvertently used your copyright material, please let us know.

Contents

Chapter 1: Introduction to the subject guide	1
1.1 Route map to the guide	1
1.2 Glossary of key terms.....	2
1.3 Introduction to the subject area	2
1.4 Syllabus.....	3
1.4.1 Volume 1	3
1.4.2 Volume 2	3
1.5 Aims of this course.....	4
1.6 Learning objectives for the course	4
1.7 Learning outcomes for students.....	4
1.7.1 Prior knowledge required	4
1.7.2 Bitwise AND.....	5
1.7.3 Hexadecimal.....	5
1.8 Overview of learning resources	6
1.8.1 The subject guide.....	6
1.8.2 Essential reading.....	7
1.8.3 Further reading	7
1.8.4 Websites.....	8
1.8.5 Online Library and the VLE.....	9
1.8.6 End of chapter Sample examination questions and Sample answers.....	9
1.9 Examination advice.....	9
1.10 Overview of the chapter	10
1.11 Test your knowledge and understanding	10
1.11.1 A reminder of your learning outcomes.....	10
Chapter 2: Network models, protocols and standards	11
2.1 Introduction.....	11
2.1.1 Aims of the chapter.....	11
2.1.2 Learning outcomes	11
2.1.3 Essential reading.....	12
2.1.4 Further reading	12
2.1.5 References cited.....	12
2.2 Glossary of key terms	12
2.3 Computer networks and internetworking.....	13
2.3.1 The ARPANET.....	14
2.3.2 Development of TCP/IP protocols.....	14
2.3.3 Design principles of TCP/IP protocols	15
2.3.4 Introduction and success of TCP/IP protocols	15

2.4 Network models	16
2.4.1 Abstraction and data encapsulation	16
2.4.2 Layering	16
2.4.3 The OSI model.....	16
2.4.4 The TCP/IP model.....	17
2.4.5 Headers and data packets in the TCP/IP model.....	17
2.4.6 What is a data packet?.....	18
2.4.7 Comparing the models	18
2.4.8 The hybrid model	19
2.5 What made the TCP/IP protocol suite so successful?	19
2.5.1 TCP/IP data transmission principles.....	20
2.5.2 The TCP and UDP protocols	20
2.6 Key features of the TCP/IP protocol suite	20
2.6.1 Packet switching	20
2.6.2 Dynamic routing	21
2.6.3 Physical addresses	21
2.6.4 Logical addresses.....	21
2.6.5 Name resolution.....	21
2.7 The client/server model	22
2.7.1 Port numbers and 'well-known' ports	22
2.8 Standards and regulation.....	23
2.8.1 The Internet Engineering Task Force and RFCs.....	23
2.8.2 The Internet Corporation for Assigned Names and Numbers (ICANN)	23
2.8.3 Links to other standards and protocols.....	23
2.9 Overview of the chapter	24
2.10 Reminder of learning outcomes	24
2.11 Test your knowledge and understanding.....	25
2.11.1 Sample examination questions.....	25
Chapter 3: The Internet layer and the IP protocol.....	27
3.1 Introduction.....	27
3.1.1 Aims of the chapter.....	27
3.1.2 Learning outcomes	27
3.1.3 Essential reading	27
3.1.4 Further reading.....	27
3.1.5 References cited.....	28
3.2 Glossary of key terms	28
3.3 The Internet layer and the IP protocol.....	30
3.3.1 Unreliable, best effort, connectionless delivery	30
3.3.2 The responsibilities of the IP protocol	31
3.3.3 An historical note	31

3.4 The IP datagram header.....	32
3.4.1 Fields of the IPv4 datagram header.....	33
3.5 Routing.....	34
3.5.1 Routing tables.....	35
3.5.2 Next-hop forwarding.....	35
3.5.3 Default addresses.....	36
3.5.4 Address masks.....	36
3.5.5 Example of calculating a network ID from an address mask.....	37
3.5.6 Processing and discarding packets.....	38
3.5.7 Dynamic routing.....	39
3.6 IPv4 and classful addressing.....	39
3.6.1 The original IPv4 classful address scheme.....	39
3.6.2 Exclusions.....	40
3.6.3 Class A network and host addresses.....	40
3.6.4 Class B network and host addresses.....	40
3.6.5 Class C network and host addresses.....	40
3.6.6 The all zeros and all ones host addresses.....	41
3.7 Exhaustion of IPv4 addresses: Subnetting, CIDR and IPv6.....	41
3.8 Subnetting.....	42
3.8.1 Subnet masks.....	43
3.8.2 Example of finding a network address from a subnet mask.....	44
3.8.3 Number of subnets and hosts.....	44
3.8.4 Subnet masks for class C addresses.....	45
3.8.5 A subnetting example.....	45
3.8.6 Finding subnet addresses.....	47
3.9 IPv6.....	47
3.9.1 Network Address Translation (NAT).....	49
3.10 Classless Inter-Domain Routing (CIDR).....	49
3.10.1 Supernetting.....	50
3.10.2 CIDR and routing.....	50
3.10.3 CIDR and the IPv4 address space.....	50
3.11 Overview of the chapter.....	51
3.12 Reminder of learning outcomes.....	52
3.13 Test your knowledge and understanding.....	52
3.13.1 Sample examination questions.....	52
Chapter 4: TCP/IP transmission.....	53
4.1 Introduction.....	53
4.1.1 Aims of the chapter.....	53
4.1.2 Learning outcomes.....	53

4.1.3 Essential reading.....	53
4.1.4 Further reading	53
4.1.5 References cited	54
4.2 Glossary of key terms	54
4.3 Transport layer protocols and port numbers	55
4.3.1 Well-known port numbers.....	56
4.3.2 Port numbers as links between the Transport and Application layers	57
4.3.3 TCP and UDP ports	57
4.4 UDP	58
4.4.1 UDP header.....	59
4.5 TCP.....	60
4.5.1 TCP header	60
4.5.2 Fields of the TCP header	60
4.5.3 Three-way handshake	61
4.5.4 Cumulative acknowledgement	62
4.5.5 The adaptive retransmission algorithm	62
4.5.6 Duplicate ACKs	63
4.5.7 Duplicate ACKs and fast retransmission	63
4.5.8 Flow control and the sliding window	63
4.5.9 Retransmission and cumulative ACKs.....	64
4.5.10 SACK	65
4.5.11 Congestion control.....	65
4.5.12 Active areas of research.....	66
4.6 Overview of the chapter	67
4.7 Reminder of learning outcomes	67
4.8 Test your knowledge and understanding	67
4.8.1 Sample examination questions	67
Chapter 5: The world wide web and email	69
5.1 Introduction.....	69
5.1.1 Aims of the chapter.....	69
5.1.2 Learning outcomes	69
5.1.3 Essential reading	69
5.1.4 Further reading	70
5.1.5 References cited.....	70
5.2 Glossary of key terms	71
5.3 An historical note: Linux.....	76
5.4 An historical note: HTML.....	77
5.4.1 Static HTML.....	78
5.4.2 Browser wars and proprietary tags.....	79

5.5 How the world wide web works: the client-server model	80
5.5.1 The Hypertext Transfer Protocol	81
5.5.2 Web browsers	81
5.5.3 Web servers	81
5.5.4 Search engines	82
5.5.5 The Domain Name System (DNS)	82
5.5.6 Domain names	83
5.5.7 Uniform Resource Locators	84
5.5.8 How a web browser works	85
5.5.9 Stateless connections and cookies	86
5.6 An historical note: Fault tolerance, XHTML and the Semantic Web	86
5.6.1 WHATWG, HTML5 and the role of W3C	88
5.7 HTML5	88
5.7.1 Writing HTML5 code	89
5.7.2 HTML5 template	94
5.7.3 HTML5 fault tolerance and future proofing	94
5.7.4 HTML5 and progressive enhancement	96
5.8 CSS and HTML5	96
5.8.1 Three ways to format a web page with CSS	97
5.8.2 CSS precedence	98
5.8.3 Syntax for CSS	98
5.8.4 Fonts	99
5.8.5 Colours	100
5.8.6 Images	101
5.8.7 Tables in HTML5 with CSS	102
5.8.8 Advantages of CSS	104
5.9 HTML5 and JavaScript	106
5.10 Dynamic web pages	107
5.10.1 Client-side scripting	107
5.10.2 Server-side scripting	107
5.11 Electronic mail and its protocols	109
5.11.1 SMTP	109
5.11.2 SMTP commands	109
5.11.3 SMTP headers	110
5.11.4 SMTP and email spoofing	110
5.11.5 MIME	111
5.11.6 POP3 and IMAP	111
5.12 Overview of the chapter	112
5.13 Reminder of learning outcomes	112

5.14 Test your knowledge and understanding	113
5.14.1 Sample examination questions	113
Chapter 6: Legal framework	115
6.1 Introduction	115
6.1.1 Aims of the chapter	115
6.1.2 Learning outcomes.....	115
6.1.3 Essential reading.....	116
6.1.4 Further reading	116
6.1.5 References cited	116
6.2 Glossary of key terms	117
6.3 Magazines and web pages of interest to computing professionals.....	119
6.4 Intellectual property	122
6.4.1 An historical note: The Berne Convention	122
6.4.2 An historical note: The Paris Convention	122
6.4.3 TRIPS	122
6.5 IP protection and TRIPS	123
6.5.1 Copyright.....	123
6.5.2 Copyright in software and databases.....	124
6.5.3 Trademarks	124
6.5.4 Geographical indications	124
6.5.5 Industrial designs	125
6.5.6 Patents.....	125
6.5.7 Layout-designs (topographies) of integrated circuits.....	125
6.5.8 Trade secrets.....	125
6.6 IP protection in international treaties post-TRIPS.....	126
6.6.1 The WIPO Copyright Treaty of 1996	126
6.6.2 The WIPO Performances and Phonograms Treaty of 1996	127
6.6.3 Moral rights	127
6.6.4 Moral rights in the Internet Treaties	127
6.7 Database right in the EU.....	128
6.8 The three-step test and fair use	130
6.8.1 Fair dealing.....	130
6.8.2 EU Copyright Directive of 2001.....	131
6.8.3 Back-up copies of software and EU law.....	131
6.9 Patent and copyright law with respect to software and applications.....	132
6.9.1 Copyright in computer software	132
6.9.2 Can software be patented?	133
6.9.3 Protecting software as a trade secret	134
6.9.4 Patenting software	135

6.10 ISPs and the law.....	136
6.10.1 ISPs and copyright.....	137
6.10.2 ISPs and defamation.....	137
6.10.3 ISPs and data protection.....	139
6.10.4 Surveillance and monitoring.....	139
6.11 An historical note: The Cybercrime Convention.....	140
6.11.1 The Cybercrime Convention and substantive criminal law	141
6.12 Computer misuse	142
6.12.1 Definition of cybercrime	142
6.12.2 Additional Protocols to the Cybercrime Convention	143
6.12.3 The Computer Misuse Act 1990 (CMA) as amended by the Police and Justice Act 2006 (PJA) and the Serious Crimes Act 2015 (the 2015 Act).....	144
6.12.4 The five categories of offence under the CMA 1990 as amended.....	145
6.12.5 What the CMA does not cover	145
6.13 Data protection and the General Data Protection Regulation (GDPR)	147
6.13.1 Application of the GDPR outside the EU	147
6.13.2 The GDPR in the UK.....	148
6.13.3 Key definitions under the GDPR.....	148
6.13.4 The seven principles of data protection.....	149
6.13.5 Lawful data processing.....	150
6.13.6 Rights of individuals with respect to personal data under the GDPR.....	150
6.13.7 Right to information.....	151
6.13.8 The right of access.....	152
6.13.9 Other rights of data subjects under the GDPR.....	153
6.13.10 Special categories of personal data	153
6.13.11 Restrictions and derogations (exemptions)	154
6.13.12 International data transfers	155
6.13.13 Higher profile for data protection under GDPR	156
6.13.14 The ePrivacy Directive of 2002	157
6.13.15 The Cookie Directive.....	158
6.13.16 Workplace monitoring of communications	159
6.14 Unresolved legal issues at the time of writing.....	160
6.14.1 The EU Directive on Copyright in the Digital Single Market and the end of the internet as we know it.....	160
6.14.2 The future of software	162
6.14.3 Net neutrality	163
6.14.4 Artificial intelligence.....	163

6.15 Overview of the chapter.....	164
6.16 Reminder of learning outcomes.....	164
6.17 Test your knowledge and understanding.....	165
6.17.1 Sample examination questions.....	165
Chapter 7: Network security issues.....	167
7.1 Introduction.....	167
7.1.1 Aims of the chapter.....	167
7.1.2 Learning outcomes.....	167
7.1.3 Essential reading.....	167
7.1.4 Further reading.....	167
7.1.5 References cited.....	168
7.2 Glossary of key terms.....	169
7.3 What is hacking?.....	172
7.3.1 White hat hackers.....	172
7.3.2 Motivations for hacking.....	172
7.3.3 What hackers do.....	173
7.4 Why are networks and computers so vulnerable?.....	173
7.4.1 Protocol design.....	174
7.4.2 The security paradigm.....	175
7.4.3 Criminal involvement and opportunity.....	175
7.4.4 Operating systems and network security.....	175
7.4.5 Patching, responsible disclosure, bug bounties and exploit Wednesday.....	176
7.4.6 Responsible disclosure and the chilling of research.....	178
7.4.7 New developments and commercial imperatives.....	178
7.4.8 Government as the problem and the solution.....	179
7.5 Is Windows more of a target than other operating systems?	180
7.6 How hackers hack.....	181
7.6.1 Use the credentials of a legitimate system user.....	181
7.6.2 Persuade the user to hack themselves.....	182
7.6.3 Arbitrary code execution exploit.....	182
7.6.4 Opportunism.....	182
7.6.5 Session hijacking (AKA cookie hijacking) and packet sniffing.....	183
7.6.6 Creating a backdoor.....	183
7.6.7 Browser plug-ins and extensions.....	183
7.6.8 Hacking mobile phones.....	184
7.6.9 The Internet of Things.....	185
7.7 Definition of malware.....	185
7.7.1 Not malware: Potentially unwanted programs (PUPs), cookies and spyware.....	186

7.7.2 Types of malware	186
7.7.3 Types of viruses	188
7.7.4 Other malicious exploits	190
7.7.5 Malware developments and current threats	191
7.8 Gaining access	192
7.8.1 Removable media	192
7.8.2 Email	192
7.8.3 The world wide web: drive by downloads	194
7.8.4 The Windows Store	194
7.8.5 The world wide web: malvertising	195
7.8.6 Self-propagation	195
7.8.7 Hacking to install malware	195
7.8.8 How viruses install themselves	195
7.9 The payload	195
7.9.1 The infection	196
7.9.2 Possible virus payloads	196
7.9.3 Payload of a worm	196
7.9.4 The Trojan payload	197
7.10 Computer security and defence against malware	198
7.10.1 Symptoms of malware infection	198
7.10.2 Anti-virus and malware detecting software	200
7.10.3 How to protect against unauthorised access, credential stealing and infection	201
7.11 Hardware vulnerabilities	204
7.11.1 Hardware with software vulnerabilities	205
7.12 Resources for further and up-to-date information	205
7.13 Overview of the chapter	206
7.14 Reminder of learning outcomes	207
7.15 Test your knowledge and understanding	207
7.15.1 Sample examination questions	207
Appendix 1: The binary number system	209
Powers of 2	209
Appendix 2: Answers to Sample examination questions	211
Chapter 2	211
Chapter 3	212
Chapter 4	213
Chapter 5	214
Chapter 6	214
Chapter 7	215

Appendix 3: Answers to activities	217
Chapter 2	217
Chapter 3	218
Chapter 4.....	219
Chapter 5	220
Chapter 6.....	221
Chapter 7	222
Appendix 4: Sample examination paper: Part B	223
Appendix 5: Sample examination answers: Part B.....	227

Chapter 1: Introduction to the subject guide

Welcome to Introduction to computing and the internet, a Level 4 course in an undergraduate degree in computing, which is divided into two parts, Volume 1 and Volume 2, which correspond to Part A and Part B in the examination. In this introductory chapter we will look at the overall structure of the subject guide – in the form of a Route map – and introduce you to the subject, to the aims and learning outcomes, and to the learning resources available. We will also offer you some examination advice.

We hope you enjoy this subject and we wish you good luck with your studies.

1.1 Route map to the guide

The aim of the full course is to give you an appreciation of some of the basic issues and concepts in computer science and the internet; including hardware, software, the representation and transmission of information as well as security and legal issues.

Volume 1 considers the way modern digital computers work at the level of hardware and the operating system; and how information is represented and processed on individual computers. There are six chapters in total in Volume 1; one introductory chapter and five main content chapters, which are described below.

Chapter 1 introduces the subject and includes some general information about reading, learning resources, as well as some examination advice.

Chapter 2 gives a brief history of computers ending with the von Neumann computer architecture and discusses why binary numbers are used in computers.

Chapter 3 looks in some detail at the different forms of computer memory, and at how data is stored and accessed.

Chapter 4 considers the components of the central processing unit (CPU) and their functions; the fetch-execute cycle; and ways of improving computer performance.

Chapter 5 describes the functions and processes of operating systems.

Chapter 6 describes how numbers, characters (text) and other forms of data are represented in a computer.

Appendix 1: Answers to the sample examination question included with each chapter

Appendix 2: Answer to activities included with the chapters.

Appendix 3: Sample examination paper: Part A.

Appendix 4: Sample examination paper answers: Part A.

Volume 2 of the guide deals with the technologies and standards that enable computers to communicate and share information across a network. The protocols and technologies that support the internet are considered, together with security and legal issues.

Chapter 1 is the introductory chapter.

Chapter 2 explains why standards for internet working were developed and how they underpin the successful development of the internet. The chapter looks at networking models, focusing on the TCP/IP protocol suite.

Chapter 3 looks in detail at the Internet Protocol, the most important protocol of the Internet layer of the TCP/IP protocol suite.

Chapter 4 looks in detail at the major data structures and processes of the most important protocols of the Transport level of the TCP/IP protocol suite.

Chapter 5 considers the world wide web and email, briefly describes the operation of web servers and browsers and then discusses web authoring with HTML, XHTML and CSS.

Chapter 6 looks at the legal framework, including patent, copyright, and applicable UK law with some reference to US and EU law. Cybercrime is defined and discussed.

Chapter 7 considers computer security from a technological perspective, and discusses the vulnerability of networked systems to malicious exploits and unwanted intrusions. Professional, legal and social issues in computer security are also reviewed.

Note: Any section prefaced with 'An historical note' is for information only, and does not contain examinable material.

Appendix 1: The binary number system. This appendix gives details of what you should understand about the binary number system.

Appendix 2: Answers to the sample examination question included with each chapter.

Appendix 3: Answers to activities included with the chapters.

Appendix 4: Sample examination paper: Part B.

Appendix 5: Sample examination paper answers: Part B.

At the end of each chapter, a Learning outcomes section lists what you should be able to do. These two volumes of the subject guides introduce vocabulary, concepts and skills that you will need to pass the examination at the end of the course.

1.2 Glossary of key terms

Some chapters include a list of new terms at their start, in order to help you to understand the material in the chapter.

1.3 Introduction to the subject area

Computers are interwoven with the fabric of our existence. A basic understanding of their architecture, storage and representation of data is an important foundation of knowledge and understanding for all computing professionals. Leading on from this, how computers work together through networks, particularly the internet, is key to understanding the opportunities and challenges of the 21st century. Computers and the internet need to be seen and understood in technological, legal, social and ethical terms. Professionals also need to keep track of their wider working environment, so as to understand and anticipate both new opportunities for development, and emerging security threats.

This subject guide covers the second half of the course, as given in the final two course objectives, and the final three course learning outcomes below. The other objectives and learning outcomes are addressed in Volume 1. Hence this subject guide focuses on the protocols and technology underpinning the internet and the world wide web plus professional, legal and social issues in computing. More specifically, it answers the following five key questions:

1. How does the internet work in terms of protocols and standards?
2. How can a computer connect to the internet and send data to particular destinations?
3. How can data be transmitted and received across the internet without loss or corruption?
4. How is a web page written?
5. What are the professional, legal, security and social contexts in which the internet operates?

In explaining the answers to these questions, network models, internet protocols and standards, web authoring and legal, ethical, security and other issues will be discussed. This includes:

1. how standards and protocols for the internet are set and maintained
2. how network models provide an abstract model of a network and a standard for implementation
3. how important internet protocols such as TCP provide a standard for implementation, with implementation left to developers
4. how layering in network models describes an implementation of data encapsulation – also known as information hiding – that restricts the number of possible interfaces between network layers in order to design, develop and implement simply
5. how data encapsulation/layering is implemented in network protocols
6. how the most important protocols of the TCP/IP suite of protocols work
7. dynamic web technologies and how to write web pages with HTML and CSS
8. the legal framework including copyright, patents, data protection and other applicable civil and criminal law
9. computer security, emerging threats and new developments (in particular, the internet of things or 'IoT').

1.4 Syllabus

1.4.1 Volume 1

Computer architecture: von Neumann architecture, CPU, memory, I/O devices, data, address and control buses.

Data storage: main memory, storage including disk storage.

Central Processing Unit: the fetch-decode-execute cycle, instructions, clocks, cache memory, pipelining, CISC versus RISC.

Data representation: representing characters, numbers, images, movies, sound.

Operating systems: file management, process management, memory management, I/O management, network management.

Terminology: hardware, software, compiler, Unix, Windows, etc.

1.4.2 Volume 2

How the internet works: addressing and routing – URLs, domain names; IP addresses. Protocols: TCP/IP; HTTP; SMTP; POP; client-server model; web servers and browsers; search engines; electronic mail; file transfer.

Standards and regulation: role of W3C consortium; domain name registration; standards for SGML, HTML, XML.

Web page design and coding: HTML basics: document structure; links; URLs; fonts; colours; images; lists; tables and frames. Dynamic HTML: Javascript; style sheets. Basic overview of advanced technologies for dynamic and active web documents: client-side and server-side scripting.

Professional issues: security of networked systems: unauthorised access and malicious code, procedural and technical defences; legal framework. Data protection and privacy: principles of data protection; role of the Information Commissioner; rights of data subjects. Internet usage in the workplace: auditing and monitoring; intellectual property rights. Liabilities of ISPs: defamation; obscenity; secure communications.

1.5 Aims of this course

To give you, the student, an appreciation of some of the basic issues and concepts in computer science and the internet, including hardware, software, the representation and transmission of information and security issues.

1.6 Learning objectives for the course

- Give students the information they need to understand some of the basic computer science terminology.
- Provide an overview of how computers work by looking at issues such as: how computers are composed; how data is stored; how information is represented; how information is processed; and how computing operations are coordinated.
- Provide an introduction to the internet and the world wide web in terms of technologies, protocols, standards and applications.
- Provide an appreciation of professional, legal and social issues relating to networked computing.

1.7 Learning outcomes for students

On completion of this course, you should be able to:

- demonstrate understanding of some common, basic terminology in computer science
- explain the operation of modern digital computers at the level of hardware and the operating system
- demonstrate understanding of some common means of representing and storing information in digital format
- demonstrate a basic understanding of internet/www mechanisms, architecture and protocols and applications such as electronic mail, file transfer and web browsers
- code web pages with some dynamic features, using a text editor or web authoring tool
- describe professional, legal and social issues impacting on provision and use of internet services with particular reference to data protection, privacy, computer misuse and cybercrime.

1.7.1 Prior knowledge required

This subject guide does not assume any prior knowledge except for some mathematics and Boolean logic. In general, you are expected to understand the place value system in decimal numbers, and how it is applied to other number bases, such as 2 for binary and 16 for hexadecimal, and to understand binary numbers, powers of 2 and the relation between them.

Chapter 3 is the only chapter requiring significant computation in order to complete exercises and answer examination questions. You will be expected to be able to:

1. Understand powers of 2, and comprehend why it is that x bits gives 2^x unique numbers, even though the highest number that can be expressed with x bits is $2^x - 1$.
2. Convert binary to decimal and vice versa (for example, when converting dotted decimal IP addresses to binary octets and vice versa).
3. Understand and be able to apply a **bitwise AND**.

In Chapter 5 of the subject guide you will be expected to understand hexadecimal numbers, and the logic of converting between binary, decimal and hexadecimal (you will not be asked to perform any conversions or calculations in hexadecimal).

For an explanation of point 1 above, please also see Appendix 1. The following is an explanation of two topics that you will need to understand for Chapters 3 and 5: Bitwise AND and hexadecimal numbers.

1.7.2 Bitwise AND

AND: a Boolean operator, which given two inputs returns **true if they are both true and false otherwise**. In terms of binary numbers:

- 1 AND 1 is 1
- 0 AND 0 is 0
- 1 AND 0 is 0
- 0 AND 1 is 0.

Clearly, this reduces to: 1 AND 1 is 1; everything else is zero.

Bitwise AND: ANDing two binary numbers by ANDing a bit from one number with the bit in the corresponding position in the other number. For example, performing a bitwise AND on 1101 0001 and 1011 1111 gives:

	1	1	0	1	0	0	0	1
	1	0	1	1	1	1	1	1
AND	1	0	0	1	0	0	0	1

Table 1.1: Performing a bitwise AND on 1101 0001 and 1011 1111.

1.7.3 Hexadecimal

You are expected to understand **hexadecimal** notation.

Hexadecimals are numbers expressed **in base 16**. In the hexadecimal system, the digits 0–9 have the same meaning as in the decimal system; while letters are used as symbols in order to represent the numbers 10–15, as follows.

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Dec	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 1.2: Comparing the hexadecimal and decimal systems.

In order to convert from hexadecimal to decimal, we would use the same place value system as with a binary conversion, except that now our base b , the number we are raising to a certain power depending on its place value, is 16. Place value is the same as for any number system, the least significant digit is multiplied by b^0 to find what it contributes to the number expressed; and the most significant digit is multiplied by $b^{(n-1)}$ where n is the number of digits.

Example 1.1

The hexadecimal number 10F represents:

$$1 \times 16^2 + 0 \times 16^1 + 15 \times 16^0 = 1 \times 256 + 15 \times 1 = 256 + 15 = 271$$

Conversion between hexadecimal and binary

There is an obvious link between binary numbers and hexadecimal numbers, in that $16 = 2^4$. It turns out that this useful property can lead to an easy conversion between hexadecimal and binary – take each hexadecimal digit, and represent it with its 4-bit binary equivalent. Hence for 10F (271 in decimal) we have:

$$(1) \quad (0) \quad (15_{10} = F_{16})$$

$$0001 \quad 0000 \quad 1111$$

Giving 000100001111 or 100001111. You should be able to verify for yourself that 100001111 in binary does represent 271 in decimal, but refer to Volume 1 of this subject guide if you need to revise this.

Conversion between binary and hexadecimal

Similarly, to convert a binary number to hexadecimal, group it into 4-bit sections and convert each section to its hexadecimal equivalent. For example, $101101_2 (= 45_{10})$ would be converted by grouping it into 10 and 1101. Adding leading zeros we would have 0010 and 1101. Converting gives:

4-bit binary	0010	1101
Decimal	2	13
Hex	2	D

Hence 101101_2 is 2D in hexadecimal.

Checking, we find that $2D = 2 \times 16 + 13 \times 1 = 32 + 13 = 45$.

Converting a decimal number to hexadecimal

Converting a decimal number to hexadecimal can be done similarly to one of the methods for converting a decimal number to binary, by repeated division by the base.

Hence to convert 271 to hexadecimal we would repeatedly divide by 16, treated as integer division, with a numerator and a remainder. The division continues until the result of the division is zero.

Number to divide	Result of division by 16	Remainder
271	16	$15_{10} (= F_{16})$
16	1	0
1	0	1

The correct hexadecimal number is found by treating the final result of the division as the most significant digit in the hexadecimal number, and the first result of the division as the least significant. It is important to remember this, as it is quite anti-intuitive. Hence $271 = 10F$.

1.8 Overview of learning resources**1.8.1 The subject guide**

Chapters 2–5 in this guide start with a list of Essential reading and some Further reading. Chapters 6 and 7 have Further reading only.

The **Essential reading** section directs you to the textbook sections that you have to read. For some chapters you are given Essential reading from Comer.

Chapter 5 has Essential reading from the W3Schools online tutorials; while Chapters 5, 6 and 7 have no Essential reading textbooks as the main chapter content constitutes the Essential reading. The W3Schools site is for web developers, providing tutorials in all relevant technology.

Certain topics in the subject guide are covered in greater depth in the Essential reading given at the start of each chapter. You do not have to read every page of each textbook, but you should pay careful attention to those sections and chapters that are listed in the Essential reading.

The **Further reading** section gives chapters and sections you are advised to read from books and one web page (there is a list in Section 1.8.3). None of this suggested reading is compulsory, but it is designed to broaden and deepen your knowledge and understanding of the topics covered in that chapter. You are also encouraged to find other relevant books in libraries and search for relevant information on the internet.

1.8.2 Essential reading

The following book is recommended to support this course:

- Comer, D.E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) 6th edition Pearson new international edition [ISBN 9781292040813].

Some tutorials from the W3Schools initiative are given as Essential reading for Chapter 5: www.w3schools.com/default.asp. You will find a complete list in Chapter 5 of the subject guide.

Detailed reading references in this subject guide refer to the edition of the set textbook listed above.

New editions of one or more of these textbooks may have been published by the time you study this course. You can use a more recent edition of any of the books; use the detailed chapter and section headings and the index to identify relevant readings. Also check the VLE regularly for updated guidance on readings.

1.8.3 Further reading

Please note that as long as you read the Essential reading you are then free to read around the subject area in any text, paper or online resource. You will need to support your learning by reading as widely as possible and by thinking about how these principles apply in the real world. To help you read extensively, you have free access to the virtual learning environment (VLE) and University of London Online Library (see below).

Other useful texts for this course include:

- Casad, J. *Sams Teach Yourself TCP/IP in 24 Hours*. (Indiana: Pearson Education, 2017) 6th edition [ISBN 9780672337895].
- Castro, E. and B. Hyslop *HTML and CSS: Visual Quickstart Guide*. (San Francisco, CA: Peachpit Press, 2013) 8th edition [ISBN 9780321928832].
- Gillespie, A.A. *Cybercrime: Key Issues and Debates*. (Abingdon, Oxon; New York, NY: Routledge, 2015) [ISBN 9780415712217 (hbk); 9780415712200 (pbk); 9781315884201 (ebk)]. Since publication of the guide a 2nd edition (2019) has been published.
- Holt, J. and J. Newton (eds) *A Manager's Guide to IT Law*. (London: British Computer Society, The Chartered Institute for IT, 2011) 2nd edition [ISBN 9781906124755 (pbk); 9781780170039 (ebk)].
- Bott, F. *Professional Issues in Information Technology*. (BCS, The Chartered Institute for IT, 2014) 2nd edition [ISBN 9781780171807 (pbk); 9781780171821 (ebk)].

- Schneier, B. *Click Here to Kill Everybody: Security and Survival in a Hyper-connected World*. (New York; London: W.W. Norton & Company, 2018) [ISBN 9780393608885 (hbk)] [ASIN: B07BLMQKZK (ebk)].
- www.theregister.co.uk/2012/12/14/first_virus_elk_cloner_creator_interviewed/ – an interview with the man credited with creating the first computer virus.

1.8.4 Websites

- www.w3.org/ – the World Wide Web Consortium, the organisation that leads the development of new standards and protocols for the world wide web; an authoritative source of information on existing protocols and standards.
- www.icann.org/ – the Internet Corporation for Assigned Names and Numbers. Coordinates the internet's naming system, and has responsibility for 'IANA functions', described by ICANN as:

The IANA functions include the management of protocol parameters, Internet number resources and domain names.
- <https://pti.icann.org/> – Public Technical Identifiers. Responsible for coordinating the internet's unique identifiers at an operational level; namely, it is the body responsible for the operation of ICANN's IANA functions.
ICANN has overall responsibility for the internet's unique identifiers (IANA functions), delegates assigning and maintenance functions to IANA, and coordination of operational matters to PTI.
- www.iso.org/home.html – International Organization for Standardization.
- www.faqs.org/faqs/ – useful information including the internet's Request for Comments (RFC) documents that describe internetworking protocols.
- www.ietf.org/ – the Internet Engineering Task Force describes itself as 'the premier Internet standards body, developing open standards through open processes.' The IETF coordinates standards development through working groups. Final approval of RFC's is the responsibility of the internet Engineering Steering Group, which is composed of Area Directors of the IETF. See: www.ietf.org/about/groups/iesg/
- www.w3schools.com/ – useful information and tutorials on HTML, CSS and web authoring.
- <https://ico.org.uk/> – the office of the UK's Information Commissioner, the authority charged with upholding the public's right to data privacy as set out in applicable legislation.
- www.bcs.org/ – BCS, The Chartered Institute for IT, promotes the global IT profession and the interests of its workers. The BCS develops courses in IT subjects such as Data Protection and Software testing, leading to globally recognised certificates.
- <https://blog.avast.com/> – a useful source of information on current and emerging computer security issues.
- www.sophos.com/threat-center/threat-analyses/hoaxes.aspx – a good place to look for up-to-date information about virus hoaxes.
- www.sei.cmu.edu/about/divisions/cert/index.cfm – Carnegie Mellon University's Software Engineering Institute's CERT Division, developed from the very first computer security organisation established by DARPA in 1998.
- www.snopes.com/ – the internet's number one site for debunking online misinformation.
- www.hoax-slayer.net/ – another good hoax site, that also contains details of circulating emails and rumours that might seem false, but are actually true.

1.8.5 Online Library and the VLE

In addition to the subject guide and the Essential reading, it is crucial that you take advantage of the study resources that are available online for this course, including the VLE and the Online Library.

You can access the VLE, the Online Library and your University of London email account via the Student Portal at: my.london.ac.uk

Unless otherwise stated, all websites in this subject guide were accessed in June 2019. We cannot guarantee, however, that they will stay current and you may need to perform an internet search to find the relevant pages.

1.8.6 End of chapter Sample examination questions and Sample answers

To help you practise for the examination, we have included some end of chapter Sample examination questions with their scope limited to the learning outcomes of the chapter they are found in. Each question is similar in style and difficulty to actual examination questions.

You will have three hours for the examination and will be required to answer four questions. This gives you about 45 minutes per question, including reading and checking time.

Once you have attempted these Sample examination sections under timed conditions, you should check the Appendix at the end of the subject guide for the Sample answers and revise as appropriate.

1.9 Examination advice

Important: the information and advice given here are based on the examination structure used at the time this guide was written. Please note that subject guides may be used for several years. Because of this we strongly advise you to always check both the current programme regulations for relevant information about the examination, and the VLE where you should be advised of any forthcoming changes. You should also carefully check the rubric/ instructions on the paper you actually sit and follow those instructions.

The examination paper for the course **Introduction to computing and the internet** contains six questions in all, three from the material covered in Volume 1 and three from the material covered in Volume 2. The examination lasts for three hours. Therefore you have about 45 minutes on each of the four questions you choose to answer (two questions from each section). Each question is worth 25 marks, with 100 marks in total available on the paper. If you answer all three questions from any section, your third answer will not be marked. You will be asked to do some calculation, hence calculators are allowed.

It is good practice to read all of the questions before you start.

When you are asked to show all your working, it is important that you do so; students who only write down the final answer, even when correct, will lose marks.

For example, a question asks students to convert a decimal number into an IEEE 754 number – for 7 marks – and to show all their working. Student A writes down the correct answer. Student B writes down the wrong answer. Student A gets one mark, student B gets six marks. The marking scheme specifies one mark for the final answer, and six marks for the various stages needed to find the final answer. Student A wrote down no working, but student B wrote down all of theirs, and it was correct, with a small mistake made when translating all of their working out into a final answer.

When answering a question, you should try to be precise and try to cover all the relevant points. When asked to explain something, imagine that you are explaining it to someone whose knowledge of the subject is very limited. Always take note of the marks awarded for a question as this is likely to reflect both the amount of time the question is expected to take you and the number of elements in the answer.

You will find Sample examination questions at the end of most chapters. Use them as a tool to remember what you have read and in order to measure your progress. You will find sample answers in the appendix that should help you both in testing yourself, and in judging how to answer examination questions successfully.

Many aspects of the examination will recur, with changes, in different years. You can access previous papers and *Examiners' commentaries* on the VLE. Practising with past examination papers is probably the single best way to prepare in the weeks before the examination. You are strongly advised to practise under timed examination conditions.

Remember, it is important to check the VLE for:

- up-to-date information on examination and assessment arrangements for this course
- where available, past examination papers and *Examiners' commentaries* for the course.

1.10 Overview of the chapter

In this chapter, we have introduced the course and this volume of the subject guide, listing the aims and objectives of the course and outlining some practical aspects of working through the subject guide, the reading and other resources, before offering some advice on examination preparation.

1.11 Test your knowledge and understanding

1.11.1 A reminder of your learning outcomes

By the end of this course, and having completed the Essential readings and activities in both Volume 1 and Volume 2, you should be able to:

- demonstrate understanding of some common, basic terminology in computer science
- explain the operation of modern digital computers at the level of hardware and the operating system
- demonstrate understanding of some common means of representing and storing information in digital format
- demonstrate a basic understanding of internet/www mechanisms, architecture and protocols and applications such as electronic mail, file transfer and web browsers
- code web pages with some dynamic features, using a text editor or web authoring tool
- describe professional, legal and social issues impacting on provision and use of internet services with particular reference to data protection, privacy, computer misuse and cybercrime.

Chapter 2: Network models, protocols and standards

2.1 Introduction

Today the internet allows someone using an Android smartphone in Hong Kong connected to a wireless network, to communicate via for example, email or Voice Over Internet Protocol (VOIP), with someone using a Windows desktop computer connected by a physical wire to a Local Area Network (LAN) in Novosibirsk. Both users are accessing the internet as a medium of information transmission and exchange. Neither user need concern themselves with the very different ways that the hardware and software they are using stores and transmits data as a result of the TCP/IP protocol suite, which describes the protocols and standards on which the internet is based.

The TCP/IP protocol suite facilitated the rise to global dominance of the internet, and is used today by billions of people, many of whom might be surprised to learn that TCP/IP was developed in the 1970s. The technology implementing TCP/IP has changed a great deal since the 1970s. However, the protocols are distinct from their implementation, in the way that a recipe for chocolate cake is separate from baking it. You might use an electric or hand-powered whisk to beat together the ingredients; or you may instead do it by hand with a wooden spoon. You might bake the cake in a gas oven, or an electric one. The recipe is independent of the method you choose and the technology you use to make the cake does not change the recipe.

2.1.1 Aims of the chapter

This chapter aims to give you a brief overview of the history and development of the internet, together with a high level understanding of the TCP/IP protocol suite that underpins the internet's successful development and world wide reach.

2.1.2 Learning outcomes

By the end of this chapter, and having completed the Essential reading and activities, you should be able to:

- explain the distinction between networks and internets, and between logical and physical addresses of network components
- explain what is meant by protocols in the context of computer networking, and how protocols make it possible for applications on remote computers to communicate easily
- explain the concept of layering and describe in general terms the differences between the best-known layered models: OSI, TCP/IP and hybrid models.

For the purposes of this course you will mainly be expected to know about the Application, Transport and Internet layers of the TCP/ IP model. The OSI and hybrid models will not be discussed **after** this chapter and we will not be concerned with networking hardware or software layers that interface directly with the hardware. You will also be expected to have some awareness of the roles of international standards and regulatory bodies.

2.1.3 Essential reading

- Comer, D. E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) 6th edition Pearson new international edition [ISBN 9781292040813]. Chapter 1, *Introduction And Overview*.

2.1.4 Further reading

- Casad, J. *Sams Teach Yourself TCP/IP in 24 Hours*. (Indiana: Pearson Education, 2017) 6th edition [ISBN 9780672337895]:
 - **Hour 1:** What is TCP/IP
 - **Hour 2:** How TCP/IP Works
 - **Hour 3:** The Network Access Layer; section titled *Physical Addressing*.
- Comer, D.E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) Pearson new international edition [ISBN 9781292040813]. Chapter 4, *Protocol Layering*, Sections 4.1 to 4.10 inclusive.

2.1.5 References cited

- Noam Chomsky interviewed by David Barsamian:
https://chomsky.info/200408_/
- RFC 1700: www.ietf.org/rfc/rfc1700.txt

2.2 Glossary of key terms

- **Network:** a system with connected components.
- **ARPANET:** the forerunner of the internet, developed by the United States in the 1960s to connect geographically separate computers.
- **Internet:** a global connection of computer networks, with standardised (non-proprietary) protocols for information transfer. Once any network of networks was considered to be an internet, and the convention was to write the global internet with a capital 'i' to signify the difference. Now 'internet' has come to mean only the global internet, and the tradition of using a capital 'i' to distinguish the global internet from other networks of networks, is no longer necessary.
- **Protocol:** In general a protocol is a set of rules for carrying out a procedure or function of some sort. In computing terms a protocol is a set of rules for sending, receiving and exchanging data. These rules will use abstraction from their hardware/software implementation, in order to highlight key principles.
- **Implementation:** How something is put into practice. In computing terms, the hardware and software used to operate a standard, protocol or algorithm.
- **Standard:** an agreed level of service or performance that those subscribing to the standard are required to reach.
- **IP address:** a binary field used to uniquely identify internet-connected devices. Currently in use IP addresses are 32-bits, and belong to the IPv4 addressing scheme. IP addresses are conventionally transcribed as the decimal equivalents of each 8-bit section or octet. For example the binary address 10011110110111110000000101101100 would be divided into 4 octets, 10011110 11011111 00000001 01101100, and written as 158.223.1.108, in the **dotted decimal** format. The IPv4 scheme and its proposed 128-bit successor, the IPv6 scheme, will be discussed in the next chapter.

- **Domain names:** a translation of IP addresses into much easier to remember (for humans) names. A web browser will request a DNS server to translate a domain name into a machine-readable IP address.
- **Domain Name System (DNS):** a protocol that translates domain names to IP addresses and vice versa.
- **Data packet:** files and data packages sent over the internet are divided into chunks for transmission; these chunks are commonly referred to as data packets.
- **TCP:** the Transmission Control Protocol provides reliable, error-checked delivery of data via the internet.
- **UDP:** the User Datagram Protocol provides connectionless delivery of data over the internet. It sacrifices guarantee of delivery for speed of throughput.
- **IP:** the Internet Protocol is the most important protocol in the TCP/IP protocol suite, establishing how a data packet is addressed and routed, it is the backbone of the internet.
- **TCP/IP:** the protocol suite that the internet implements, named after its most important protocols.
- **Data encapsulation** (also known as data hiding), means that information is revealed strictly on a 'need to know' basis, and otherwise is hidden. Usually an interface is provided that prevents users from accessing or knowing implementation details. This is closely related to **abstraction**.
- **Abstraction:** a way of masking complexity in order to make obvious the key components of a system.
- **Layering:** an implementation of data encapsulation that separates the tasks of networking into layers. Adjacent layers only can communicate with each other, limiting the number of possible interfaces, and meaning that the details of implementation are hidden from non-adjacent layers.
- **Bitstream:** binary data sent through a transmission medium sequentially.
- **Proprietary:** refers to a system, process or invention that belongs to a private company or a corporation, and that the private entity has rights over, meaning that the use of the system, process or invention by others is restricted in some way.
- **ICANN:** The Internet Corporation for Assigned Names and Numbers. Responsible for coordinating and overseeing the internet's unique identifiers.
- **IANA functions:** Part of the responsibilities of ICANN, and including the management of protocol parameters, Internet number resources and domain names.
- **IANA:** The Internet Assigned Numbers Authority. IANA has the responsibility for IANA functions delegated to it by ICANN.
- **PTI:** Public Technical Identifiers. Responsible for co-ordinating the internet's unique identifiers at an operational level; this means that it is the body responsible for the operation of ICANN's IANA functions. See: <https://pti.icann.org/>

2.3 Computer networks and internetworking

A computer network, at its most basic, is a system, or set, of linked computers that can transmit, receive and/or exchange information. A computer network requires: (1) a transmission medium; (2) that each computer has a way of being

uniquely identified by other computers in the network; and (3) that computers share common standards for representing, transmitting and decoding data.

The internet is a 'network of networks'. Any computer network can interface with the internet via a router, normally a hardware device. A networked computer can either send data to another computer on the same local network, or send it to a router to be delivered to another network via the internet.

The global internet is the infrastructure of the world wide web, which will be discussed in Chapter 5 of the subject guide.

2.3.1 The ARPANET

In the 1960s there were few computers in use. Of those that did exist, many were in universities, often those taking money from the United States Department of Defence, through their Defense Advanced Research Projects Agency (DARPA), which historically funded scientific research of all kinds, as noted by Noam Chomsky:

it's kind of interesting to look at the records of DARPA, the advanced research agency of the Pentagon (previously ARPA). In, I think it was 1971, in the context of a lot of antiwar pressure, Mike Mansfield introduced legislation called the Mansfield Amendment, which required that military funding by the Congress be used for military purposes. You take a look at the ARPA and DARPA reports before and after. They're interesting. I did it once. Before that, they simply reported what they were doing, namely, creating the economy of the future. After that, the reports are divided sort of into two parts. The first part talks about possible military applications, which mostly are imaginary, and the second part is like the old reports: Here's what we're doing. It's the economy of the future.

See: <https://chomsky.info/200408/>

DARPA, www.darpa.mil/, was founded by President Eisenhower in the 1950s as ARPA, a response to Russia's successful launch of the first satellite, Sputnik. The United States was determined to take and keep the technological lead, and Eisenhower realised that funding research was the key. Over the years ARPA/DARPA has funded the research and development of such things as GPS, stealth technology and espionage carried out by psychics (allegedly a less successful project).

In the 1960s, ARPA funded the research and development of the ARPANET, designed to network geographically separated machines. Some commentators have alleged that the development of ARPANET was motivated by the military's desire to have a communication network that would survive a nuclear war, and that this is reflected in the design – the lack of a central authority combined with packet switching technology meant that parts of the network could drop out without affecting its functioning. Others have claimed that the motivation for ARPANET was the very small number of computers in the 1960s, leading to a desire among academics to link researchers to computers that they were geographically separated from. Whatever the truth the first message over the new network was transmitted in 1969.

2.3.2 Development of TCP/IP protocols

Moving into the 1970s, computing technology became more widely used, and commercial and research organisations began to develop the technology

for local area networks (LANs). In a large organisation different departments might use different networking technologies for their LAN, often proprietary, such that it might not be possible for networks even within the same organisation to communicate. The TCP/IP protocol suite was designed in the 1970s by Vint Cerf and Bob Kahn, funded by DARPA, and was intended to be **internetworking** technology, that is a way to connect networks to other networks, whatever their underlying hardware and software.

TCP/IP protocol suite was based on some simple ideas, as outlined in section 2.3.3.

2.3.3 Design principles of TCP/IP protocols

- Networks could connect without having to make any internal changes to design or implementation.
- There would be no hierarchy of control.
- Data would be transmitted in chunks, and lost data packages would be retransmitted.
- The connections to networks (later called routers) would not store information about data packets they forwarded; this would make transmission more fault tolerant. The network protocols would only specify how to transmit data between end nodes; the end nodes would perform all other functions.

As Casad notes (2017, **Hour 1: What is TCP/IP?** In the section *The Development of TCP/IP*):

The original decentralized version of the ARPAnet survives to this day in the design of the TCP/IP protocol suite [...] Two important features of TCP/IP that provide for this decentralized environment are as follows.

- **End-node verification:** The two computers that are actually communicating – called the end nodes because they are at each end of the chain passing the message – are responsible for acknowledging and verifying the transmission. All computers basically operate as equals and there is no central scheme for overseeing communications.
- **Dynamic routing:** Nodes are connected through multiple paths, and the routers choose a path for the data based on present conditions.

2.3.4 Introduction and success of TCP/IP protocols

In 1982 TCP/IP was complete, and became the protocol suite for the ARPANET. At the same time, the increasing use of LANs and personal computers meant an environment in which the early internet, the network of networks facilitated by TCP/IP, could grow. In the 1980s, most academic institutions used the internet, as did the military; internet service providers began to appear at the end of the decade; and by the mid-1990s the internet was fully open to commercial use. Business and personal use of the internet was facilitated by the development of the world wide web by Tim Berners-Lee in 1989, who also wrote the first web browser in 1990, the same year that the ARPANET, widely seen as the precursor to the internet, was decommissioned.

Today all commercially available operating systems contain TCP/IP implementation. Comer notes (2014, Section 1.2 *The TCP/IP Internet*):

Although TCP/IP technology is noteworthy by itself, it is especially interesting because its viability has been

demonstrated on a large scale. It forms the base technology for the global Internet that connects approximately two billion individuals in homes, schools, corporations and governments in virtually all populated areas of the planet.

Activity 2.1

Do some reading and research about Ethernet technology and answer the following questions:

1. What is Ethernet technology used for?
 2. What level of the TCP/IP network model does Ethernet technology belong to?
 3. How successful has the technology been and why?
 4. Briefly describe one problem the technology suffered from.
 5. Who developed Ethernet technology and in what decade was it developed?
-

2.4 Network models

Network models are ways of describing standards for networking, independently of their implementation.

2.4.1 Abstraction and data encapsulation

Recall that abstraction is a way of masking complexity in order to make obvious the key components of a system. In terms of computer systems, lower level details are hidden so that higher levels can be viewed more simply, in order to bring out their most important design features and processes. This naturally leads to a hierarchy of processes. Closely related to abstraction is data encapsulation, or data hiding, where access is provided via an interface, concealing certain data and implementation details.

2.4.2 Layering

Network models are given in layers, an abstraction incorporating data encapsulation. Networking protocols are assigned to a particular 'layer' or level in a vertical stack, and networking programs will only communicate with other programs in the same or adjacent layers. Layering simplifies design and development by keeping the number of interfaces to a minimum. The modular design makes it easier to develop software and hardware for different platforms as the software/hardware interfaces are restricted, simplifying the development process.

Two well-known networking models are the OSI model defined by the ISO (International Standards Organisation) and the Internet or TCP/IP model which specifically applies to internetworked systems.

2.4.3 The OSI model

The OSI model was proposed as a general standard which designers were expected to follow when developing communications software. In the OSI model networking functions are assigned to the OSI layers as follows:

Physical layer: software at this level manages the actual physical communication, in the sense of converting digital data into electronic or optical pulses and transmitting it across communications media such as cables or fibre-optic links.

Data link layer: interfaces between the physical and network layers, grouping data into frames of a few hundred or thousand bytes according to the requirements of network hardware.

Network layer: supports logical addressing and routing, determining how packets are routed from the source to the destination.

Transport layer: is responsible for accepting data from upper layers, segmenting it if necessary, passing the segments to the network layer and ensuring that all pieces arrive at the destination free of errors and in the correct sequence.

Session layer: allows users (applications) on different machines to establish sessions between them, so that they can synchronise and coordinate their exchanges of data.

Presentation layer: supports standard ways of transmitting data between computers with different data representations.

Application layer: contains a variety of protocols to support network applications such as file transfer and electronic mail.

The OSI model was developed before the protocols for each layer had been fully specified and it turned out that not all layers were equally important; in particular the session and presentation layers have very little content.

2.4.4 The TCP/IP model

The TCP/IP model is named after its two most important components, the **Transmission Control Protocol** and the **Internet Protocol**. TCP/IP was already under development when the OSI model was published and does not include all the OSI layers.

Network Access Layer: provides an interface with the physical network. The model has very little to say about this level: physical networks here could comprise nationwide networks such as Janet, a high-speed network for the UK research and education community, or Local Area Networks (LANs) connecting computers within a single enterprise.

Internet layer: responsible for making sure that any computer on the Internet can be uniquely identified via an IP address and that data packets can travel independently to the right destination.

Transport layer: provides error control, flow control and acknowledgement services. This means that software at this level is responsible for making sure that data arrives at its destination free of errors and in the correct sequence; and that messages are not sent either too quickly for the receiving system to process them; or so slowly that resources are not used efficiently.

Application layer: contains higher-level protocols for applications such as web browsers and servers, email clients, mail servers and name servers.

2.4.5 Headers and data packets in the TCP/IP model

In the TCP/IP model, layering is implemented by dividing packets into header and data sections. When an application wishes to transmit data to an application on another system, the data is divided into chunks and passed from the Application layer to the Transport, Internet and Network Access layers before being sent out into the internet. At each layer, the sending stack attaches a header containing information for software at the same stack layer in the receiving system.

Casad compares this process to being like a nested Russian doll (Casad 2017, **Hour 2: How TCP/IP Works**, section titled *Data Packages*). A data packet at the top of the stack is comprised of a header and data. When sent down to the next level, the original header and data becomes the data, and another header is attached. This continues as the data packet passes down the levels and finally out into the internet. Figure 2.3 in Casad (2017) is a graphical depiction of this process.

The receiving machine sends the data packet up the stack, starting of course, with the lowest level. This lowest level retrieves the header given by the corresponding level on the sending machine, and makes appropriate use of it, before passing the data to the next layer. This continues with each successive layer removing its header, until the data reaches the application it is intended for, at the top of the stack.

2.4.6 What is a data packet?

We have been referring to data packets, but it is important to note that at each layer the data packet has a different name. Casad 2017 (**Hour 2: How TCP/IP Works**, section titled *Data Packages*) describes them as follows:

- The data package created at the Application layer is called a **message**.
- The data package created at the Transport layer, which encapsulates the Application layer message, is called a **segment** if it comes from the Transport layer's TCP protocol. If the data package comes from the Transport layer's **User Datagram Protocol (UDP)** protocol, it is called a **datagram**.
- The data package at the Internet layer, which encapsulates the Transport layer segment, is called a **datagram**.
- The data package at the Network Access layer, which encapsulates and may subdivide the datagram, is called a **frame**. This frame is then turned into a bitstream at the lowest sub-level of the Network Access layer.

'Data packet' has become the generic name for data chunks at all levels of the network, but Casad goes on to note:

it is still worthwhile to consider that the different protocol packages have different names because they are actually quite different. Each layer has a different purpose, and each header contains different information.

Headers, data packets and data transmission will be discussed further in Chapters 3 and 4 of the subject guide.

2.4.7 Comparing the models

The TCP/IP model is more like a rationalisation of the way its distinct protocols fit together, and is not particularly suitable for describing non-TCP/IP networks. However, it is possible to define an approximate correspondence between the OSI and TCP/IP model, see Figure 2.1 below. This course will focus mainly on the TCP/IP model; however the OSI model is referred to in all standard textbooks and is considered to be a standard to which other models can be compared, so you need to be aware of the structure of the model and the general characteristics of its different layers.

TCP/IP model	OSI model
Application layer	Application layer
	Presentation layer
	Session layer
Transport layer	Transport layer
Internet layer	Network layer
Network access layer	Data link layer
	Physical layer

Figure 2.1: Comparing the OSI network model with the TCP/IP internetworking model.

Both the OSI and the TCP/IP models have their critics. One common complaint is that the OSI model has too many layers; while the TCP/IP model has too few. The OSI model's Session and Presentation layers appear unnecessary, while TCP/IP's Network access layer combines the OSI model's Data link and Physical layers, despite their different tasks:

- The OSI's **Data Link layer** is responsible for packaging data into frames and signalling the start and end of frames. The size of a frame is limited by hardware considerations.
- The OSI's **Physical layer** is concerned with transmitting raw bits of data through the transmission medium, which might be, for example, cables, fibre-optic connections or wireless links.

Data packaged in frames, and data as a stream of raw bits are clearly in quite different forms, yet in the TCP/IP model they coexist in the same layer, the Network Access layer. Casad notes (2017, **Hour 2: How TCP/IP Works**, in the section titled *Data Packages*):

the data package at the Network Access layer [...] is called a **frame**. This frame is then turned into a bitstream at the lowest sub-level of the Network Access layer.

2.4.8 The hybrid model

In practice a hybrid model is often implemented, based on TCP/IP layers, but replacing the Network Access layer with the Data Link and Physical layers of the OSI model.

TCP/IP model	OSI model	Hybrid model
Application layer	Application layer	Application layer
	Presentation layer	
	Session layer	
Transport layer	Transport layer	Transport layer
Internet layer	Network layer	Internet layer
Network access layer	Data link layer	Data link layer
	Physical layer	Physical layer

Figure 2.2: The hybrid network model is a combination of the OSI and TCP/IP models.

Note: In this subject guide we will not be concerned with anything below the Internet layer, so to keep things simple we will consider only the four-layer TCP/IP model from this point onwards.

2.5 What made the TCP/IP protocol suite so successful?

Fault tolerance means that faults do not cascade through a system, such that a small fault could cause serious problems, up to and including system collapse. Instead, the system recovers by switching to use other resources. Almost by definition fault tolerant systems need to be **non-hierarchical** in order to prevent faults from propagating through the hierarchy, increasing in seriousness as they go. Fault tolerant systems also need to avoid **choke points**, points where a blockage or congestion causes the entire system to slow down or stop. Systems that are centralised and/or hierarchical may contain points that all system resources must access in some way, and these points may become choke points under the right circumstances.

Packet switching technology was proposed and developed in the 1960s. Previously data, very much based on the model provided by a telephone call, was sent in a continuous stream. Packet switching divided data into discrete chunks; chunks from the same data package could then take different routes through a network. The ARPANET was an early adopter of packet switching, since its designers wanted to build a system that was fault tolerant. The designers of the TCP/IP protocol suite also adopted packet switching as a standard for implementation, meaning that the internet's most important protocols supported dynamic routing and decentralisation, thus making the internet fault tolerant and robust.

2.5.1 TCP/IP data transmission principles

- Data is divided into chunks for transmission (data packets).
- A data packet is transmitted by passing it from router to router, forming a chain.
- Routers forward data packets to other routers dynamically; that is, the router makes the decision of where next to send the data packet, based on current conditions.
- There is no central authority that knows where a data packet has to go and sees that it gets there; once a router has forwarded a data packet to the next link in the chain, it forgets it.

2.5.2 The TCP and UDP protocols

Two important protocols are defined for the Transport layer of the TCP/IP model:

TCP, the Transmission Control Protocol, is what is known as a connection-oriented protocol where both computers establish a connection before data is transmitted: this means that the sender will not send any data until it has confirmation that the receiver is ready to accept it. Sender and receiver monitor communications to ensure that all data is received and any faulty data is retransmitted, and close the connection 'gracefully' at the end of transmission. TCP implements error control by means of **positive acknowledgement with retransmission**: this means that the receiving system sends acknowledgment of all correctly received data packets, and the sender will retransmit a data packet if no acknowledgement is received. Unlike some other protocols there is no **negative acknowledgement**; that is, the receiver will never request retransmission of a corrupt or missing data packet.

UDP User Datagram Protocol is a 'connectionless' unreliable protocol where no connection needs to be established prior to data transmission and speed is more important than accurate delivery. The destination computer does not return any status or acknowledgement information to the source.

These two protocols are suitable for different applications. For example, when broadcasting voice or image data you do not want to wait until every system on the network has confirmed it is ready to receive the message, nor do you want every system to send an acknowledgement when it gets the message.

2.6 Key features of the TCP/IP protocol suite

Some of the key features of the current TCP/IP protocol suite are listed below.

2.6.1 Packet switching

With packet-switching the data stream is divided into discrete chunks which are transmitted independently through the network quite possibly by different routes. This allows for faster transmission if different channels are used in parallel, prevents any system tying up a transmission medium by sending long files in

a continuous stream and allows for data to be switched to another path if a router becomes unavailable.

2.6.2 Dynamic routing

Different routes from source to destination may be available and routers can choose a path for the data based on present conditions.

2.6.3 Physical addresses

Every component connected to a network has a static address, which is a physical identifier for the device. A physical address may be burned into the network adaptor hardware at the factory, although more recently some network adaptors allow system administrators to configure physical addresses for devices or to assign addresses dynamically when a device starts up. The 48-bit physical address may be referred to as a **MAC (Media Access Control)** address or, given the popularity of Ethernet LAN technology, as an **Ethernet** address. An international registry ensures that each MAC/Ethernet address is unique, but there is no guarantee that two computers on the same LAN will have any part of this address in common. While MAC/Ethernet addresses are unique, configurable and dynamic addresses will only be unique on a given network; that is to say, other networks may have the same internal addresses for components.

Since static addresses are of differing formats and are not distributed in a hierarchical scheme which would make it easy to route a message to the right destination, they cannot be used as unique identifiers for internet connectivity. Hence the need for logical addressing to relate static physical addresses to IP addresses such that each computer connected to the internet can be uniquely identified in a hierarchical scheme allowing for fast identification.

2.6.4 Logical addresses

Logical addressing is a uniform hierarchical addressing format, which allows any computer connected to the internet to be uniquely identified by its IP address, not only by the source computer but by intermediate systems which forward the data packets.

The addressing protocols in TCP/IP provide for:

- **uniform format:** IP addresses in current use are all 32-bit words divided into portions identifying the network and host according to universally recognised conventions
- **uniqueness:** no two systems can connect to the internet with the same address at the same time, though addresses can be reused by different computers and a computer may use different addresses each time it connects
- **address resolution:** transmission of messages within a local network, or from one router to another, requires that IP addresses can be translated into physical/static addresses and vice versa.

2.6.5 Name resolution

The name doc.gold.ac.uk is probably more meaningful and memorable than 158.223.1.108, though both of these refer to the same computer. The name makes it obvious that the computer belongs to a UK academic institution, and you could make a good guess that gold is short for Goldsmiths College and doc is an abbreviation for 'Department of Computing'. None of this information is encoded in the IP address. TCP/IP provides mapping of domain names to

numeric addresses known as name resolution, carried out by special-purpose servers called name servers.

2.7 The client/server model

Since TCP/IP takes care of the hard problems – such as locating the destination host, establishing connections, making sure that data is received in the correct order and free from errors – coding internet applications turns out to be rather straightforward.

The key concept in internet programming is the client-server model. Client and server processes operate on machines that can communicate through a network. The client requests a connection to the server when it needs the service that the server provides. The server listens for and accepts connection requests, and listens for requests from connected clients. When the server hears a request it fulfils it and sends a response to the client. Clients may disconnect when they have what they need. Examples of servers are: web servers, DNS servers; examples of client applications are web browsers, email clients.

2.7.1 Port numbers and ‘well-known’ ports

Port numbers were developed as part of the ARPANET, as a way of connecting to particular services on remote machines. There are no literal ports on these machines; port numbers are a logical construct, used to indicate what service the client wants to access at the end point of a logical connection. Hence port numbers are entirely arbitrary. IANA keeps a record of ‘well-known’ port numbers, meaning port numbers that are assigned to certain ‘privileged’ services. Network administrators do not have to use the well-known port numbers, although it clearly makes network communication a lot simpler if they do. RFC 1700, page 15, justifies the use of well-known ports with:

Ports are used in the TCP [RFC793] to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the “well-known port”.

See: www.ietf.org/rfc/rfc1700.txt

Ports are primarily used for Transport layer protocols such as UDP or TCP. A client connects to a server at a particular port number in order to request the service offered by the server. Ports are accessed by appending the port number to the IP address. For instance, a client on a remote computer can contact the FTP server on doc.gold.ac.uk by appending the ‘well-known’ FTP port number, 21, to the machine’s IP address giving 158.223.1.108.21. Other well-known port numbers are: 25 (SMTP) for sending mail, 110 (POP3) for retrieving mail, 80 (HTTP) for transfer of web documents.

Once a connection has been established between the client and server applications, requests and responses are passed between them. Applications generally have a fairly small repertoire of request and response types. For instance, an email client may send a series of requests to an SMTP (Simple Mail Transfer Protocol) server to identify the sender and recipient of a mail message, and to announce the intention to start transmitting data.

Port numbers will be discussed further in Chapter 4 of the subject guide.

2.8 Standards and regulation

In order to support the success and continued development of the internet, it is clear that bodies that propose, coordinate and maintain standards and protocols are necessary.

2.8.1 The Internet Engineering Task Force and RFCs

Authoritative, if sometimes terse and technical, descriptions of internet protocols can be found in documents known as RFCs (Requests for Comments). RFCs can comprise authoritative specifications of protocols, and are also how new internet protocols and standards are consulted on, developed and agreed.

The Internet Engineering Task Force – www.ietf.org/ – describes itself as ‘the premier Internet standards body, developing open standards through open processes’. The IETF coordinates standards development through RFC working groups, which develop and publish RFCs for comment. Final approval of RFCs is the responsibility of the Internet Engineering Steering Group, which is composed of Area Directors of the IETF. See: www.ietf.org/about/groups/iesg/ for more information.

2.8.2 The Internet Corporation for Assigned Names and Numbers (ICANN)

ICANN has a crucial role to play in maintaining a stable and successful internet by coordinating and overseeing its unique identifiers. See www.icann.org/

ICANN has responsibility for ‘IANA functions’, described by ICANN as:

The IANA functions include the management of protocol parameters, internet number resources and domain names.

ICANN has overall responsibility for the Internet’s unique identifiers (IANA functions), delegates assigning and maintenance functions to IANA and coordination of operational matters to PTI.

IP numbers, allowing unique identification of computers connected to the internet, are an obvious example of the need for oversight and coordination in order to maintain the proper functioning of the internet. Port numbers are another example of the coordination of numbers. The first proposal for a register of port numbers for well-known services was made in RFC 349, published in 1972. This document has been superseded many times, since, as the internet has developed new services have become ubiquitous, demanding their own dedicated port numbers, for example HTTP, now assigned port 80, did not exist when RFC 349 was published. Today the first 1,024 numbers are reserved for well-known services, with the Internet Assigned Numbers Authority (IANA), a part of ICANN, maintaining a list, see: www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

2.8.3 Links to other standards and protocols

www.iso.org/home.html – the International Organization for Standardization.

www.faqs.org/faqs/ – useful information including the internet’s RFC (Request for Comments) documents that describe internetworking protocols.

www.w3.org/ – the world wide web Consortium, the organisation that leads the development of new standards and protocols for the world wide web; an authoritative source of information on existing protocols and standards.

Activity 2.2

Find **RFC 793** on the internet.

1. What protocol is the RFC describing?
2. Why does the RFC reference the Defence Advanced Research Projects Agency?

Find **RFC 2468**.

3. What is unusual about RFC 2468?

Find **RFC 349**.

4. What is the RFC proposing?
5. What relation does RFC 3232 have to RFC 349?

Find **RFC 1812**. Read up to the end of the bullet points on page 10, and answer the following questions:

6. What technology is RFC 1812 concerned with? Briefly summarise the purpose of the RFC
7. The final paragraph of Section 1.1.1 states:

Under many of the individual topics in this memo, there is parenthetical material labeled DISCUSSION or IMPLEMENTATION. This material is intended to give a justification, clarification or explanation to the preceding requirements text. The implementation material contains suggested approaches that an implementor may want to consider. The DISCUSSION and IMPLEMENTATION sections are not part of the standard.

Why are the IMPLEMENTATION sections not part of the standard?

2.9 Overview of the chapter

In this chapter we gave an overview of the historical development of the internet and its infrastructure, the TCP/IP protocol suite, which is named for its most important protocols: the Transmission Control Protocol and the Internet Protocol. We considered how network models provide a level of abstraction, simplifying the design and development of networks, examined some network models, including the TCP/IP model. We considered some reasons for the success of the TCP/IP protocol suite, and noted its main features, discussed the client/server model of network computing then briefly considered the bodies responsible for coordinating and developing standards for the global internet.

2.10 Reminder of learning outcomes

Having completed this chapter, and the Essential reading and activities, you should be able to:

- explain the distinction between networks and internets, and between logical and physical addresses of network components
- explain what is meant by protocols in the context of computer networking, and how protocols make it possible for applications on remote computers to communicate easily
- explain the concept of layering and describe in general terms the differences between the best-known layered models: OSI, TCP/IP and hybrid models.

2.11 Test your knowledge and understanding

2.11.1 Sample examination questions

- a.** (i) What does packet switching mean? [2 marks]
1. Data is transmitted in chunks with dynamic routing; once a router has forwarded the data packet its memory is cleared and it is ready for other tasks.
 2. Data packets that do not arrive at their destination are re-transmitted.
 3. Data is encapsulated before transmission.
 4. None of the above.
- (ii) The internet has no central authority because: [2 marks]
1. The original designers adopted decentralised technology as it was fault tolerant.
 2. A central authority can be destroyed, with potential catastrophic loss of function.
 3. The adoption of packet switching as a design standard made a central authority unnecessary.
 4. All of the above.
- (iii) ICANN stands for: [2 marks]
1. Internet Control of Assigned Names and Numbers.
 2. Internet Corporation for Assigned Names and Numbers.
 3. Internet Coordination of Assigned Names and Numbers.
 4. Internet Corporation for Appropriate Names and Numbers.
- b.** (i) Describe the client-server model of internet computing. [6 marks]
- (ii) Explain the use and purpose of port numbers in the client-server model of internet computing. [3 marks]
- c.** (i) Explain the advantages of layering in network models. [6 marks]
- (ii) List the levels of the TCP/IP network model, and describe how headers implement layering in the TCP/IP network model. [4 marks]

Notes

Chapter 3: The Internet layer and the IP protocol

3.1 Introduction

This chapter considers in some detail the IP protocol of the Internet layer, and its major responsibilities for:

- **The IP datagram header:** defining the data packet format for the internet.
- **Routers and forwarding:** how routers forward data packets across the internet, including routing tables and next-hop forwarding.
- **Processing data packets and the Internet Control and Messaging Protocol (ICMP):** rules for hosts and routers in processing, discarding and sending error and diagnostic messages about data packets via ICMP.
- **The IP logical addressing scheme:** maintaining and supporting the logical, hierarchical addressing scheme that allows every machine connected to the global internet to be uniquely identified.

The chapter also considers the Internet Protocol version 4 (IPv4) addresses, and explores why IPv4 addresses are still in use, despite officially running out in 2011.

3.1.1 Aims of the chapter

This chapter aims to help you to:

- develop a good understanding of the IP protocol and its importance in the architecture of the internet
- demonstrate understanding of routing at a high level of abstraction, and describe in some detail how a router processes, forwards and discards datagrams
- develop an understanding of the logical, hierarchical addressing scheme for the internet and why the current IPv4 scheme is being replaced.

3.1.2 Learning outcomes

By the end of this chapter, and having completed the Essential reading and activities, you should be able to:

- explain in general terms how information in the IP header is used in routing
- identify different classes of IP addresses
- break down an IP address into network, subnet and host IDs using a subnet mask
- list the factors contributing to a shortage of IP addresses and some techniques that have been proposed to alleviate it.

3.1.3 Essential reading

- Comer, D.E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) 6th edition Pearson new international edition [ISBN 9781292040813]. Chapter 5, Sections 5.1–5.12 and Chapter 7, Sections 7.1–7.7.

3.1.4 Further reading

- Casad, J. *Sams Teach Yourself TCP/IP in 24 Hours*. (Indiana: Pearson Education, 2017) 6th edition [ISBN 9780672337895]:

- **Hour 4:** The Internet Layer
- **Hour 5:** Subnetting and CIDR
- **Hour 13:** IPv6: The Next Generation.
- Comer, D.E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) 6th edition Pearson new international edition [ISBN 9781292040813].
 - Section 5.13 *IPv4 CIDR Blocks Reserved For Private Networks*
 - Section 5.14 *The IPv6 Addressing Scheme*
 - Section 7.8 *Datagram Type of Service and Differentiated Services*
 - Section 7.10 *Datagram Size, Network MTU, and Fragmentation*
 - Chapter 8 *Internet Protocol: Forwarding IP Datagrams*
 - Chapter 9 *Internet Protocol: Error And Control Messages (ICMP) Sections 9.1–9.4 inclusive.*

3.1.5 References cited

- **RFC 760**
<https://tools.ietf.org/html/rfc760>
- **RFC 791**
<https://tools.ietf.org/html/rfc791>
- **RFC 950**
<https://tools.ietf.org/html/rfc950>
- **RFC 2460**
www.ietf.org/rfc/rfc2460.txt

3.2 Glossary of key terms

- **Data packet:** generic term meaning a discrete chunk of data transmitted across the internet.
- **Header:** metadata attached to data at each level of the TCP/IP network model.
- **Data encapsulation in the TCP/IP network model:** Levels of the TCP/IP model encapsulate data packets from the level above by adding a header to the data packet received. For example, when the TCP protocol implementation receives a **message** from the Application layer the protocol implementation adds a header to the **message** to create a TCP **segment**, hence the Application layer's **message** becomes the payload of the Transport level's **segment**. Layers therefore do not need to read or understand their data packet's payload.
- **Message:** the data packet created at the Application layer.
- **Segment:** the data packet created in the transport layer by the TCP protocol that encapsulates the Application layer's message.
- **Datagram:**
 - the data packet created in the transport layer by the UDP protocol, which encapsulates the message received from the Application layer
 - the data packet created by the Internet layer. This layer encapsulates segments and datagrams it receives from the Transport layer.

Note that Comer, Section **10.4 UDP Message Format** calls the UDP datagram the **user datagram** in order to avoid confusion with the **IP datagram** in the Internet layer.

- **Frame:** the data packet created at the Network Access layer. The frame encapsulates the IP datagram, possibly first subdividing it. The frame is what is actually transmitted over the hardware of the network or internet.

The above four definitions mean that the message is the payload of the Transport level datagram or segment, the datagram or segment is the payload of the Internet level IP datagram and the IP datagram is the payload of the frame.

- **MTU:** networks place an upper limit on the size of data that can be transmitted across the network in one frame. This limit is known as the **maximum transfer unit** or the **maximum transmission unit**. The MTU varies from network to network.
- **Fragmentation:** in order to forward an IP datagram to a router on a network that cannot accept such a large chunk of data, the Network Access layer may divide the datagram into two or more frames. This process is known as fragmentation.
- **Host:** on the internet a machine that can communicate with other machines connected to the internet (which may be, for example, a server, a printer, a smartphone, a vehicle or a domestic appliance such as a fridge) is a host.
- **Router:** a machine connected to the internet that forwards data packets.
- **Time-to-live:** if a destination cannot be found, then a data packet could in theory be forwarded as long as the internet exists. To prevent this, data packets have a time-to-live number. This number is decremented by one each time the packet passes through a router. When the number reaches zero, the data packet is discarded and an ICMP message is sent to the source IP address.
- **Direct delivery:** the delivery of a datagram from one machine on a network, to another machine on the same network. This might be a router delivering a datagram received from the internet, or a host machine directly delivering a datagram to another host.
- **Indirect delivery:** when a machine sends a datagram to a router, which then forwards the datagram on towards its final destination.
- **Octet:** a group of eight.
- **IP address:** a binary field used to uniquely identify internet-connected devices. IP addresses currently in use are 32-bits, and belong to the IPv4 addressing scheme. IP addresses are conventionally transcribed as the decimal equivalents of each 8-bit section, or octet. For example, the binary address 10011110 11011111 00000001 01101100 would be written as 158.223.1.108; this is done to make addresses more human readable. The IPv4 scheme and its proposed 128-bit successor, the IPv6 scheme, will be discussed in this chapter.
- **The Internet Control Message Protocol (ICMP):** Internet layer protocol for sending error and diagnostic messages, originally defined in **RFC 792**. ICMP supports the IP protocol in sending error and diagnostic messages. For example, if a router cannot deliver a data packet because the router cannot determine where next to forward the data packet, then it sends an error message to the source IP address via ICMP.
- **AND:** a Boolean operator, which given two inputs returns true if they are both true and false otherwise. In terms of binary numbers:
 - 1 AND 1 is 1
 - 0 AND 0 is 0

- 1 AND 0 is 0
- 0 AND 1 is 0.
- **Bitwise AND:** ANDing two binary numbers by ANDing a bit from one number with the bit in the corresponding position in the other number.

3.3 The Internet layer and the IP protocol

The internet is composed, at the simplest conceptual level, of hosts and routers. A router forwards IP datagrams, and a host sends and receives IP datagrams. A router cannot be a host, and a host should not be configured to also be a router, as the protocols do not support this. Provided that the IP protocol maintains a logical addressing scheme, and that the host is visible to the internet, routing is normally quick and efficient. If a datagram is found to be undeliverable for some reason, the router discards the packet, and sends a message to the source host. The datagram is defined by the Internet Protocol (IP), and error and diagnostic messages are sent via ICMP – the Internet Control and Messaging Protocol. Both IP and ICMP belong to the Internet layer of the TCP/IP network model.

Recall that the Internet layer is responsible for making sure that any computer on the internet can be uniquely identified via an IP address and that data packets can travel independently to the right destination. Hence its core protocol is the Internet Protocol, and this is reflected in the name of the TCP/IP internetworking model.

Comer (2014, Section 7.4 *Principles Behind the Structure*) describes the TCP/IP model as providing ‘three sets of services’:

Internet protocols are designed around three conceptual levels of service. A connectionless service at the lowest level matches underlying hardware well, a reliable transport service provides service to applications, and a variety of applications provide the services that users expect.

So we can view the TCP/IP model as having three levels of service: the top level provides applications; the middle level a reliable delivery service; and the bottom level provides a connectionless packet delivery service. The Internet layer provides the software for this connectionless packet delivery service, and the Network Access layer the hardware. In this chapter we will focus on the Internet Protocol, looking in some detail at how it provides a connectionless delivery service.

3.3.1 Unreliable, best effort, connectionless delivery

The packet delivery system provided by the IP protocol is an unreliable, best effort, connectionless packet delivery system.

- **Unreliable:** there is no guarantee that any one particular packet will be delivered.
- **Best effort:** the system will discard packets that cannot be delivered. Packets are not discarded arbitrarily, but by following certain rules. Hence all that is promised is a credible attempt at delivering every packet, with no promise that delivery will actually be made.
- **Connectionless:** Each packet in a data stream is treated as an independent unit, with no connection to previous or subsequent packets in the same stream. Hence packets in the same stream may follow different paths to the same destination, they may be delivered out of order; some may not be delivered at all (see Section 3.5 in the subject guide for more about packet delivery).

3.3.2 The responsibilities of the IP protocol

Comer (2014, Section 7.6 *Purpose And Importance of The Internet Protocol*) describes the Internet Protocol as providing three specifications:

The Internet Protocol provides three important specifications. First, IP defines the basic unit of data transfer used throughout a TCP/IP internet. Thus, it specifies the exact packet format used by all data as the data passes across an internet. Second, IP software performs the *forwarding* function, choosing a path over which a packet will be sent. [...] Third [...] IP includes a set of rules that embody the basis of unreliable delivery. The rules characterize how hosts and routers should process packets, how and when error messages should be generated, and the conditions under which packets can be discarded.

In addition to this, Casad (2017, **Hour 4: The Internet Layer**, heading *Addressing and Delivering*) discusses the IP protocol's role in the logical, hierarchical addressing scheme that allows every device connected to the internet to be uniquely identified:

Unfortunately, on a routed network, it is not possible to deliver data by physical address. The discovery procedures required for delivering by physical address do not work across a router interface. Even if they did work, delivery by physical address would be cumbersome because the permanent physical address built in to a network card does not allow you to impose a logical structure on the address space.

TCP/IP therefore makes the physical address invisible, and instead organizes the network around a logical, hierarchical addressing scheme. This logical addressing scheme is maintained by the IP protocol at the Internet layer.

The addressing scheme makes hosts visible on the internet, and its logical and hierarchical structure makes routing between hosts computationally simple and fast. The IP protocol only needs to know destination IP addresses in order to deliver to the right host. Data packets are forwarded from router to router until they reach their destination, or are found to be undeliverable.

3.3.3 An historical note

The RFCs specifying the original definition of the IP protocol for the ARPANET emphasised the protocol's responsibilities for 'addressing and fragmentation'. **RFC 760** from 1980 (<https://tools.ietf.org/html/rfc760>) and **RFC 791** from 1981 (see: <https://tools.ietf.org/html/rfc791>) both give the definition of the IP protocol as follows:

The internet protocol implements two basic functions: addressing and fragmentation.

The internet modules use the addresses carried in the internet header to transmit internet datagrams toward their destinations. The selection of a path for transmission is called routing.

The internet modules use fields in the internet header to fragment and reassemble internet datagrams when necessary for transmission through "small packet" networks.

RFC 760 and **RFC 791** define an ‘internet module’ as a local implementation of the IP protocol as follows:

The model of operation is that an internet module resides in each host engaged in internet communication and in each gateway that interconnects networks. These modules share common rules for interpreting address fields and for fragmenting and assembling internet datagrams. In addition, these modules (especially in gateways) have procedures for making routing decisions and other functions.

3.4 The IP datagram header

At the Internet layer, the data packet from the Transport layer is encapsulated into an IP datagram, with the addition of a header, which can be conceptualised as:



Figure 3.1: IP datagram.

The IP datagram payload is most likely to be a TCP segment, or a UDP datagram. Whenever you see the word ‘datagram’ used without qualification, it is safe to assume that it is the IP datagram that is being discussed.

When a router receives a data packet, it takes information from the header, and consults its routing table to make a decision about whether to forward, accept or discard the packet. Note that forwarding and discarding packets will be considered in more detail in Section 3.5.6 Processing and discarding packets.

0	4	8	16	24	31
VERSION	HEADER LENGTH	SERVICE TYPE	TOTAL LENGTH		
IDENTIFICATION			FLAGS	FRAGMENT OFFSET	
TIME-TO-LIVE		PROTOCOL	HEADER CHECKSUM		
SOURCE IP ADDRESS					
DESTINATION IP ADDRESS					
IP OPTIONS (IF ANY)				PADDING (IF NECESSARY)	
PAYLOAD					
.					
.					
.					

Figure 3.2: Fields of the IPv4 datagram header.

Figure 3.1 shows the format of a header for IPv4, the version of the IP protocol that we will consider in detail. Casad 2017 (**Hour 4: The Internet Layer**, section titled *Addressing and Delivering*) notes:

The most common version of IP is IPv4, although the world is theoretically in transition to a new version of IP known as IPv6.

3.4.1 Fields of the IPv4 datagram header

The meanings of the fields shown in Figure 3.1 are listed below.

Version: version of the IP protocol that determines how to interpret the header. Currently the only possible values are 4 (0100) or 6 (0110). The sender and receiver check the version field to make sure that they agree on the version of the IP protocol they are using. Intermediate routers also check it. Since the IPv6 header also has the first 4 bits for the version, routers and the receiver can easily check and reject datagrams they do not have the software to interpret.

Header length: a 4-bit field that gives the length of the header as a number of 32-bit words. Comer (page 124) states that:

all fields in the header have fixed length except for the *IP OPTIONS* and corresponding *PADDING* fields. The most common datagram header, which contains no options and no padding, measures 20 octets and has a header length field equal to 5 [0101].

Service type: because this 8-bit field, originally intended to indicate priority, was never much used, its meaning has been defined and redefined over the years of use of the IP protocol, in an attempt to find a use that would have widespread application. Comer (2014) discusses its current use in Section 7.8 *Datagram Type Of Service And Differentiated Services*.

Total length: this 16-bit field gives the size of the entire datagram, including header and payload, in bytes. Since the field is 16 bits, the maximum size is $2^{16}-1$ or 65,535 bytes (for an explanation of why this number is $2^{16}-1$ and not 2^{16} , see Appendix A). Comer (2014, Section 7.7.1 *IPv4 Datagram Format*) notes:

For most applications the limit does not present a problem. In fact, most underlying network technologies use much smaller frame sizes.

Identification: A unique number given to each datagram in a stream of datagrams. Typically these numbers are sequential, that is a datagram immediately following will have the identification number of the immediately preceding datagram incremented by one. When datagrams are fragmented each fragment has a copy of the Identification number, allowing fragments to be grouped for reassembly.

Flags, Fragment Offset: These fields, together with the **Identification** field, allow the datagram to be divided into two or more frames at the Network Access layer, and successfully reassembled. Note that networks have a maximum transfer unit, or MTU. If a router needs to forward a datagram to a network with an MTU that is too small to accept the datagram, the router will divide the data into separate frames for transmission, a process known as fragmenting. Comer (2014) discusses this in more detail in Section 7.10 *Datagram Size, Network MTU, and Fragmentation*. Fragmented datagrams are reassembled only when they reach their destination.

Time-to-live: originally the number of seconds that the datagram was permitted to be alive on the internet, before being discarded if it had not reached its destination. Routers were required to decrement the number. The sender would assign the limit, which was intended to prevent 'zombie' datagrams congesting the internet. The first problem with this was that routers have no mechanism that allows them to know exactly how long a datagram was in transmission from the previous router. The second was that as technology developed routers became faster, hence decrementing in whole seconds did not make sense when datagrams could be forwarded in

milliseconds. Now the number is interpreted as a 'hop count', which is what the IPv6 header calls this field. Each time a datagram is forwarded counts as one hop. Hence each router is required to decrement the time-to-live number by one. If the time-to-live reaches zero before the packet is delivered, the packet is discarded and a diagnostic message sent to the source IP address via ICMP.

Protocol: this 8-bit field has a number that uniquely identifies the Transport layer protocol that created the payload. This will typically be TCP ($6_{10}/0000\ 0110_2$) or UDP ($17_{10}/0001\ 0001_2$) but other values are possible. There is a list of protocol numbers, managed and maintained by IANA, in order to enforce internet wide agreement:

www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

Header checksum: this 16-bit field is computed by treating each field of the header as a number (the checksum field contributes zero). Routers re-compute the number as does the receiving host, to check for data corruption. If the checksum does not match the value computed by the router then the datagram has been corrupted. Corrupted datagrams are discarded and an ICMP message sent to the originating host. The checksum only applies to the header, not the payload, which has its own checksum. Note that any change to the values in the header mean that the checksum must be recalculated, thus routers must recalculate the checksum after decrementing the 'time-to-live' field.

Source IP address/Destination IP address: these 32-bit fields do not change as the datagram is forwarded. The source IP address is always the IP address of the originating machine, and the destination is always the IP address of the host that the datagram is intended for. Intermediate router IP addresses are not included in the datagram.

IP options: This field is often unused. Comer (2014, Section 7.14 *Optional IP Items*) notes that:

In practice, few datagrams on the global Internet include optional items. Many of the options in the standards are intended for special control or for network testing and debugging. Options processing is an integral part of the IP protocol: all standard implementations must include it.

Padding: a variable number of bits used to make sure that the IP options field comes out to a multiple of 32.

3.5 Routing

If you are trying to find your way around a strange city and ask one of the locals for directions, they might give you various different answers. If you happen to meet someone who has an encyclopaedic knowledge of the city, they may describe the exact route you need to take: which streets to follow, how many blocks to count before turning left or right and so on. On the other hand, your informant may only have a vague knowledge of the city's geography and will only be able to point you in the right direction. After going in this direction for a while you might ask someone else, and so on until you get close enough to your goal that someone can point it out for you.

Routing on the internet is more like the second scenario: there is no one system that holds a map of all the connections on the internet, but there are thousands of systems called routers that know about their neighbourhood and what direction to travel to get to other neighbourhoods.

Comer (2014, Section 12.2 *The Origin of Forwarding Tables*) describes routing as follows:

Each router attaches to two or more physical networks and forwards IP datagrams among them, accepting datagrams that arrive over one network interface and sending them out over another interface. [...] In the general case, a datagram travels from router to router until it reaches a router that attaches directly to the same network as the final destination. Thus the router system forms the architectural basis of an internet and handles all traffic except for direct delivery from one host to another.

Datagrams pass from router to router until they arrive at one that can directly deliver the datagram to the host machine. Hence delivery consists of zero or more hops across the network through intermediate routers, followed by direct delivery to the host. If there are zero hops then the sending machine can directly deliver to the host machine since it is on the same network.

The router knows if the datagram should be delivered to a directly connected network by finding the network part of the destination IP address and comparing to its directly connected network or networks. Comer (2014, Section 8.4 *Transmission Across A Single Network*) notes that this comparison is 'computationally efficient, which highlights why the Internet address scheme works well':

Because the internet addresses of all machines in a single network include a common network prefix and extracting that prefix requires only a few machine instructions, testing whether a destination can be reached directly is efficient.

3.5.1 Routing tables

A host only knows about directly connected networks; a router can deliver to any address on the global internet. Routers and hosts maintain an **Internet Protocol forwarding table**. Host machines may send datagrams to other machines on the same network (direct delivery, which does not involve a router); or a host may forward a datagram to a router in order to communicate with other machines on the internet. For this reason hosts have a very simple routing table.

Comer (2014, Section 8.2 *Forwarding In An Internet*), notes that a router provides **transit forwarding**, which means that a router will accept datagrams from all the networks to which it is directly connected, and will forward them onwards as necessary. Hosts do not provide transit forwarding, in fact Comer also notes:

[T]he TCP/IP standards draw a sharp distinction between the functions of a host and those of a router. Anyone who tries to mix host and router functions on a single machine by configuring a host to provide transit forwarding discovers that the machine may not perform as expected.

3.5.2 Next-hop forwarding

If a router cannot directly deliver a datagram then it forwards the datagram to a router it is directly connected to. The idea being that the data packet is moved one step closer to its destination. This is known as next-hop forwarding.

Entries in a forwarding table come in pairs, as Comer (2014, Section 8.7 *Next-Hop Forwarding*) describes:

Conceptually, a forwarding table contains a set of pairs (N, R) , where N is the *network prefix* for a network in the internet and R is the IP address of the “next” router along the path to network N . Router R is called the *next hop*, and the idea of using a forwarding table to store a next hop for each destination is called *next-hop forwarding*.

In practice R , the entry in the next hop column of the table, may be a router on a network directly connected to the router, but it may also be an instruction to deliver the datagram directly. In such a case the network address, N , will correspond to a network the router is directly connected to.

3.5.3 Default addresses

Comer (2014, Section 8.6 *Table-Driven IP Forwarding*) notes that:

routing tables only need to contain network prefixes and not full IP addresses. The distinction is critical: the global Internet has over 800,000,000 individual computers, but only 400,000 unique IPv4 prefixes.

As well as pairs of network addresses and routers, the routing table will also contain a default address. The router paired with the default address will be chosen for the next hop if the destination network address does not match any network address N in the pairs (N, R) in the routing table. This allows routing tables to be shorter, as all of the networks that can be reached by the default address do not need to have their own separate entry in the table.

Comer (2014, Section 8.8 *Default Routes And A Host Example*), notes that default addresses work well for ‘typical host computers that obtain service from an ISP’ since:

In such cases, the forwarding table in the host table only needs two entries: one for the local net at the ISP and a default entry that points to the ISP’s router.

3.5.4 Address masks

Successful use of routing tables require that the network address N , has an address mask. Each network address has its own distinct address mask, a number that tells the router how many bits, counting from the most significant bit, of the IP destination address to consider as the network part of the destination address. For example, if the number was 21, then the most significant 21 bits of the destination IP address would be extracted, and checked against the network address N stored in that entry in the table.

Comer gives an algorithm, the **Unified IP Forwarding Algorithm** that is used in routing tables to find a match between an IP destination address, and a network address in the routing table (Comer 2014, Section 8.11 *Longest-Prefix Match Paradigm*). The algorithm’s first step states ‘Insure forwarding table is ordered with longest-prefix first’; **longest-prefix** means the address mask. The longest mask is that with the numerically highest number in decimal; namely, 24 is longer than 23. By arranging routing tables in this way, the routing algorithm needs only check each entry in turn for a match until one is found, or as Comer (2014, Section 8.10 *The IP Forwarding Algorithm*) puts it:

In essence, the algorithm iterates through entries in the forwarding table until it finds a match. The algorithm assumes entries are arranged in longest-prefix order (i.e., the entries with the longest mask occur first). Therefore, as soon as the destination matches an entry, the algorithm can send the datagram to the next specified hop.

The network part of the destination address is computed separately for each entry in the table, since each entry has its own distinct address mask. In the case of the default address, the mask is set to zero, as is the default address. Setting the mask to zero will result in a destination address of zero, hence the default address will always match the destination address. Arranging the routing table with entries that have the longest mask first, means that the default address will only match if nothing else does, since the default address has the shortest possible mask and will therefore be the last entry in the table.

The address mask is given in slash notation following the network address, for example 193.25.11.0/28. An address mask has one bit corresponding to each bit of a 32-bit IP address. Network administrators give the address mask in decimal to reduce the possibility of mistakes that might creep in if they had to give the 32-bit binary number corresponding to the address mask that the router will actually use to find network addresses.

To find the 32-bit binary form of the address mask from the decimal number given in slash notation, starting from the most significant figure, write down as many 1s as the decimal number given. Any remaining places in the 32-bit number are filled with zeros. Hence /28 gives an address mask of 11111111 11111111 11110000. This corresponds to 255.255.255.240 in dotted decimal notation.

We will discuss address masks further in Section 3.5.5, and in Section 3.10 *Classless Inter-Domain Routing (CIDR)*.

3.5.5 Example of calculating a network ID from an address mask

In the Unified IP Forwarding Algorithm given by Comer (2014, Section 8.11 *Longest-Prefix Match Paradigm*), and applied by routers to find a network address for next-hop forwarding, the second step is to extract the destination IP address, D , from the header. Note this is really two steps, as it means firstly removing the datagram from the frame in which it arrived. Once this has been done, then the third step is that for each table entry 'Compute the *logical and* of D with mask'. In order to understand what that means, we will consider an example.

Suppose that the destination IP address is 221.58.136.11, and the current network entry in the routing table is 221.58.128.0/17.

The network entry, in binary octet form is: 11011101 00111010 10000000 00000000

An address mask of 17, in binary octets is: 11111111 11111111 10000000 00000000 (255.255.128.0 in dotted decimal notation).

'ANDing' is a Boolean operation that means that given two inputs, the output is true if both inputs are true, false otherwise. In terms of binary numbers, if 0 is false and 1 true, then 0 AND 0 is 0, 1 AND 1 is 1, 0 AND 1 is 0. This comes down to anything compared to zero is zero, else the result is 1. A bitwise AND over two binary numbers means each bit from the first number is ANDed with the bit in the corresponding place in the other number. Hence performing a bitwise AND with the destination IP address and the address mask gives:

```
11111111 11111111 10000000 00000000 - address mask of 17
11011101 00111010 10001000 00001011 - destination IP address
11011101 00111010 10000000 00000000 - result of AND
```

Comparing the result of the bitwise AND, and the network address N , it can easily be seen that there is a match.

11011101 00111010 10000000 00000000 - result of AND

11011101 00111010 10000000 00000000 - network entry in routing table

221.58.128.0 – result of ANDing the IP destination address with the bitmask

221.58.128.0 – network address from routing table

Activity 3.1

A routing table contains the following two network entries:

162.168.0.0/27

221.61.192.0/19

Given a destination address of 221.61.199.133, is there a matching network entry in the routing table? Justify your answer.

3.5.6 Processing and discarding packets

Once a match is found the routing table next hop entry is used – this may be a router, or it may be an instruction to deliver directly. If the next hop is a router, the current router must decrement the time-to-live before encapsulating the IP datagram into a frame.

One reason for discarding a datagram is if no matching entry is found (perhaps a network administrator forgot to include the default address). In this case the router sends a diagnostic message via ICMP back to the source IP address saying the message could not be delivered. There are many other reasons to discard a datagram. For example, there are the following reasons arising directly from the information in the header:

- **Version:** Intermediate routers will discard datagrams if they do not have the software to interpret the header; for example, if a router expects an IPv4 datagram, but finds from the version number that the datagram is IPv6.
- **Time-to-live:** The time-to-live number is already zero.
- **Header checksum:** If the checksum does not match the value computed by the router then the datagram has been corrupted.
- **Flags, Fragment Offset:** The 'don't fragment' flag has been set. Hence the router cannot fragment the datagram in order to forward to a network with an MTU that is too small to accept the datagram.

Examples of reasons to discard that are not triggered by the header:

- A datagram arrives at the destination network, but the specified host does not exist.
- The router does not have the storage necessary to create the outgoing frame.
- A host attempting to reassemble a fragmented datagram cannot complete the task within the set time limit due to a missing fragment or fragments.

Whenever a datagram is discarded an error or diagnostic message is sent via ICMP to the sender.

To sum up, once a router receives a frame, it extracts the IP datagram, calculates the checksum, and checks the time-to-live. If the result of this is not to discard the packet, then the router makes a routing decision, decrements the time-to-live and recalculates the checksum. At this point it may fragment into two or more frames and forward, encapsulate in a single frame and forward, directly deliver or discard.

3.5.7 Dynamic routing

Routing tables can be static or dynamic. Static tables are defined by the network administrator. Dynamic tables are updated by the routers themselves using information they receive from other routers about network paths. There are many algorithms for dynamic routing, which are beyond the scope of this course.

Activity 3.2

Consider the fields of the IP datagram header.

- i. How many of these fields are changed by routers when forwarding datagrams?
- ii. Identify the fields that are changed and explain why they are altered.

3.6 IPv4 and classful addressing

IP addressing is a hardware-independent convention, which in principle allows every computer attached to the internet to be given a unique logical address. IP addresses are currently 32-bit binary strings which are normally seen by humans (for example, network administrators), in dotted decimal as in the following example:

223.58.1.101

The decimal numbers here have no meaning in isolation; this is simply a convenient way of encoding the numbers so that administrators and others find it easier to recognise them. What happens is that the 32-bit address is broken up into four 8-bit sequences, or octets, each of which is converted to decimal. So the above address in binary is:

11011111 00111010 00000001 01100101

An IPV4 address is divided into a network portion and a host portion. Routers only look at the network portion when deciding whether to forward, discard or directly deliver a data packet. We have looked at how routers know what the network part of an address is, which has changed from the original IPv4 address scheme. In the original IPv4 classful addressing scheme, the network ID always ended on an octet boundary.

3.6.1 The original IPv4 classful address scheme

The original IPv4 address scheme was divided into 5 classes, A, B, C, D and E. Class D is for multicast addresses (sending data to many destinations at once), and class E is reserved for future use. Classes A, B and C are the addresses that are used for hosts and networks.

Class	Most significant bits of the binary address	Lowest network address	Highest network address
A	0	1.0.0.0	126.0.0.0
B	10	128.0.0.0	191.255.0.0
C	110	192.0.0.0	223.255.255.0

Table 3.1: IPv4 address classes A–C: first and last addresses.

A router can find the address class simply by considering the three most significant bits of the network address. All class D addresses start with 1110 hence considering only the first two most significant bits of an address could lead to confusion between class C (110) and D (1110) addresses. An address with the most significant bit set to zero is a class A address, one with most significant bits 10 is class B, and 110 indicates class C. Once a router knows the

class of the address, it knows the network portion, since the network part is measured in whole octets as follows:

	0	8	16	24
Class A	Network ID	Host ID		
Class B	Network ID		Host ID	
Class C	Network ID			Host ID

Table 3.2: IPv4 address classes A–C: network and host bits.

3.6.2 Exclusions

Some address ranges are excluded from the above classes. Addresses that start 127 are used for diagnostics. The following address ranges are designated as ‘private’ meaning that they can only be used on networks that are not connected to the global internet, or networks that are connected but hidden, see the section on **Network Address Translation** later in this chapter:

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

3.6.3 Class A network and host addresses

- **Networks:** The first octet only is used for the class A network part of the address. Since the most significant bit is fixed at zero, only 7 bits give unique network addresses. This means that there are 2^7 (= 128) addresses, numbered from zero to 127 (see Appendix 1 for an explanation of why there are exactly 2^7 potential addresses including zero). However, only 126 class A networks are possible since network addresses 0 and 127 are not used.
- **Hosts:** Each class A network has 24 bits for the host part, giving 2^{24} potential hosts in each class A network. However, the zero and all ones host addresses are not used in any network, giving $2^{24} - 2$, or 16,777,214 possible hosts in each class A network.

3.6.4 Class B network and host addresses

- **Networks:** A class B network uses the first two octets for the network address. Since the 2 most significant bits of the first octet are fixed, this leaves 14 bits for unique network addresses, hence there are 2^{14} , or 16,384 class B networks.
- **Hosts:** With 16 bits for host addresses, and with the all zeros and all ones host addresses not used, this gives $2^{16} - 2$ host addresses, or 65,534 possible host addresses in a class B network.

3.6.5 Class C network and host addresses

- **Networks:** A class C network uses the first three octets for the network address. Since the 3 most significant bits of the first octet are fixed, this leaves 21 bits for unique network addresses, hence there are 2^{21} , or 2,097,152 class C networks.
- **Hosts:** With 8 bits for host addresses, and with the all zeros and all ones host addresses not used, this gives $2^8 - 2$ host addresses, or 254.

Class	Unique bits for network	Bits for hosts	Number of networks	Number of hosts
A	7	24	$2^7 - 2$	$2^{24} - 2$
B	14	16	2^{14}	$2^{16} - 2$
C	21	8	2^{21}	$2^8 - 2$

Table 3.3: IPv4 address classes A–C: total number of networks and hosts.

3.6.6 The all zeros and all ones host addresses

The host address that is all zeros is not used, as it is the network address. For example, given the class C network 192.56.52.0 there is no difference between the network address, and the host address of zero. The all ones address is not used because it is the broadcast address. That is, a router that receives a data packet for direct delivery that has the all ones host address understands that the data packet is to be delivered to every host on the network. The all ones host address for the above network would be 192.56.52.255, since 255 is 11111111 in binary. Similarly, for the class B network 137.236.0.0 the host address of 0.0 and the network address are the same. The broadcast address for this network would be 137.236.255.255, giving all ones in the host part of the address when it is converted to binary.

3.7 Exhaustion of IPv4 addresses: Subnetting, CIDR and IPv6

Recall that ICANN is the global regulatory body that oversees the allocation of IP addresses and coordinates IP addresses with domain names. The management of IP addresses is delegated to five Regional Internet Registries (RIRs); for example, there is an RIR for Africa, another serves Europe, Central Asia, Russia, and West Asia. RIRs in turn allocate blocks of IP addresses to ISPs and other major clients.

The classful addressing scheme's convention that the part of the address which identifies a physical network must come on an 8-bit boundary meant that the address space could not be allocated efficiently and a significant proportion of addresses were wasted. For instance, a class A network can in theory have 16 million host IDs, which is many more than would be possible for a single physical network (subnet). Many enterprises were allocated class B addresses, giving them a theoretical maximum of 65,535 hosts, but had no more than a few hundred hosts: under the classful scheme the remaining host IDs for that network ID were unused.

The problem partly arose because the designers of the IPv4 scheme made the assumption that there would be few networks. As Comer notes (Comer 2014, Section 5.4 *Dotted Decimal Notation Used With IPv4*), in the early 1980s, with the growth of LANs, it was clear that this assumption was mistaken, and that the world would run out of network addresses. Hence the subnetting scheme was devised in order to make more efficient use of IPv4 addresses (see Section 3.8). When the pressure on the address space continued to grow, two further things happened in the 1990s:

- (1) work began on a new standard for IP addresses
- (2) a temporary solution to the pressure on IP addresses, known as *Classless Inter-Domain Routing* (CIDR) was adopted.

We will consider in turn subnetting, the new standard for IP addresses now known as IPv6 and CIDR.

3.8 Subnetting

While there are only 126 class A networks, each network can have more than 16 million hosts. However it is not physically possible for engineering infrastructure to support 16 million hosts on one network. With more than 2 million network addresses, clearly the most common network address is class C. Each class C network has only 254 possible hosts, but even here an organisation might find it more convenient to divide these hosts into networks, for example if the company was in two buildings, or was on different floors of the same high rise office. One solution would be to ask for a second network address, but as noted by Comer (2014, Section 5.5 *IPv4 Subnet Addressing*):

In the early 1980s, as Local Area Networks became widely available, it became apparent that the classful addressing scheme would have insufficient network addresses, especially class B prefixes. The question arose: how can the technology accommodate growth without abandoning the original classful addressing scheme? The first answer was a technique called *subnet addressing* or *subnetting*. Subnetting allows a single network prefix to be used for multiple physical networks.

Subnetting is the logical division of hosts on a network into distinct sub-networks. In 1985 RFC 950 set out the industry wide standards for network administrators to apply to subnetting.

Subnetting in practice is done by network administrators taking some of the host bits from the IP address, and using them to divide the host addresses into separate internal networks. Intermediate routers (that is, routers not directly connected to the network) do not see these subnetworks, since routers do not store host addresses, but only network addresses. Hence, as far as the internet is concerned, there is only one visible public network, however many subnets the network may have.

Subnetting is achieved by using some of the first bits of the host part of the address for the address of subnets. The number of bits borrowed from the host part of the address for subnetting can be chosen by network administrators to suit their organisation's needs, however the same number of bits is normally used for every subnet, meaning that each subnet is limited to the same number of hosts. The standard described in RFC 950 does allow for variable-length subnet addresses, which would allow for an organisation to have variable size subnets, as suited to their particular needs. However, as Comer notes (Comer 2014, Section 5.7 *Variable-Length IPv4 Subnets*):

[V]ariable-length subnetting has serious disadvantages. The most severe disadvantage arises because the scheme can be difficult to administer. The partition for each subnet and the values chosen for subnet numbers must be assigned carefully to avoid *address ambiguity*, a situation in which an address is interpreted differently on two physical networks. [...] Invalid variable-length subnets may make it impossible for all pairs of hosts at the site to communicate. Furthermore, the ambiguity cannot be resolved except by renumbering. Thus network managers are discouraged from using variable-length subnetting.

Note that the bits taken from the host address should be contiguous, and while the original standard set out in RFC 950 did not specify contiguous bits should be used, in practice using non-contiguous bits for the subnet addresses makes network management very difficult and is now discouraged.

RFC 950

Since the bits that identify the subnet are specified by a bitmask, they need not be adjacent in the address. However, we recommend that the subnet bits be contiguous and located as the most significant bits of the local address.

See: <https://tools.ietf.org/html/rfc950>

Comer (2014, Section 5.6 Fixed Length IPv4 Subnets) notes that:

The idea of allowing a site to choose a division for the local portion of its address and then using the division throughout the site is known as fixed-length subnetting. Fixed-length subnetting is easy to understand because it partitions the local portion of an address between networks and hosts. In essence, as a manager chooses how many networks the site can have, the manager also determines the maximum number of hosts on a given network.

We will only consider fixed-length subnetting with contiguous bits.

3.8.1 Subnet masks

The basic idea behind subnetting is that we take away some bits from the host portion of the address and use them to identify a subnet. This enables the address space below the level of the network ID to be segmented into different sub-networks. This division will only be visible to hosts and routers on the local network; from the point of view of the internet at large only the network ID portion will have any particular meaning. The way this works is that the network administrator defines a subnet mask, a string of 32 binary digits indicating the boundary between the subnet and host sections of the 'local' portion of an IP address.

Subnets are specified by a 32-bit subnet mask. We can consider an IP address as consisting of a network part, and a host part. When a network is subnetted, we can consider the IP address to consist of a network part, or internet part, since this part is what the internet sees. This part is followed by what Comer (2014, Section 5.8 *Implementation of IPv4 Subnets With Masks*) calls the 'physical network' part (namely, those bits of the host address used to identify subnets). The final part of the address is the remaining bits of the host part, now used to identify hosts on subnets. A subnet mask will consist of ones indicating the network and physical network part of the address, and zeros for the subnet host part. A subnet mask is only seen and used by the local network.

Seen by the router:	Network address	Host address	
Seen by the local network:	Network address	Subnet address	Host address

Table 3.4: Router and local network view of IP addresses.

For example, the following subnet mask means that there are 5 bits for host addresses on the network:

11111111 11111111 11111111 11100000

How many bits have been borrowed from the host part of the address depends on the class of the address that the mask is applied to. If applied to a class C address, it would mean that 3 bits had been taken from the 8-bits of the host address part for subnet addressing. If applied to a class B address, it would mean that 11 bits had been borrowed from the 16-bits of the host part of the address for the subnet addresses. In both cases, 5 bits would be used to identify hosts in the subnets.

This brings up the important point that it is up to organisations to determine the best subnet mask for their purposes – the standard outlined in RFC 950 allows for network administrators to choose the number of bits to borrow from the host for the subnet address.

3.8.2 Example of finding a network address from a subnet mask

The subnet mask can be used to find the network part of an IP address, including the subnetwork part of the address. For example, suppose the IP address:

192.168.67.133

was that of a host on a network with the subnet mask

11111111 11111111 11111111 11100000. The network part of the address can be found by converting the host address to binary and performing a bitwise AND with the subnet mask. This means that corresponding bits in each binary number are ANDed (remembering that AND means that 1 AND 1 is 1, and everything else is zero) and gives the following:

```
11111111 11111111 11111111 11100000  subnet mask
11000000 10101000 01000011 10000101  host address
11000000 10101000 01000011 10000000  result of bitwise AND
```

The above result means that the network part of the address is 11000000 10101000 01000011 10000000, or 192.168.67.128. Since this is a class C address, the internet part of the network address is 192.168.67.0, while 192.168.67.128 gives the internet and subnet network address combined. Or, in other words, the subnet address for this host is 128_{10} or 10000000_2 .

Since three bits are used for the subnet address, the subnet address is 100_2 (the part in bold in the host address above), or 4_{10} . This can be confusing when network administrators refer to subnet address 4, which is in fact designated with 192.168.67.128 or 11000000 10101000 01000011 10000000. Since there are 5 bits for the host address, the address of the host on the subnet 128 is 00101, or host 3.

3.8.3 Number of subnets and hosts

Subnet bits	Number of subnets	Host bits	Number of hosts
P	2^P	Q	$2^Q - 2$

Table 3.5: How to calculate the number of subnets and hosts.

In general, if there are m bits taken from the host part of the address for the subnet addresses, then there can be 2^m subnets. If there are n host bits remaining, then there are $2^n - 2$ hosts. This is because the all ones and all zeros host addresses are not used, since all ones is the broadcast address, and all zeros gives the subnet address.

It used to be the case that the all zeros and all ones subnet addresses were not used, as RFC 950 recommended against their use because of possible address ambiguity. In the case of the all zeros subnet, the network and the all zeros subnet would have the same address. In the case of the all ones address, the broadcast address for the network, and the all ones subnet address would have the same address. In the past hardware could not cope with this ambiguity, but modern hardware can and does. Casad notes (Casad 2017, **Hour 5: Subnetting and CIDR, Section *The Old Way: Subnet Mask***):

Although the use of the zero subnet and the all ones subnet is officially discouraged, some router manufacturers are unwilling to give up this valuable address space and support them anyway.

3.8.4 Subnet masks for class C addresses

In a class C, there are 9 potential subnet masks as follows:

Subnet mask	Subnet mask in binary	Number of subnets	Number of hosts
255.255.255.0	11111111 11111111 11111111 00000000	$2^0 = 1$	$2^8 - 2 = 254$
255.255.255.128	11111111 11111111 11111111 10000000	$2^1 = 2$	$2^7 - 2 = 126$
255.255.255.192	11111111 11111111 11111111 11000000	$2^2 = 4$	$2^6 - 2 = 62$
255.255.255.224	11111111 11111111 11111111 11100000	$2^3 = 8$	$2^5 - 2 = 30$
255.255.255.240	11111111 11111111 11111111 11110000	$2^4 = 16$	$2^4 - 2 = 14$
255.255.255.248	11111111 11111111 11111111 11111000	$2^5 = 32$	$2^3 - 2 = 6$
255.255.255.252	11111111 11111111 11111111 11111100	$2^6 = 64$	$2^2 - 2 = 2$
255.255.255.254	11111111 11111111 11111111 11111110	$2^7 = 128$	$2^1 - 2 = 0$
255.255.255.255	11111111 11111111 11111111 11111111	$2^8 = 256$	$2^0 - 2 = -1$

Table 3.6: Subnet masks for class C addresses.

The number of subnets for any particular mask is given by the number of bits borrowed from the host part of the address for the subnet, raised to the power of 2. The number of bits devoted to the subnet can be easily calculated from the subnet mask, by counting the number of 1s in that part of the mask that would correspond with the host part of the address. In the table above, since we are dealing with subnet masks for class C addresses, we need to count the 1s in the final octet of the subnet mask. The number of possible hosts in a particular subnet is given by the host bits of the IP address minus the bits borrowed for the subnet, raised to the power of 2, but with 2 subtracted. For example if there are 5 subnet bits, then the number of hosts is $(8-5)^2 - 2$. Two is subtracted from the number of hosts because the all zeros host address is the same as the subnet address, so cannot be used, and the all ones subnet address is the broadcast address for that subnet, so is not used as a host address.

The first entry in the table above is the default subnet mask – it has 0 in the network part of the address. Hence the default subnet mask for a class B address would be 255.255.0.0. The default mask will return the network address seen by the internet, so that the subnet will be the entire host address space for the network. The final two subnet masks are clearly unusable, as they do not give any possible hosts. The subnet mask 255.255.255.252 is unlikely to be used in practice, as it gives 64 subnets, each with only 2 hosts. Ignoring the default address, but including 255.255.255.252, there are 6 possible subnet masks for a class C network.

3.8.5 A subnetting example

A network administrator with an IPv4 address of 193.168.67.0 wishes to apply a subnet mask such that each subnet will have at least 25 hosts.

Questions

1. What is the category of this network address? How many bits of the address are reserved for the network, and how many for the host?
2. How many network addresses are there in this category? How many host addresses per network? How many host addresses in total? Explain your answers.
3. What subnet mask should the administrator use in order to have the most possible subnets?
4. How many total usable hosts are there with your chosen subnet mask? How many hosts without subnetting?
5. Give the address of each subnet, the corresponding range of usable host addresses, and the broadcast address for each subnet.

Answers

1. Since the address starts with 193 (in binary 11000001) it is a class C address, with 24 network bits and 8 host bits.
2. Since all class C addresses must start with 110, only 21 network bits can actually give unique addresses, so there are 2^{21} class C network addresses. With 8 bits for the host address, each network can have $2^8 - 2$ hosts. Two host addresses cannot be used, the all zeros, as it is the same as the network address, and the all ones (255), as this address is the broadcast address for the network. This means that there are $2^{21} \times (2^8 - 2)$ total potential hosts. This works out to 532,676,608 or more than half a billion.
3. In Table 3.6, we can see that the subnet mask 255.255.255.240 will only give 14 hosts per subnet, and the mask 255.255.255.192 will give 62 hosts per subnet. But in order to have the most possible subnets meeting the condition of at least 25 hosts the administrator should use the subnet mask 255.255.255.224, as this will give 30 hosts per subnet.
4. Without subnetting there are $2^8 - 2$ hosts, or 254. With subnetting, there are 8 subnets times 30 hosts, or 240 hosts. There are 30 rather than 32 hosts as the all zero host address is the same as the subnet address and so is not used, and the all ones host address is the broadcast address for the subnet.
5. The address of each subnet, the corresponding range of usable host addresses, and the broadcast address for each subnet is in the table below.

Subnet number	Subnet address	Host range	Subnet broadcast decimal/binary	
0	193.168.67.0	1–30	31	00011111
1	193.168.67.32	33–62	63	00111111
2	193.168.67.64	65–94	95	01011111
3	193.168.67. 96	97–126	127	01111111
4	193.168.67.128	129–158	159	10011111
5	193.168.67.160	161–190	191	10111111
6	193.168.67. 192	193–222	223	11011111
7	193.168.67.224	225–254	255	11111111

Table 3.7: The address, host range and broadcast address of every subnet for the address 193.168.67.0 with subnet mask 255.255.255.224. (In the above table assume that each host address and broadcast address is prefixed with 193.168.67.)

In Table 3.7 the host part of the binary number for the broadcast address is in bold, so that you can see it is all ones. If you were to convert the last octet of the subnet address to binary, you would find that in each case the 5 bits of the host address are zero, for example 96 is 011**00000** and 192 is 11**000000**.

3.8.6 Finding subnet addresses

How would the network admin work out the subnet addresses? In the above example, the first address is easy, it is 193.168.67.0; namely the same as the network address, and discouraged from use by RFC 950, to avoid confusion with the network address. Similarly, the use of the all ones subnet address (in this case 224, or 11100000) was also discouraged so that it would not be possible to have a network and a subnet with the same broadcast address. From the table above you can see that the 224 subnet address has the broadcast address 255 (or 1111 1111), which is the same as the broadcast address for the entire network. In recent years manufacturers of hardware, such as Cisco, have started to produce routers that can support the use of the all zeros and all ones subnet addresses, as they can manage the ambiguity in addresses successfully. Hence we will assume that all subnet addresses can be used.

Since we are using the subnet mask 11111111 11111111 11111111 11100000, there are 3 bits for subnet addresses. Therefore we can find all subnet addresses by writing down all the binary numbers that have 3 bits, and then translating them into the most significant bits of an 8-bit binary field, and working out the corresponding decimal number, as follows:

3-bit binary number	8-bit binary field	8-bit decimal equivalent
000	0000 0000	0
001	0010 0000	32
010	0100 0000	64
011	0110 0000	96
100	1000 0000	128
101	1010 0000	160
110	1100 0000	192
111	1110 0000	224

Table 3.8: Finding all 3-bit subnet addresses.

Activity 3.3

A company with a network address of 156.157.0.0 wants to segment the network into at least 18 different subnets.

1. What subnet mask is needed? Justify your answer.
2. Give the number of bits used to represent a host and the number of usable hosts per subnet.
3. Give the addresses of the first, second, next-to-last and last subnets.
4. Write down the range of host addresses in the second subnet.

3.9 IPv6

The new standard for IP addresses, IPv6, was described in RFC 2460, issued in 1998. The abstract states:

This document specifies version 6 of the Internet Protocol (IPv6), also sometimes referred to as IP Next Generation or IPng.

See: www.ietf.org/rfc/rfc2460.txt

The original IPv4 classful addressing scheme allowed for more than 2 million networks and provides for more than 3 billion hosts. However, as Casad (2017), notes:

Is three billion addresses truly enough for a world with over seven billion people, many of whom now connect to the Internet with multiple devices?

As well as the popularity of LANs, there are other issues raising demand for IP addresses, which were mostly impossible for the original designers of the IPv4 to anticipate:

- Take-up by business and domestic users.
- Always on internet connections such as broadband connections mean that each user needs a dedicated IP address. When machines connected and disconnected via modems, ISPs could share out a limited number of IP addresses by reallocating them as customers disconnected.
- Convergence of computing, communications and entertainment industries such that televisions, smart phones, games consoles and DVD players may require an IP address.
- The Internet of Things.

The problem of running out of network addresses has been anticipated and discussed for many years, hence since 1998 there has been agreement on a new 128-bit address scheme, known as IPv6. Comer (2014, Section 5.10 *The Current Classless IPv4 Addressing Scheme*) notes:

By 1993, it became apparent that subnetting alone would not prevent Internet growth from quickly exhausting the address space, and preliminary work began on defining an entirely new version of IP with larger addresses.

IPv6 was specified by RFC 2460 December 1998 see: www.ietf.org/rfc/rfc2460.txt. IPv6 has a 128-bit address format, and was designed to allow for one billion **networks**.

In February 2011 ICANN announced that it had officially run out of IPv4 addresses. In addition as Casad (2017, **Hour 13**: IPv6: The Next Generation, section titled *Why a New IP?*) notes, 'at this writing, four of the five regional authorities that report under IANA are also out of addresses'. That raises two inter-related questions: (1) why is it that IPv4 addresses are still in use, and that new devices are able to connect to the internet? And (2) why has the world not already switched over to the IPv6 addressing scheme?

Casad expresses some scepticism about the transition to IPv6:

The most common version of IP is IPv4, although the world is theoretically in transition to a new version of IP known as IPv6.

Casad (2017, **Hour 4**: The Internet Layer, section titled *Addressing and Delivering*)

In the introduction to the hour on IPv4, Casad notes:

The Internet Protocol (IP), which defines the all-important IP address system has been poised for an upgrade for years.

Casad (2017, Introduction to **Hour 13**: IPv6: The Next Generation)

Casad goes on to say:

1998 is ancient history by internet standards. Why hasn't the whole world already embraced this powerful and vast new IPv6 address space and abandoned the sinking ship of IPv4 addressing?

Casad (2017, **Hour 13**: IPv6: The Next Generation, section titled *Why a New IP?*)

The answer to this question, claims Casad, is the popularity of **Network Address Translation (NAT)**.

3.9.1 Network Address Translation (NAT)

With NAT a single router can hide a large network behind it. In this scheme only the router has a public IP address; the network behind the router has a private addressing scheme. When a networked computer belonging to the hidden network attempts to connect to the internet, a NAT device makes and manages the connection such that the transmission appears to be coming from the IP of the router. NAT devices are very popular for a number of reasons, including improved security. With NAT a potential attacker does not know that there is a network to attack, since any router acting as a NAT device appears to be a single host. NAT devices have been developed and used for some time; in fact RFC 2663 attempts to define common terminology, and was written in 1999. See: <https://tools.ietf.org/html/rfc2663>

Within the hidden network, hosts and routers, while they could theoretically use any IP address, in practice use addresses designated as private under the IPv4 scheme. This means that any data packet that is accidentally forwarded onto the internet by a machine on the private network will be discarded, since routers will recognise the private source address as invalid.

Casad 2017 (**Hour 4**: The Internet Layer, section titled *Special IP Addresses*) notes that:

RFC 1597 (which was later updated with RFC 1918) reserves some IP address ranges for private networks. The assumption is that these private address ranges are not connected to the Internet, so the addresses don't have to be unique. In today's world, these private address ranges are often used for the protected network behind Network Address Translation (NAT) devices [...]

Because the private address ranges don't have to be synchronized with the rest of the world, the complete address range is available for any network. A network administrator using these private addresses has more room for subnetting, and many more assignable addresses.

3.10 Classless Inter-Domain Routing (CIDR)

Note that in Section 3.5.4 we considered how routers use a bit mask to determine the network part of the destination IP address to consider. We now consider the development of this bit mask, and its application in **Classless Inter-Domain Routing (CIDR)**.

While Casad claims that NAT is the reason why IPv4 addresses are still in use today, he acknowledges that CIDR relieved pressure on the IPv4 address space: 'CIDR has prolonged the life of the IPv4 Internet by greatly simplifying Internet routing tables' (Casad, 2017), **Hour 5**: Subnetting and CIDR, Section titled *The New Way: CIDR*). Comer believes that CIDR is **the** major reason that IPv4 addresses are still in use.

Comer (2014, Section 5.10 *The Current Classless IPv4 Addressing Scheme*) writes that CIDR was the **temporary** solution adopted in the 1990s, to relieve pressure on the IPv4 address space until IPv6 could be agreed and introduced:

Known as *classless addressing*, the temporary address scheme does away with class A, B and C addresses. In place of the three classes, the new scheme extends the idea used in subnet addressing to permit a network prefix to be an arbitrary length. ... [I]n addition to a new addressing model, the designers modified forwarding and route propagation techniques to handle classless addresses. As a result, the entire technology has become known as *Classless Inter-Domain Routing (CIDR)*.

Two important things to note about the classful schemes, is that while there are over 2 million class C networks, demand for them is slight, since the number of hosts is too restrictive for many organisations. However demand for class B networks has always been strong, but there are only 65,534 of them. In the 1990s it was expected that class B networks would be exhausted first, and it was the demand for class B networks that CIDR originally addressed.

3.10.1 Supernetting

Comer (2014, Section 5.10 *The Current Classless IPv4 Addressing Scheme*) notes that the original intention of classless addressing was to group class C addresses, a process known as **supernetting**. The idea was that instead of giving one class B address, many class C addresses could be grouped into a contiguous block, and assigned instead. Since all networks in a contiguous group could be identified in routing tables by the starting IP network address (known as hierarchical routing aggregation), this simplified routing tables, as well as reducing the demand for class B addresses. Comer writes: 'Although the first intended use of CIDR involved blocks of class C addresses, the designers realized that CIDR could be applied in a much broader context'

3.10.2 CIDR and routing

With CIDR as it is now implemented, the classful system is abolished, and the network boundary can appear anywhere in an address. Routing tables use a number called a 'CIDR prefix' to determine how many bits of an address to consider as part of the network ID. As we saw in Section 3.5.4, routing tables store a network address and its CIDR prefix. When deciding how to forward a data packet, the router uses the CIDR prefix to determine an address mask. The address mask is applied to the destination IP address, to get a destination network address to compare to the network address paired with the CIDR prefix. We considered an example of how to extract the network ID using an address mask and the destination IP address in Section 3.5.5.

3.10.3 CIDR and the IPv4 address space

As well as making routing faster, CIDR also means the better use of addresses, because organisations need no longer be allocated many more addresses than they need, but can be allocated just enough, rounded to the nearest power of 2. They are rounded to the nearest power of 2 because, as Comer notes, assigning a block of addresses using CIDR notation, is akin to giving each customer of an ISP 'a (variable-length) subnet of the ISP's CIDR block'. (Comer 2014, Section 5.12 *A Classless IPv4 Addressing Example*).

For example in 3.8.5 above, we considered an example of subnetting an IPv4 address of 193.168.67.0 into 8 subnets using the subnet mask 255.255.255.224, and came up with the following table:

Subnet number	Subnet address	Host range	Subnet broadcast decimal/binary	
0	193.168.67.0	1–30	31	00011111
1	193.168.67.32	33–62	63	00111111
2	193.168.67.64	65–94	95	01011111
3	193.168.67.96	97–126	127	01111111
4	193.168.67.128	129–158	159	10011111
5	193.168.67.160	161–190	191	10111111
6	193.168.67.192	193–222	223	11011111
7	193.168.67.224	225–254	255	11111111

Table 3.9: Copy of Table 3.7. (The address, host range and broadcast address of every subnet for the address 193.168.67.0 with subnet mask 255.255.255.224.)

Under the CIDR scheme, each subnet address would correspond to a network address, and the subnet mask would correspond to the CIDR prefix. Hence the subnet mask 255.255.255.224, which in binary octets is 11111111 11111111 11111111 11100000, corresponds in CIDR notation to /27. So any provider with a CIDR block including 193.168.67.0 could give clients the following network addresses, with each network having 5 bits for the host addresses:

- 193.168.67.0/27
- 193.168.67.32/27
- 193.168.67.64/27
- 193.168.67.96/27
- 193.168.67.128/27
- 193.168.67.160/27
- 193.168.67.192/27
- 193.168.67.224/27

As you can tell from the above, CIDR is very similar to subnetting, the difference being that the CIDR prefix is seen by intermediate routers, unlike subnet masks that are hidden from the wider internet. In addition, CIDR prefixes are set by ISPs, while network administrators determined the subnet mask they wished to use.

Comer notes that the flexibility of CIDR addressing comes from allowing the subdivision into network and host bits to be flexible. This allows ISPs to provide an appropriate number of host addresses for each client, making better use of the address space.

3.11 Overview of the chapter

In this chapter we considered the IP protocol, IP addressing and routing. We finished by looking at the IPv4 address space, and the reasons why IPv4 addresses are still in use even though they have officially run out, including Network Address Translation and Classless Inter-Domain Routing.

3.12 Reminder of learning outcomes

Having completed this chapter, and the Essential reading and activities, you should be able to:

- explain in general terms how information in the IP header is used in routing
- identify different classes of IP addresses
- break down an IP address into network, subnet and host IDs using a subnet mask
- list the factors contributing to a shortage of IP addresses and some techniques which have been proposed to alleviate it.

3.13 Test your knowledge and understanding

3.13.1 Sample examination questions

- (a) (i) What is the data packet created in the transport layer by the TCP protocol called? [2 marks]
1. A datagram
 2. A segment
 3. A frame
 4. None of the above
- (ii) A router is best described as: [2 marks]
1. A machine that can communicate with other machines connected to the internet
 2. A machine that waits for connection requests
 3. A machine connected to the internet that forwards data packets
 4. None of the above
- (iii) ICMP stands for: [2 marks]
1. The Internet Control and Messaging Protocol
 2. The Internet Creative Messaging Protocol
 3. The Internet Credible Maintenance Program
 4. None of the above
- (b) (i) 193.25.11.0/29 is a network entry in a routing table. Give the address mask of the entry in 32-bit binary form and in dotted decimal notation. [6 marks]
- (ii) What address would you find as the final entry in a routing table? [3 marks]
- (c) (i) What does connectionless delivery mean in the context of the TCP/IP protocol suite? [5 marks]
- (ii) Give a field of the IP datagram header that may cause a router to discard an IP datagram. Explain how the field may cause a datagram to be discarded. [5 marks]

Chapter 4: TCP/IP transmission

4.1 Introduction

This chapter looks in detail at the major data structures and processes of one of the most important components of the TCP/IP protocol suite. After reading Chapter 2 you should have a high level understanding of how the success of the internet relies on communication protocols. This chapter goes into more detail about the complexities of the Transmission Control Protocol (TCP), and the necessity for them.

4.1.1 Aims of the chapter

This chapter aims to help you to develop a good understanding of the communication protocols that underlie the success of the internet, in particular the Transmission Control Protocol (TCP).

4.1.2 Learning outcomes

By the end of this chapter, and having completed the Essential reading and activities, you should be able to:

- explain in general terms how information in the TCP header is used in:
 - opening and closing connections
 - flow control
 - reliable data transmission.

4.1.3 Essential reading

- Comer, D.E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) 6th edition Pearson new international edition [ISBN 9781292040813]. Chapter 11, *Reliable Stream Transport Service (TCP)*, Sections:
 - 11.1 *Introduction*
 - 11.2 *The Need For Reliable Service*
 - 11.4 (*Reliability: Acknowledgement And Retransmission*) to 11.11 (*TCP Segment Format*) inclusive
 - 11.15 *Acknowledgement, Retransmission and Timeouts*
 - 11.24 (*Establishing a TCP Connection*) to 11.27 (*TCP Connection Reset*) inclusive
 - 11.34 *Summary*

4.1.4 Further reading

- Comer, D.E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) 6th edition Pearson new international edition [ISBN 9781292040813].
 - Chapter 10 '*User Datagram Protocol (UDP)*'; Sections:
 - 10.1 (*Introduction*)–10.5 (*Interpretation Of the UDP Checksum*) inclusive
 - 10.9 *UDP Encapsulation And Protocol Layering*
 - Chapter 11, '*Reliable Stream Transport Service (TCP)*'; Sections:
 - 11.3 *Properties Of The Reliable Delivery Service*

- 11.16 (*Accurate Measurement Of Round Trip Samples*) to 11.21.1 (*Selective Acknowledgement (SACK)*)
- 11.30 *Reserved TCP Port Numbers*
- Casad, J. *Sams Teach Yourself TCP/IP in 24 Hours*. (Indiana: Pearson Education, 2017) 6th edition [ISBN 9780672337895]:
 - **Hour 6:** The Transport Layer
 - **Hour 7:** The Application Layer

4.1.5 References cited

- <https://tools.ietf.org/html/rfc6335>
- www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
- <https://stackoverflow.com/questions/5330277/what-are-examples-of-tcp-and-udp-in-real-life>
- <https://tools.ietf.org/html/rfc2018>
- <https://tools.ietf.org/html/rfc8547>

4.2 Glossary of key terms

- **UDP:** the User Datagram Protocol. Uses the underlying IP data stream to provide an unreliable, best effort, connectionless data delivery system. Unlike IP, UDP uses port numbers to deliver data packets to a particular service or application on the receiving host.
 - **Unreliable:** there is no guarantee that any one particular packet will be delivered.
 - **Best effort:** the system will discard packets that cannot be delivered. Packets are not discarded arbitrarily, but by following certain rules. Hence all that is promised is a credible attempt at delivering every packet, with no promise that delivery will actually be made.
 - **Connectionless:** Each packet in a data stream is treated as an independent unit, with no connection to previous or subsequent packets in the same stream. Hence packets in the same stream may follow different paths to the same destination; they may be delivered out of order; some may not be delivered at all.
- **TCP:** the Transport Control Protocol. A connection-oriented protocol providing reliable data transmission, and including congestion, error and flow control. Reliable transmission is based on positive acknowledgement with retransmission: it is up to the sending system to take remedial action if there is no confirmation that a packet has been received. Unlike some other communication protocols, there is no negative acknowledgement; that is, the receiving system does not ask for data to be sent again if it is missing or corrupted.
- **Reliable transmission:** delivery is guaranteed, under the right circumstances; if there is an error that cannot be worked around (for example, a server falls over) the transmission failure will be reported.
- **Congestion:** congestion means crowding, which in terms of the internet means it is crowded with data packets, all wanting to be delivered.
- **Congestion collapse:** Routers have finite capacity and may have to queue data packets at busy times, leading to delays. In the worst case, routers may run out of space to queue data packets and may have to discard some. Congestion collapse happens when the network is so crowded with data packets that delivery slows, and packet loss increases, so much so that the network can no longer function effectively.

- **Error control:** the ability to find errors (for example, lost and corrupted data packets), identify them, report on them and correct them.
- **Flow control:** defined by Comer (2014, Section 10.3 *The User Datagram Protocol*) to be control of 'the rate at which information flows between a pair of communicating hosts'.
- **Acknowledgement services:** it is not enough to deliver data packets; there must be some way of knowing that data packets have been delivered. Acknowledgement services are implemented to allow hosts to acknowledge receipt of data packets allowing the integrity of the data stream to be maintained.
- **ACK:** an acknowledgement of receipt of a TCP segment or segments.
- **Positive acknowledgement:** the receiver transmits to the sender details of what it has received.
- **Negative acknowledgement:** the receiver returns details to the sender of data packets that were not received, or were received incorrectly.
- **Selective acknowledgement:** the receiver returns details of data packets received, plus details of any gaps in the data stream from packets not received.
- **Cumulative acknowledgement:** TCP's positive acknowledgement scheme where only in order data packets will be acknowledged, and an acknowledgement with sequence number $x + 1$, implicitly acknowledges that all data with sequence numbers up to and including x , has been received.
- **Sequence numbers:** before transmitting a data stream, the TCP protocol assigns consecutive numbers to all bytes in the data stream. These are known as sequence numbers and allow the receiving host to track and acknowledge received data, and to reconstruct the data stream. The sequence number of the first byte in the data stream is chosen randomly each time a connection is made.
- **Three-way handshake:** TCP guarantees a reliable delivery service. An important part of this is establishing a reliable connection between the sending and receiving host. The three-way handshake means sending three messages between sender and receiver in order to establish a connection and synchronise sequence numbers.
- **Hardware ports:** physical ports where connections are made; for example, a USB port on a computer, a headphone port on a tablet.
- **Logical ports:** 16-bit numbers used to indicate what Application layer service the client wants to access at the end point of a logical connection. Port numbers are assigned by IANA and provide a bridge between Transport level protocols such as TCP, and the Application layer service they are collecting data from at one end, and delivering data to at the other end.
- **Well-known ports:** IANA keeps a record of 'well-known' port numbers, meaning port numbers that are assigned to certain 'privileged' services.

4.3 Transport layer protocols and port numbers

Recall that the Transport layer provides error control, flow control and acknowledgement services. At this level are communication protocols, with the most important being the UDP and TCP communication protocols. UDP provides unreliable data transmission while TCP provides reliable data transmission incorporating error control, flow control and acknowledgement

services. UDP's unreliable data transmission is fast, while TCP's reliable service is slower, hence it could be said that UDP is built for speed, and TCP for reliability.

There are other Transport level protocols, such as the Stream Control Transmission Protocol (SCTP), but this course will concentrate on TCP and UDP, hence you are not required to know about other protocols at the Transport layer.

4.3.1 Well-known port numbers

As discussed in Chapter 2 of the subject guide, there is a registry of well-known port numbers, maintained by the Internet Assigned Numbers Authority (IANA), a part of ICANN. As RFC 6335 describes, the port numbers are divided up as follows:

- the System Ports, also known as the Well Known Ports, from 0-1023 (assigned by IANA)
- the User Ports, also known as the Registered Ports, from 1024-49151 (assigned by IANA)
- the Dynamic Ports, also known as the Private or Ephemeral Ports, from 49152-65535 (never assigned)

See: <https://tools.ietf.org/html/rfc6335>

Ports 0–1023, the 'well-known' ports, have been assigned by IANA to the most frequently used applications. Organisations can apply to IANA to have particular protocols assigned to ports in the User Ports range; while the Dynamic Ports are available for use as and when different processes require them, and cannot be assigned to particular services or protocols. Not all assigned ports will be in use on every machine, and in fact, the IANA web page with the list of port numbers carries the following disclaimer on every page (there are 143 of them):

```
*****
* PLEASE NOTE THE FOLLOWING: *
*
* ASSIGNMENT OF A PORT NUMBER DOES NOT IN ANY WAY IMPLY AN *
* ENDORSEMENT OF AN APPLICATION OR PRODUCT, AND THE FACT THAT NETWORK *
* TRAFFIC IS FLOWING TO OR FROM A REGISTERED PORT DOES NOT MEAN THAT *
* IT IS "GOOD" TRAFFIC, NOR THAT IT NECESSARILY CORRESPONDS TO THE *
* ASSIGNED SERVICE. FIREWALL AND SYSTEM ADMINISTRATORS SHOULD *
* CHOOSE HOW TO CONFIGURE THEIR SYSTEMS BASED ON THEIR KNOWLEDGE OF *
* THE TRAFFIC IN QUESTION, NOT WHETHER THERE IS A PORT NUMBER *
* REGISTERED OR NOT. *
*****
```

See: www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Nevertheless, all machines connected to the internet will interpret the well-known numbers in the same way. Some well-known port numbers (shown in decimal) are:

Port	Protocol	Use
20	FTP	File transfer (data)
21	FTP	File transfer (control signals)
22	SSH	Secure communication
25	SMTP	Sending email
53	DNS	Domain names to IP address translation

Port	Protocol	Use
80	HTTP	World wide web
110	POP-3	Retrieving email
443	HTTPS	Encrypted HTTP
631	IPP	Printer communication

Table 4.1: Well-known port numbers.

4.3.2 Port numbers as links between the Transport and Application layers

The IP protocol only needs an IP address to forward data packets, but in the Transport layer port numbers are also needed. Transport layer protocols interact with services in the application layer, and this is done via ports. These ports are not literal ports, such as USB ports, but logical constructs that identify the application layer service on the receiving machine to connect to.

Recall that in Chapter 2 we discussed how ports, a 16-bit integer, are used to identify and access a particular service on a machine. Comer (2014, Section 10.2 *Using A Protocol Port As An Ultimate Destination*) calls these ports **protocol ports**. They are an example of abstraction – when implementations of the UDP or TCP protocol access a service via a port the implementation does not need to know anything about the process that the data transmitted will be passed to. The protocol implementations do not need to understand details of how data is buffered, queued and passed to the receiving process, and more broadly they need to know nothing about the operating system the receiving machine is running. Transport layer protocols simply need the port number, and the IP address of the receiving machine.

A connection operates (at a certain level of abstraction) between a sending IP address and port number, and a receiving IP address and port number. Each paired IP address and port number (called a **socket** by Casad, 2017) uniquely identifies a service on a receiving computer, hence machines can operate multiple connections at one time without confusion. For example a mail server can operate more than one TCP connection with the same port number, as ports and IP addresses together uniquely identify a connection, or, in other words, each socket is unique. Hence each socket represents a unique connection allowing the server to distinguish between incoming segments from different data streams.

4.3.3 TCP and UDP ports

There are 65,536 port numbers, numbered from 0 to 65,535. That is, ports are numbered from 0 to $2^{16}-1$, hence 16 bits are needed to hold the port number in the TCP or UDP header. In a sense, port numbers are logical constructs that provide a bridge between the transport layer and its protocols, and the application layer.

Consulting the list provided by IANA, at: www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml you can see that port numbers are assigned to protocols as well as to applications. Every port number is assigned to both a TCP and a UDP port, so that there are 65,535 UDP ports, and 65,535 TCP ports. Not every port number is assigned to the other transport protocols such as SCTP. For example, port numbers 20 and 21 (File Transfer Protocol ports) are listed for TCP, UDP and for SCTP, while port number 25 (Simple Mail Transfer Protocol) is listed for TCP and UDP only.

The point to remember is that for every TCP port there is a UDP port, and vice versa. This allows applications to connect with the TCP protocol and the UDP protocol, and to know which protocol they are communicating with. You will come across literature referring to TCP port X or UDP port Y. For example, port 25 may be referred to as TCP port 25 or as UDP port 25. Certain applications are more likely to be associated with a TCP port rather than a UDP port and vice versa. The HTTP application is assigned to port 80, and would normally use TCP port 80 since the application needs a reliable connection. However, developers could choose to use UDP for speed of transmission, and handle any necessary reliability issues within the application.

Comer (2014, Section 11.30 *Reserved TCP Port Numbers*) notes that:

We should point out that TCP and UDP port numbers are independent, the designers have chosen to use the same integer port numbers for any service that is accessible from both UDP and TCP.

Because they provide access, port numbers are also a vulnerability, and network administrators will seek to close ports that are not used. This is particularly important for well-known ports, since they are more likely to be targeted by hackers. For example, a file server might have all ports apart from 20 and 21 closed, since only FTP services are needed. Access to ports can also be restricted to improve computer security, for example a firewall could be configured to allow traffic to port 25, used by the Simple Mail Transfer Protocol (SMTP) so that an organisation can send email, but traffic to other ports may be blocked. Network security issues will be addressed further in Chapter 7 of the subject guide.

4.4 UDP

The User Datagram Protocol (UDP) is a Transport level protocol that does not guarantee that data packets sent will arrive in order, or that they will arrive at all, and makes no attempt at flow control, meaning that data packets could arrive faster than a host can process them. It provides in fact, the same functions as the IP protocol in the Internet layer, and it relies on the IP protocol to do this work for it. What UDP adds to the IP protocol is the ability to distinguish between services on a particular machine. Therefore the header added by the UDP protocol to the message includes source and destination ports. Comer (2014, Section 10.9 *UDP Encapsulation and Protocol Layering*) sums this up with:

The IP layer is responsible only for transferring data between a pair of hosts on an internet, while the UDP layer is responsible only for differentiating among multiple sources or destinations within one host.

Thus only the IP header identifies the source and destination hosts; only the UDP layer identifies the source or destination ports within a host.

Thus UDP is simple and fast, but unreliable. It is suitable for such things as broadcasting video, where a few dropped frames would not be a problem. Comer (2014, Section 11.2 *The Need For Reliable Service*) notes that the developers of applications that need reliable transmission must either implement it themselves, or use the TCP protocol to transmit data, and implies that most developers would prefer to use TCP since 'it is difficult to design, understand and implement software that correctly provides reliability'. In addition the TCP protocol exists and has already done the work of implementing reliability, simplifying the developer's task. However, one user of StackOverflow claims that UDP is used in safety-critical applications, (see:

<https://stackoverflow.com/questions/5330277/what-are-examples-of-tcp-and-udp-in-real-life>). Be that as it may, applications using UDP either accept the unreliable nature of the protocol, or implement their own proprietary algorithms to provide reliability.

4.4.1 UDP header

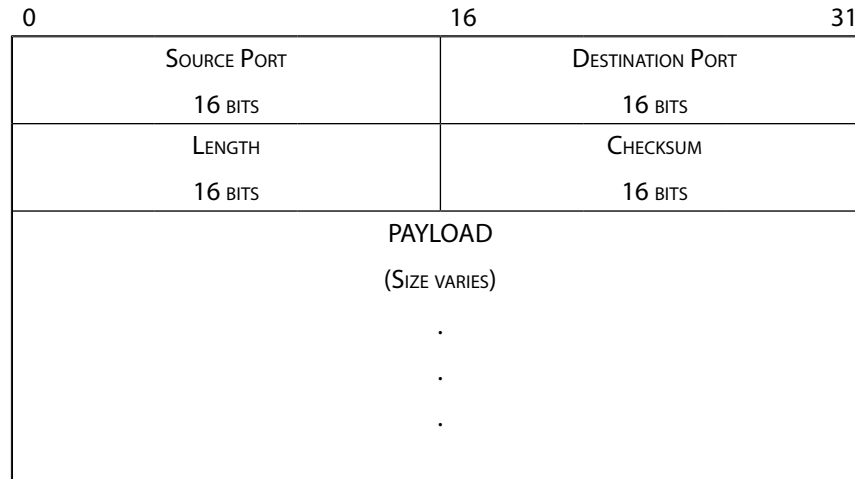


Figure 4.1: UDP header.

The fields of the UDP header are practically self-explanatory, in keeping with the simplicity of the protocol. The header consists of two 32-bit words, each word divided into two sections. The first word contains the numbers of the UDP source and destination ports. The **source port** would be included so that the receiving application could reply. Since the protocol recognises that no reply may be needed or wanted, for example if the protocol is used for video or audio streaming, the source port can be zero in both IPv4 and IPv6.

The **destination port** is the UDP protocol port number for the receiving application, hence making a bridge between the transport layer UDP protocol and the application layer service receiving the datagram.

The **length** is the size of the header and its payload in bytes e.g. if the payload was zero then there would be 8 bytes of header in total, so 8 is the smallest possible number. The largest is 65,535, or $2^{16}-1$, although in practice the UDP datagram must fit into an IP datagram, which, combining header and data, is itself limited to $2^{16}-1$ bytes. Hence the size of the header in the IP datagram limits the size of the UDP datagram somewhat. In practice the size of an IP datagram is limited by the network MTU, which is usually smaller; for example, Ethernet limits the MTU to 1500 bytes. Hence large UDP datagrams will be fragmented in the Internet layer by the Internet Protocol, depending on the MTU of the network.

The **checksum** is optional in IPv4 (required in IPv6), allowing for low transmission overheads in reliable networks. The checksum, when used, is calculated over the UDP header and payload. The UDP datagram checksum is the only check on the integrity of the data in the UDP datagram. The checksum in the IP datagram header is used for the IP header only; the payload, which could be a UDP datagram or a TCP segment, is ignored. For this reason Comer (2014, Section 10.5 *Interpretation of the UDP Checksum*) writes 'the UDP checksum provides the only way to guarantee that data has arrived intact and should be used'.

4.5 TCP

The most important protocol at this level is TCP, the Transport Control Protocol, which is a connection-oriented protocol providing reliable data transmission and flow control. Note that reliable transmission in TCP is based on positive acknowledgement with retransmission: it is up to the sending system to take remedial action if there is no confirmation that a packet has been received. Unlike some other communication protocols, there is no negative acknowledgement; that is the receiving system does not ask for data to be sent again if it is not received, or received with an error.

4.5.1 TCP header

0	8	16	24	31
SOURCE PORT 16 BITS		DESTINATION PORT 16 BITS		
SEQUENCE NUMBER 32 BITS				
ACKNOWLEDGEMENT NUMBER 32 BITS				
LENGTH 4 BITS	RESERVED 6 BITS	CONTROL FLAGS 6 BITS	WINDOW 16 BITS	
CHECKSUM 16 BITS			URGENT POINTER 16 BITS	
OPTIONS 320 BITS (MAX)			PADDING (IF NEEDED)	
PAYLOAD (SIZE VARIES) . . .				

Figure 4.2: TCP header.

The TCP header is much more complex than the UDP header, reflecting the protocol's greater responsibilities. Like the UDP header it is divided into 32-bit words, which may be sub-divided. The following sections describe how various fields in the header are used for interfacing with applications, opening and closing connections, making sure data is communicated without loss or corruption and adjusting the rate at which data is transferred to avoid either overloading the receiving system or under-utilising the network connection.

4.5.2 Fields of the TCP header

The **source port** gives the TCP port that identifies the application that the payload originated from. The **destination port** is the TCP port identifying the application that the payload is being sent to.

Bytes in a data stream are sequentially numbered for transmission. If the SYN control flag is on, then the **sequence number** is being used as part of the three-way handshake (see Section 4.5.3) to synchronise sequence numbers when the connection is established. During a transmission the **sequence number** is normally the number of the first byte in this particular segment.

The **acknowledgement number** is the number of the next byte that the receiver expects.

Length means the size of the header in 32-bit words. Hence this number can be used to determine where the header ends and the payload data begins. The payload does not start at a standard point, because the length of the header is not standardised, but depends on the size of the options field plus padding. That is the options plus padding (if needed) will comprise up to ten 32-bit words.

Reserved is a 6-bit field that is currently unused, but may be needed in some future implementation.

The **control flags** field (called CODE BITS by Comer 2014, see Section 11.11 *TCP Segment Format*) indicates the purpose of the segment. There are six one bit flags; hence each flag can be on (denoted by 1); or off (denoted by 0). The flags are:

- URG – the segment is urgent and the **urgent pointer** field needs to be read
- ACK – if the flag is on then the ACK field is significant, for example, the connection is being established via the three-way handshake, or is being closed
- PSB – push the data
- RST – reset the connection
- SYN – if on then the sequence numbers are being synchronised in the three-way handshake
- FIN – used in closing the connection, if the flag is on then the sender has no more data.

The important flags for this course are the **SYN** and **ACK** flags.

The **window** field is used for flow control. It specifies the amount of data that the host is able to receive.

The **checksum** is computed by the receiver to determine the integrity of the entire segment – the header and the payload data.

The **urgent pointer** field is used to point at the sequence number that urgent information starts from. The segment may not contain urgent information, hence the URG flag is used to indicate whether or not the field is significant.

Options is for optional settings, of which there are a small number, including those that have been added since the protocol was first designed; for example, SACK (see Section 4.5.10). You are not required to know anything further about the options field.

Padding is added as needed, to ensure that the options field comes out as a multiple of 32-bits.

4.5.3 Three-way handshake

The TCP header contains sequence and acknowledgement numbers. Reliable transmission relies on both systems in a connection being able to synchronise their sequence and acknowledgement numbers. Connections are opened and closed with the three-way handshake using specific parts of the header. If a client application on host A wants to establish a connection with a server on remote host B the requirements are as follows.

- Host A requests a connection with B. Host A needs confirmation that a route is available for sending packets to host B, and that B accepts the request.

- Host B accepts the connection request. B needs confirmation that a route is available for sending packets to A.
- Both A and B need to be able to identify packets belonging to this connection, to avoid accepting packets which have been delayed from a previous connection.

These requirements are satisfied as follows:

1. A sends a packet to B with the SYN bit on and a random value x for the Sequence number. Using a new, random Sequence number for each connection means that packets from previous connections will not be accepted.
2. If B receives the packet and accepts the connection, B sends a packet with SYN and ACK bits on, a random number y in its own Sequence field, and $x + 1$ in the Acknowledgement number field.
3. If A receives B's packet it has confirmation that routes are available in both directions and that the sequence numbers have been synchronised. B still needs confirmation of these factors so A sends one more packet to complete the 'handshake', without any flags switched on but Sequence = $x + 1$ and Acknowledgement = $y + 1$.

The three-way handshake has been proved to be sufficient to ensure reliable connection establishment and transmission of data.

4.5.4 Cumulative acknowledgement

The protocol uses positive acknowledgement with retransmission, and operates a **cumulative acknowledgement** scheme. TCP can send segments of varying size, but each segment will have sequence numbers assigned to each byte of data it contains. The receiver reconstructs the byte stream from the segments received. The receiver must acknowledge (ACK) every in order byte it has received. It does this by sending an ACK containing the sequence number of the next byte it expects to receive. The sender starts a timer for every segment that it sends. If it does not receive an ACK before the timer expires it resends the segment.

Under cumulative acknowledgement an ACK implicitly acknowledges receipt of all segments with smaller sequence numbers. Hence out of order segments cannot be acknowledged. Assume that a receiver has received and ACKed up to sequence number 400. It receives some out of order segments, and reorders them, finding that it has bytes 401–600 and 701–900. It ACKs with the number 601, meaning that 601 is the sequence number of the byte it expects to receive next. This demonstrates three things about the cumulative ACK process:

1. Not all segments must be explicitly ACKed.
2. Only in order segments will be ACKed.
3. An ACK implicitly acknowledges all bytes received up to that point.

4.5.5 The adaptive retransmission algorithm

TCP senders start a countdown every time they send a packet. If the countdown expires without an acknowledgement (ACK) of the packet, the sender resends the packet. The problem is, how to decide what is a reasonable amount of time to wait for an ACK? It might be reasonable to wait for the average round trip time, that is the average time for a segment to be sent and an ACK received. In fact the protocol will keep track of round trip times, calculating and recalculating

the average time using **round trip samples**. Round trip samples are simply the time elapsed between transmission and ACKing of a segment. Each time the protocol calculates a round trip sample, it recalculates the average round trip time. Retransmitted packets are excluded from the calculations, since if a segment is sent twice, but ACKed once, is the ACK for the first or second time the segment was sent? There is no way of knowing.

Comer (2014, Section 11.15 *Acknowledgements, Retransmission, And Timeouts*) sums this up as:

To accommodate the varying delays encountered in an internet environment, TCP uses an adaptive retransmission algorithm that monitors delays on each connection and adjusts its timeout parameter accordingly.

In fact the algorithm used by TCP to work out the retransmission time is more complicated than simply maintaining an average of transmission times, but you are not required to know the details.

4.5.6 Duplicate ACKs

When a receiver receives an out-of-order packet, defined to be a segment that does not have a sequence number that is consecutive with the last correctly received byte's sequence number, the receiver ACKs again the last correctly received byte. For example, if the receiver had bytes 100–150, and then received bytes 170–190, it would ACK again byte 150, by sending 151 as the acknowledgement number. If the receiver gets another segment that does not contain byte 151, it responds by ACKing again the last correctly received segment: that is, it will again send an ACK with 151 as the acknowledgement number, the duplicate of the last ACK that it sent. The receiver will continue sending a duplicate ACK for each out of order packet received, until it receives byte 151.

The receiving TCP engine saves any out-of-order data (provided that it has the space to do so). The receiver cannot tell the sender that it has successfully received some of the data after the missing bytes. Comer (2014, Section 11.15 *Acknowledgements, Retransmission, And Timeouts*) notes:

A major disadvantage [of cumulative acknowledgement] is that the sender does not receive information about all successful transmissions, but only about a single position in the stream that has been received.

4.5.7 Duplicate ACKs and fast retransmission

Over the years many changes have been proposed to the TCP protocol, including changes intended to increase transmission times. One of them is fast retransmission, first proposed in 1990, which includes using duplicate acknowledgements to trigger retransmission. The standard incorporated into the protocol is that if the sender receives three duplicate ACKs (the original plus three more) it retransmits the first unacknowledged segment (segments are ordered by sequence number) without waiting for the timer to expire.

4.5.8 Flow control and the sliding window

The receiving computer uses the Window field of the TCP header for flow control: ensuring that the sender does not overwhelm the receiver with data it is not ready to accept or process. Flow control is implemented with what is known as the **sliding window**. The receiving computer will enter into the Window field a range of sequence numbers that it can accept, starting

with the sequence number after the most recently acknowledged sequence number; this is called a **window advertisement**. The sending computer can only send segments with sequence numbers included in the Window field. With each acknowledgement the receiver changes the Window field, discarding the sequence numbers of received bytes and incorporating new sequence numbers, conceptually sliding through the sequence numbers in order. This allows the amount of data that the receiver will accept to grow and shrink during the transmission, according to prevailing conditions, preventing congestion.

If a host is unable to accept data the Window field will be set to 0, and a new segment with a non-zero value will be sent when the host is ready to accept data.

Activity 4.1

Which one of the following statements is **not** true?

1. Transport layer protocols interact with services in the application layer, via ports.
 2. TCP stands for Transmission Control Protocol. UDP stands for User Datagram Protocol.
 3. UDP is unreliable but fast, while TCP is reliable but slower than UDP.
 4. TCP is used to break a message up into packets and reform the packets back into the original message.
 5. UDP is more suitable than TCP for VoIP (voice over Internet Protocol) applications such as Skype, as retransmitting data for real-time services is not usually helpful.
 6. TCP compresses and decompresses packets so they can be transferred efficiently.
 7. TCP includes error checking so any segments that are missing are resent until the entire message can be recreated.
 8. The IPv4 version of the UDP header has two optional 16-bit fields.
 9. TCP's cumulative acknowledgement scheme means that the receiver does not notify the sender of corrupt or lost data packets.
 10. With TCP, a slow receiver can limit how much data a sender can transmit.
-

4.5.9 Retransmission and cumulative ACKs

Once the sender reaches the time out for acknowledgement of a missing segment it sends the segment again (alternatively it resends once it has received three duplicate ACKs). However, the sender does not know if all packets from that point are lost, or if only some, or just one. So the sender could resend the missing segment, and wait for the ACK, to determine what it needs to resend, if anything. For example, if the receiver is ACKing with 401, but by that point the sender has sent segments with sequence numbers up to 1100, then it can send again the segment with sequence number 401 and wait for the ACK to find out if it should resend all data from that point. If it receives an ACK with 1101, the sender knows it does not need to resend any other data. If it receives an ACK with 501, the sender knows that it must resend other segments too. Alternatively, the sender could not wait, and just send again all packets starting with the missing one, which is inefficient. But as Comer (2014, Section 11.15 *Acknowledgements, Retransmission, And Timeouts*) makes clear, so is waiting:

If the sender follows the accepted standard and retransmits only the first unacknowledged segment, it must wait for the acknowledgement before it can decide what and how much to send. Thus, retransmission reverts to a send-and-wait paradigm, which loses the advantages of having a large window.

4.5.10 SACK

SACK, the Selective ACKnowledgement scheme, was first proposed in RFC 2018 (see: <https://tools.ietf.org/html/rfc2018>), in 1996, and was intended to mitigate the disadvantage of the cumulative acknowledgement scheme described above. The abstract of RFC 2018 states:

TCP may experience poor performance when multiple packets are lost from one window of data. With the limited information available from cumulative acknowledgments, a TCP sender can only learn about a single lost packet per round trip time. An aggressive sender could choose to retransmit packets early, but such retransmitted segments may have already been successfully received.

A Selective Acknowledgment (SACK) mechanism, combined with a selective repeat retransmission policy, can help to overcome these limitations. The receiving TCP sends back SACK packets to the sender informing the sender of data that has been received. The sender can then retransmit only the missing data segments.

SACK allows a receiver to give details of gaps in received bytes. Hence if a receiver had bytes 15–18 and 90–99 it could indicate that it was waiting for segments with sequence numbers including 19 and 100. It would do this by including the starting sequence number of each contiguous block (in this case 15 and 90) and the sequence number following on from the last sequence number in the block (in this case 19 and 100). Hence, in the above example, when sending an ACK, the receiver would include 15 and 19 for the first block, plus 90 and 100 for the second. The scheme is not mandatory and does not replace cumulative acknowledgement. Senders indicate in the TCP header that SACK is possible while making the connection.

4.5.11 Congestion control

The designers of the internet assumed there would be few networks, hence the need for congestion control was not anticipated. Clearly, the internet has grown to the point where congestion control is essential, and much research and intellectual effort has gone into efficient ways of handling congestion in TCP. Note that congestion is separate from the connection made by the protocol between two endpoints, because TCP segments have to travel across the internet, forwarded by the Internet Protocol from router to router. Congestion can lead to packet loss, as routers must queue packets that they cannot transmit. Routers do not have infinite space, so may reach a point where incoming packets can no longer be stored for later transmission. Note that the internet is not designed to give priority to a particular TCP connection, so the protocol needs to find a way to negotiate congestion without adding to it.

Since TCP responds to packet loss and delay by resending segments, this could, in theory, increase congestion, to the point that the network can no longer work effectively (known as **congestion collapse**). Comer notes (2014, Section 11.19 *Response to Congestion*):

To avoid congestion, the TCP standard now recommends using two techniques: *slow-start* and *multiplicative decrease*. The two are related and can be implemented easily.

You are not required to know the details of congestion control mechanisms.

Activity 4.2

Answer TRUE or FALSE to each of the following statements. Where you think that the answer is false, state your reasons in no more than **five** sentences:

1. TCP is a Transport layer protocol, while UDP is Application level.
 2. The UDP header consists of 32-bit words, while the TCP header has 64-bit words.
 3. The well-known ports are numbered 0–1023 inclusive.
 4. The well-known port number for HTTP is 423.
 5. For every TCP port there is a UDP port with the same number.
 6. UDP implements flow control to prevent hosts from being overwhelmed with data packets.
 7. TCP implements reliable transmission by positive acknowledgement with retransmission; this means that the receiver will both: (1) confirm receipt of data packets; and (2) send requests for missing packets.
 8. TCP will fragment data packets that are too large.
 9. Under cumulative acknowledgement in the TCP protocol only in-order segments received will be acknowledged.
 10. Under cumulative acknowledgement in the TCP protocol every in-order segment received must be explicitly acknowledged.
 11. A TCP sender who has implemented fast retransmission should, on receiving three duplicate acknowledgements, assume that the data packet immediately following the packet – being multiply acknowledged – is corrupt and send it again.
 12. The SACK scheme allows TCP receivers to notify senders of any gaps in the sequence numbers of the received data stream.
 13. The SACK scheme is compulsory.
 14. Congestion collapse means that network protocols to deal with congestion have temporarily suspended network traffic.
 15. Congestion control was built into the original TCP/IP protocol suite.
-

4.5.12 Active areas of research

TCP is a living protocol, with active areas of research. For example, consider RFC 8547 from May 2019: <https://tools.ietf.org/html/rfc8547>. The RFC's status is 'experimental', so there is more work to be done before what it proposes can be implemented, if it ever is. The abstract states:

a significant fraction of TCP traffic on the Internet remains unencrypted. [...] TCP Encryption Negotiation Option (TCP-ENO) [proposes] a new TCP option kind providing out-of-band, fully backward-compatible negotiation of encryption.

4.6 Overview of the chapter

In this chapter we considered TCP in detail, examined its historical development and the reason for its many complications. We discussed cumulative ACKs, congestion management and flow control in the TCP protocol. We finished by observing that there is active research on ways to improve the TCP protocol, such that may well be amendments or additions to the protocol in future to allow for changing conditions on the global internet.

4.7 Reminder of learning outcomes

Having completed this chapter, and the Essential reading and activities, you should be able to:

- explain in general terms how information in the TCP header is used in:
 - opening and closing connections
 - flow control
 - reliable data transmission.

4.8 Test your knowledge and understanding

4.8.1 Sample examination questions

- (a) (i) Data packets received by the Network Access Layer of the TCP/IP network model, journey in a particular order up through the other layers to reach their destination. This order is: [2 marks]
1. Network Access → Internet → Transport → Application
 2. Network Access → Transport → Internet → Application
 3. Network Access → Transport → Application → Internet
 4. None of the above
- (ii) Which one of the following is a Transport Layer protocol? [2 marks]
1. IP
 2. TCP
 3. SGML
 4. None of the above
- (iii) The 'well-known' ports are assigned by IANA to the most frequently used applications. Which one of the following is true about the well-known ports? [2 marks]
1. The well-known ports are numbered 0–1023, and are also known as the 'System Ports'
 2. The well-known ports are numbered 1024–49151, and are also known as the 'Registered Ports'
 3. The well-known ports are numbered 49152–65535 and are also known as the 'Dynamic Ports'
 4. None of the above
- (b) (i) Why is the **checksum** optional in the UDP datagram header, in the IPv4 version of the protocol? [2 marks]
- (ii) Why is the **source port** field optional in the UDP datagram header, but compulsory in the TCP segment header? [7 marks]

- (c)** (i) Briefly describe how the cumulative acknowledgement process of the TCP protocol works in practice. [4 marks]
- (ii) Explain why the cumulative acknowledgement process can be inefficient when some data packets are not received the first time that they are sent. [6 marks]

Chapter 5: The world wide web and email

5.1 Introduction

For many people now the internet is just the world wide web: a linked collection of clients and servers, which enables users to transfer documents, images and sound files from arbitrarily remote locations without needing any information apart from the name of the server's website. This chapter briefly describes the operation of web servers and browsers and then discusses HTML. The history of HTML is covered in Section 5.4 for background, and is not examinable. The chapter then looks in some details at the current version of HTML, HTML5 and discusses how it is integrated with CSS and JavaScript.

The protocols discussed in this chapter, DNS, HTTP, SMTP, POP and IMAP are all Application layer protocols.

5.1.1 Aims of the chapter

This chapter aims to help you to understand how the world wide web with its client/server model of connected machines differs from the internet with its host/router model. The chapter aims to help you to learn the basics of writing web pages with HTML and CSS; to develop confidence in writing simple web pages; and to help you to develop an appreciation of the technologies used by a web server to support dynamic web pages. The chapter further aims to develop your knowledge and understanding of the technologies underpinning email.

5.1.2 Learning outcomes

By the end of this chapter, and having completed the Essential reading and activities, you should be able to:

- explain in general terms how web documents are transferred across the internet and what processes are triggered when you click on a hyperlink
- code web pages using style sheets
- describe some of the technologies available for dynamic web pages, and be able to select the appropriate type of technology for your requirements
- describe the most widely used email protocols and summarise the important differences between them.

Note that any section whose heading includes **an historical note** is given for background and is not examinable.

5.1.3 Essential reading

Please ensure you read this entire chapter of the subject guide in addition to the links in this section.

Links to the W3Schools tutorials on HTML and CSS:

www.w3schools.com/html/html_colors.asp – using colours with HTML

www.w3schools.com/css/css_colors.asp – using colours with CSS

www.w3schools.com/colors/colors_names.asp – a list of named HTML colours with their hexadecimal equivalent representations

www.w3schools.com/html/html_styles.asp – background colour

www.w3schools.com/css/css_font.asp – fonts and CSS

www.w3schools.com/html/html_images.asp – using images with HTML

www.w3schools.com/html/html_filepaths.asp – file paths in HTML

www.w3schools.com/html/html_lists.asp – lists in HTML

www.w3schools.com/html/html_attributes.asp – HTML attributes, including the language attribute

www.w3schools.com/css/css_syntax.asp – syntax for CSS

www.w3schools.com/cssref/pr_background-image.asp – how to set a background image on a web page using CSS

www.w3schools.com/cssref/pr_font_font-family.asp – the CSS font-family property

www.w3schools.com/html/html_tables.asp – tables in HTML

www.w3schools.com/css/css_table.asp – table formatting in CSS, including border-collapse, width of a table and aligning text.

www.w3schools.com/tags/tag_tfoot.asp – example, of a table with a header, footer and body sections.

5.1.4 Further reading

- Castro, E. and B. Hyslop *HTML and CSS: Visual Quickstart Guide*. (San Francisco, CA: Peachpit Press, 2013) 8th edition [ISBN 9780321928832]. Chapters 1–8, Chapter 10, Chapter 15 and Chapter 18.
- Casad, J. *Sams Teach Yourself TCP/IP in 24 Hours*. (Indiana: Pearson Education, 2017) 6th edition [ISBN 9780672337895]:
 - **Hour 16:** The Internet: A closer look: Heading URLs and URLs
 - **Hour 17:** HTTP, HTML, and the World Wide Web
 - **Hour 20:** Email
- Comer, D.E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) 6th edition Pearson new international edition [ISBN 9781292040813].
 - Chapter 23 *The Domain Name System (DNS)*: Sections 23.1–23.13 inclusive
 - Chapter 24 *Electronic Mail (SMTP, POP, IMAP, MIME)*
 - Chapter 25 *World Wide Web (HTTP)*: Sections 25.1–25.11 inclusive
 - www.w3.org/ – definitive specifications for web technologies, including HTML and CSS
 - www.w3schools.com/ – tutorials for HTML, CSS, JavaScript, server-side technologies including PHP and SQL, plus other web software.

5.1.5 References cited

Berners Lee, Tim *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. (HarperBusiness, 2000) [ISBN 9780062515872].

- www.wired.com/2016/08/linux-took-web-now-taking-world/
- www.theregister.co.uk/2018/09/17/linus_torvalds_linux_apology_break/
- www.ietf.org/rfc/rfc1866.txt
- <https://tools.ietf.org/html/rfc2854>
- www.w3.org/MarkUp/Cougar/
- www.ietf.org/rfc/rfc2068.txt

- <https://news.netcraft.com/archives/2018/09/24/september-2018-web-server-survey.html>
- www.theregister.co.uk/2010/01/29/google_web_server/
- https://groups.google.com/forum/#!msg/comp.archives/LWVA50W8BKk/wyRbF_IDc6cJ
- www.ietf.org/rfc/rfc1034.txt
- www.ietf.org/rfc/rfc1035.txt
- www.icann.org/resources/pages/domain-name-industry-2017-06-20-en
- www.w3.org/Help/
- www.ietf.org/rfc/rfc2068.txt
- https://edps.europa.eu/node/3100#e-privacy_directive2009-136-ec
- www.w3schools.com/html/html_xhtml.asp
- www.ietf.org/rfc/rfc1866.txt
- <https://tools.ietf.org/html/rfc2854>
- www.w3.org/MarkUp/Cougar/
- <https://validator.w3.org/nu/#textarea>
- www.w3schools.com/css/css_intro.asp
- www.w3schools.com/html/html5_syntax.asp
- www.w3schools.com/php/php_intro.asp
- <https://tools.ietf.org/html/rfc5321>

5.2 Glossary of key terms

- **World wide web:** a system that links HTML documents to each other, using hyperlinks accessed via the internet.
- **W3C (The World Wide Web Consortium):** an international organisation led by Sir Tim Berners-Lee that is responsible for standards for the world wide web, including HTML and CSS.
See: www.w3.org/
- **WHATWG (The Web Hypertext Application Technology Working Group):** was started in 2004 to push for a new standard of HTML more useful to industry and ecommerce, and was largely responsible for the development of HTML5. WHATWG is now overseen by a steering group of representatives from Apple, Google, Microsoft, and Mozilla and currently has a role in developing new standards for the world wide web.
- **W3Schools:** the largest site for web developers, with many free tutorials about HTML, CSS and other web technologies. Not affiliated to the W3C, although it derives its name from it. See: www.w3schools.com/about/default.asp
- **Client/server model:** the world wide web is organised into clients and servers. Servers wait for requests, and fulfil requests if they can, or send an error message to the requesting client. A machine can be both a client and a server.
 - **Client:** sends requests to the server, such as a web browser requesting to view a web page, or an email client sending a message to a mail server
 - **Server:** waits (listens) for requests from clients, fulfils valid requests and sends an appropriate response. Examples are a mail server listening for incoming email messages, and a web server sending HTML documents when requested by a web browser.

- **Linux:** an open source operating system that powers much of the servers behind the world wide web.
- **Open source software:** software that is freely available, together with its source code. Developers are free to modify the source code. Modified software can be shared freely or sold, but the source code must be freely available in a form that developers can edit. There are other restrictions which can be found here: <https://opensource.org/osd>
- **CERN** operates the European Laboratory for Particle Physics, which includes the Large Hadron Collider. Its name is an acronym of the French for European Council for Nuclear Research. Currently, 23 European states are member states, contributing to capital and operating costs.
- **SGML (Standard Generalised Markup Language):** not a markup language, but a meta-language, in that it specifies how a markup language should be written. SGML specifies tags to markup the structural elements in text (for example, paragraphs, headings). A language derived from SGML has a DTD – a document type definition – that describes how its tags should be interpreted.
- **Hypertext:** a way of providing links to different content to help readers to navigate a document. Used for links in a web page to another web page, or to another part of the same web page.
- **FTP:** The File Transfer Protocol, how files were shared on the internet before the World Wide Web was invented. The protocol is still supported and used.
- **HTTP (Hypertext Transfer Protocol):** the Application layer protocol underlying the world wide web, developed by Tim Berners-Lee starting in 1989. The protocol communicates data in HTML format using the client/server model.
- **HTML (Hypertext Markup Language):** a protocol for marking up documents to be viewed with a computer linked to the internet via the world wide web. Documents are structured using tags to denote structural elements with attributes to add formatting. Web browsers read the tags in order to understand how to display the web page to the viewer.
- **HTML element:** 'An HTML element usually consists of a start tag and an end tag, with the content inserted in between'. See: https://www.w3schools.com/html/html_elements.asp
- **HTML5:** the current version of HTML, which provides strong support for JavaScript for interactivity and CSS for presentation. Users are expected to use HTML tags for the structural elements of a document (for example, paragraphs, headers and footers) and CSS for presentation and formatting.
- **DOCTYPE:** the keyword DOCTYPE is used to declare the DTD, used to specify the meaning to be applied to tags for structural elements in languages derived from SGML, such as HTML. Each HTML document starts with a DOCTYPE declaration that lets the web browser know what version of HTML the page is written in, and hence how to interpret tags in order to render text and images.
- **Character encoding:** characters (letters, numbers, punctuation symbols, etc) are given numerical references. Character encoding schemes are used to tell a computer how to interpret and display text characters.
- **ASCII (American Standard Code for Information Interchange):** a character-encoding scheme designed to represent the printable and control characters needed to write American English on a computer.

- **UTF (Unicode Transformation Format):** a set of character encodings developed by the computer industry and designed to be able to represent all alphabets, not just the English one. The UTF-8 encoding has become the default encoding for the world wide web.
- **CSS (Cascading Style Sheets):** a language used for presentation of a document. CSS can be incorporated directly into HTML files, or CSS formatting can be written into an external style sheet file, to which a web page or pages can link. A single CSS external style sheet can be linked to an unlimited number of web pages. CSS styling has three levels:
 - **Inline:** applies to a single element and has the highest level of precedence
 - **Document:** applies to the entire web page
 - **External style sheet:** CSS formatting saved in a separate file for HTML documents to link to; has the lowest level of precedence.
- **JavaScript:** a scripting language used for user interactivity on web pages, strongly supported by HTML5.
- **Scripting language:** a scripting language is a language that can be used as a bridge between applications speaking different languages. One of the most common uses of a scripting language on the web is to send a command to the back end database in SQL, and return the result to the front end web page formatted in HTML.
- **Web browser:** a client application allowing users to view and interact with web pages on remote machines using a connection to the internet. Browsers read HTML files and use the HTML (and any CSS) code to tell them how to display the page. HTML files on the local machine can also be displayed.
- **Web server:** a web server processes network requests received from the HTTP protocol.
- **Search engine:** a search engine indexes the title and content of web pages, storing the results in a database. Users connect to the search engine with a web browser and enter their search terms. The search engine queries its database for the search terms and returns results with links to the web pages containing the terms.
- **Domain name:** invented to make internet addresses easier for humans to read, understand and remember, domain names are composed of discrete blocks of text (labels) separated by dots. Each domain name starts with the most local label, and ends with a label known as the top-level domain name. Every domain name matches to at least one IP address.
- **Top-level domain name:** the final part of a domain name, with the widest scope and application. Domain names are grouped and organised under their top-level domains. Top level domain names, in general, are either **country specific** with each country having a two-letter code, or **generic**, describing particular types of organisation such as **edu** for education.
- **Sub-domain:** in a domain name the hierarchy of domains descends from right to left; each label to the left specifies a subdivision, or sub-domain of the domain to the right. Hence each time the leftmost label is removed gives another sub-domain. For example, in www.gold.ac.uk the lowest sub-domain is www.gold.ac.uk, which is a sub-domain of **gold.ac.uk**. Removing the left-most label, we find that **gold.ac.uk** is a sub-domain of **ac.uk**, and once more removing the left most label reveals that **ac.uk** is a sub-domain of **uk**, the top-level domain name, one of the two letter country codes.


- **DNS (Domain Name System):** an Application level protocol that coordinates a system of databases recording domain names, and has a logical method of matching IP addresses to domain names.
- **URI (Universal Resource Indicator):** is a way of identifying a resource on the internet.
- **URL (Universal Resource Locator):** is a way of making a request over the internet, and is a special case of a URI. Requests can be for various services including FTP and HTTP. A URL has a scheme (HTTP or HTTPS for web pages) followed by the location of the server, and the path to the desired resource on the server. The location of the server and resource is given with either an IP address or via the domain name system. For example:

`www.gold.ac.uk/computing/`

In the above URL, the scheme is HTTPS, the domain name is www.gold.ac.uk and the path to the local resource is /computing. A URL always has the above information (scheme/address of server/path to resource), but can optionally include other information such as the port number of the service it is requesting.

- **Sandbox:** software that provides an environment to run and test programs in, isolated from critical systems.
- **Stateless:** a stateless protocol is one that does not require an implementing service to keep track of its internal state. For example, the IP protocol is designed such that all necessary information is contained in the IP datagram header; hence routers have no need to keep track of datagrams, or to remember previous datagrams. Servers can treat each IP datagram received as a discrete entity with no connection to another; that is, they forward, deliver or discard a datagram as appropriate then deal with the next one, without saving or accessing any information from the previous.
- **Cookies:** are text files placed on the local machine by a remote service with information that the service can access in order to keep track of a particular session between a client/server pair, or for other purposes such as tracking the local machine's actions. Cookies range from the useful, for example, keeping a user logged in to a particular page, to tracking cookies that may be used by spyware. However, as simple text files, cookies themselves cannot be malicious code. See Chapter 6 for data protection issues and cookies and Chapter 7 for more on security issues and cookies.
- **Fault tolerance:** protocols that attempt to deal with errors in such a way that they do not cause the system to slow, stop or crash. HTML is fault tolerant, meaning that a browser will display as much information as it can, by interpreting as much of the HTML and CSS code as it can, and ignoring the rest.
- **XML:** 'XML was conceived as a means of regaining the power and flexibility of SGML without most of its complexity. Although a restricted form of SGML, XML nonetheless preserves most of SGML's power and richness, and yet still retains all of SGML's commonly used features.' See: www.w3.org/TR/xhtml1/
- **XHTML:** HTML rewritten with the rules of XML.
- **The semantic web:** a dream of Sir Tim Berners-Lee, the idea that all web pages should be entirely machine-readable.
- **Well-formed:** well-formed HTML5 syntax is that which is compatible with XML. W3Schools recommends its use in order to 'future-proof' web pages.

The recommendation is probably made because the W3C still harbours the dream of the semantic web, and having web pages that an XML reader could completely decode would be an important step towards the semantic web.

- **Progressive enhancement:** an approach to web page development, that starts with the core of the web page design, written with HTML code that is accessible to all browsers, and then adds enhancements that the browser may or may not be able to display. The idea being that all users, whatever device (small or large) and web browser they are using, can access at least enough of the page to see its core message or access its core service.
- **Deprecated:** something that exists, but its use is no longer encouraged or supported by the originating organisation.
- **Hexadecimal:** a number system in base 16 that uses the digits 0–9 and A–F to represent numbers.
- **Plug-ins:** Plug-ins are pieces of software that adds something to existing software, usually a utility that the existing software does not provide. Since earlier versions of the HTML protocol were static, meant for display and not interactivity, developers used plug-ins to give their web pages the interactivity needed for ecommerce and other services. Most web-based plug-ins are deprecated now (for example Flash and Java applets), not just because there is less need for them since HTML5 but also because they were notorious security holes. See Chapter 7 for more on security issues and plug-ins.
- **Dynamic web pages:** web pages that interact with the user. Historically, interactivity used to be provided by various plug-ins, but now the use of HTML5 and JavaScript is encouraged by the W3C and others.
- **SMTP (Simple Mail Transfer Protocol):** is the most commonly-used protocol for sending email. SMTP follows the client-server model, with the end-user's email application as the client. The server will be a mail server that is always listening for connection requests.
- **Hamburger icon:** The hamburger icon or button , is often included in graphical user interfaces, and will be familiar to many users from web browsing. Its nickname derives from its similarity to a hamburger. Behind the hamburger button is a menu or some other optional display item.
- **Spoofing:** SMTP is vulnerable to spoofing. Email spoofing is when an email is sent with a false address in the 'From' field of the header. Email spoofing is possible because SMTP commands are not linked to header fields. Thus it is possible to give one correct address to the *MAIL FROM command*, and put a completely different address in the 'From' field of the header that will be seen by the recipient. The protocol will not check and flag up the contradiction. See Chapter 7 for more on email and security issues.
- **Spam:** unsolicited emails that are typically seeking to introduce the recipient to hitherto undreamt of opportunities in educational, medical, gambling and other contexts. The offers contained in spam emails are sometimes legal, but often completely fraudulent. Spam emails are usually sent to millions of email addresses at the same time. See Chapter 7 for more on email and security issues.
- **MIME (Multipurpose Internet Mail Extensions):** the protocol that specifies, among other things, how non-ASCII data such as word processed documents, spreadsheets, images, etc, should be transmitted as email attachments.

- **POP3 (Post Office Protocol version 3):** a protocol for retrieving email messages. POP enforces that messages must be downloaded from the mail server before they can be read and replied to; messages are normally deleted after being downloaded. POP is suitable for people who only want to use one device for reading and sending messages.
- **IMAP:** a protocol for retrieving email messages that allows the user to read, reply-to, forward and organise their email messages on a server. IMAP is suitable for users who will use more than one device (for example, a workstation, a home laptop and a smart phone) to access their email.

5.3 An historical note: Linux

Much of the world wide web is run by Linux, open source software, as *Wired* noted in August 2016:

Linux – is one of the most important pieces of computer software in the world. Chances are, you use it every day. Linux runs every Android phone and tablet on Earth. And even if you're on an iPhone or a Mac or a Windows machine, Linux is working behind the scenes, across the Internet, serving up most of the webpages you view and powering most of the apps you use. Facebook, Google, Pinterest, Wikipedia – it's all running on Linux.

Plus, Linux is now finding its way onto televisions, thermostats, and even cars. As software creeps into practically every aspect of our lives, so does the OS designed by Linus Torvalds.

See: www.wired.com/2016/08/linux-took-web-now-taking-world/

In 1991, Linus Torvalds, a Finnish computer science student, started to write a free operating system, as a project. This was the start of the Linux operating system, based on the Unix OS created by Bell Labs in 1969. Bell Labs was part of AT&T, a US telecommunications company. Originally established in 1925 as Bell Telephone Laboratories Inc, Bell Labs was an important and famous research laboratory, responsible, among other things, for inventing the transistor.

Unix could only be run on powerful machines, and was proprietary software that could not be edited or changed by system administrators. Richard Stallman started the open source software movement in 1983, when he founded GNU (GNU's Not Linux) in order to rewrite the Unix operating system as open source software, motivated by a belief that software should be free, with the source code available so that all who want to can play a part in its development. By 1991, the project was complete except for one important thing: no kernel. The kernel is the part of the operating system that is always in memory and interacts with the CPU to control the machine.

In 1991, Torvalds wrote the Linux kernel, and this was incorporated into GNU and the resulting operating system quickly became popular with developers, computer hobbyists and system administrators. As the web grew, so did Linux, in part because it was free, and in part because other free software available at the time of its release was caught up in a legal battle with AT&T, who claimed infringement of their intellectual property rights, because they thought that some of their original Unix code had been reused (instead of rewritten entirely).

In the early web, the Linux kernel was used on web servers far more than Windows, partly because Windows had to be paid for and in part because Windows was not seen as reliable. System administrators regularly claimed that a Linux server could run for more than a year without crashing, and contrasted this very unfavourably with the reliability of Windows' servers.

While Windows servers are much more reliable now, that early head start meant that Linux servers came to dominate the web, and then, with the Android operating system released in 2008, came to dominate mobile phones too. *Wired* further reported on the dominance of Linux in 2016 with the following:

Android now dominates the smartphone market. According to industry research firm Gartner, the operating system accounted for about 84 percent of the market during the first quarter of 2016. But Linux's reach now extends so much further than smartphones. You can already find Linux in smart TVs from companies like Samsung and LG, Nest thermostats, Amazon's Kindle e-readers, and drones from companies like 3DR. [...] And when self-driving cars finally hit the road, you can bet they'll be powered by Linux.

Companies turn to Linux today when they want to build new technology for the same reason that web developers turned to the operating system in the 1990s: they can customize it to meet their needs, and then share (or sell) the results without having to get permission.

See: www.wired.com/2016/08/linux-took-web-now-taking-world/

In September 2018 *The Register* reported:

The Linux kernel sits at the heart of hundreds of millions of Android devices, as well as countless internet and enterprise servers, supercomputers, Chromebooks and other PCs, routers and networking equipment, Internet-of-Things gizmos and other embedded electronics, and so on.

Torvalds still manages kernel releases, all crafted from source code submitted by more than 10,000 developers from more than 1,200 organizations worldwide and filtered through a team of maintainers who are responsible for various components that make up the project – from device drivers to memory management.

See: www.theregister.co.uk/2018/09/17/linux_torvalds_linux_apology_break/

5.4 An historical note: HTML

When Tim Berners-Lee wrote HTML in 1989 he was working at CERN and wanted a simple way to use the internet to share information. At that time CERN had various powerful and personal computers, from different manufacturers, whose proprietary software hampered the sharing of information, or as Berners-Lee wrote in his book *Weaving the Web*: 'In those days, there was different information on different computers, but you had to log on to different computers to get at it. Also, sometimes you had to learn a different program on each computer. Often it was just easier to go and ask people when they were having coffee'. Berners-Lee called his new language Hypertext Markup Language. 'Hypertext' signalled the intention of using the language to embed links in documents. 'Markup' meant the language was designed to structure the display of text; namely to mark up what parts of the text were headings, what were paragraphs, what were list items, etc.

Berners-Lee based HTML on SGML (Standard Generalized Markup Language), taking from the internationally agreed standard for specifying a markup language HTML tags for paragraphs, headings and lists, among others. To this

Berners-Lee added hypertext, a way of embedding links to documents on remote machines. This was done with a Hypertext Reference (called HREF in HTML syntax), which comprised a URL, and some 'anchor' text; that is, text that the user would click on to access the linked document. Hypertext links had been in use since the 1960s, originally to make links with files on the local machine.

The URL part of the hypertext was designed by Berners-Lee to be a way of requesting a resource from a computer. A URL gives the location of the server, of the resource wanted on the server and the protocol – or scheme – that will be used to access it. Berners-Lee invented the `www.name1.name0` format, using the DNS system to make links more accessible and easier to implement. 'www' was used to indicate that the site was part of the new world wide web, and is a sub-domain of the DNS (see Section 5.5.5 for more information on the Domain Name System). URLs were in use from 1990 in Berners-Lee's original web browser, although Berners-Lee did not formally specify URLs until RFC 1738 in 1994, see: www.ietf.org/rfc/rfc1738.txt

Berners-Lee and others started the World-Wide Web global information initiative in 1990, a way for Berners-Lee to pull other academics and researchers into a discussion about developing the nascent web. Once Berners-Lee left CERN he started The World Wide Web Consortium (W3C) in 1994, which develops web standards, see: www.w3.org/

HTML documents are written in plain text, and this, combined with the small number of tags in the prototype, meant that within a few hours anyone could master writing HTML. Hence anyone with a computer could write web pages using a simple text editor, and post them online for little or no money. This was a factor in the rapid growth of the early web, another being the web browser Mosaic, developed in the United States at the National Center for Supercomputing Applications (NCSA), released in 1993 and the most popular of the early web browsers.

RFC 1866, codifying HTML practice into HTML 2.0, was published by the Internet Engineering Task Force, in 1995: www.ietf.org/rfc/rfc1866.txt. The RFC aimed to standardise common practice, or as Section 1.1 noted:

HTML has been in use by the World-Wide Web (WWW) global information initiative since 1990. Previously, informal documentation on HTML has been available from a number of sources on the Internet. This specification brings together, clarifies, and formalizes a set of features that roughly corresponds to the capabilities of HTML in common use prior to June 1994.

5.4.1 Static HTML

HTML was static in its first incarnation; that is, it did not change as it was viewed, the user could not interact with the text except by clicking on a link. As the web grew companies soon wanted interactivity, for example, Amazon.com, founded in 1994, could not succeed as a business unless there was some way for the user to add items to their shopping basket, to enter an address for delivery and to pay. Because HTML could not do these things developers used plug-ins, or extensions, and browser vendors developed their own technology. For example, Netscape, whose browser dominated the early web, developed scripting; in fact, their engineers developed JavaScript, which has no link with Java despite the name. Java applets were also used for interactivity and Flash for video. Java applets are small programs that executed on the user's machine. Microsoft developed ActiveX technology for developers to write plug-ins for Internet Explorer.

The drawback was that plug-ins could be attack vectors for hackers; since they were run without sandboxing, they could give access to the operating system. In fact, in time Java applets and Flash became very popular attack vectors; ActiveX technology also had security issues. In addition the proprietary nature of plug-ins went against the internet ideals of protocols that can be implemented on any hardware and work the same, as they only worked on supported platforms. Finally, plug-ins were often responsible for crashes.

Java applets are now deprecated – which means that they still exist, but are no longer supported by Sun and their use is discouraged. In 2015, Microsoft released Microsoft Edge, the successor to Internet Explorer. This had no support for ActiveX, bringing the technology to an end. Instead the new browser supports HTML5 and JavaScript. Flash still exists, but is used less and less now that HTML5 allows audio and video to be embedded directly in web pages.

5.4.2 Browser wars and proprietary tags

Netscape Navigator, developed by some of the team behind Mosaic, was released in 1994, and quickly became the dominant web browser, eclipsing Mosaic. Microsoft, which was rather late in grasping the importance of the developing world wide web, released Internet Explorer in 1995, at which point Netscape Navigator had been available for download since the previous year, and, despite competition, dominated the market. The late 1990s were a time when Netscape Navigator and Microsoft fought for market share and dominance, a battle that Microsoft was destined to win, when Internet Explorer version 4 was released in 1997, and bundled with Windows, which then had more than 90 per cent of the desktop market. By 2002, Netscape Navigator had lost its entire market share, which once had been more than 80 per cent.

During the period in the 1990s when Microsoft and Netscape fought for dominance – known as the ‘browser wars’ – both sides were developing their browsers faster than HTML was officially developing. In fact, a great deal of work was being put into developing the HTML standard by the IETF HTML Working Group but agreement was hard to come by. The draft specification for HTML 3.0 was published by the IETF in 1995, and an email discussion group was started, open to anyone with an interest in the development of HTML. As it turned out there were rather a lot of them, and this made agreement hard to reach, especially as the draft specification was ambitious in scope. By December 1995 the Working Group had run out of steam, and was closed in 1996 without agreement of HTML 3.0:

The IETF HTML working group closed Sep 1996, and work on defining HTML moved to the World Wide Web Consortium (W3C) .

See: <https://tools.ietf.org/html/rfc2854>

By this time Berners-Lee had started W3C, which became formally responsible for developing HTML standards in 1996, when the IETF working group closed.

Without agreement on the protocol, the commercial vendors pressed ahead with innovations such as scripting, frames, and their own proprietary tags for marking up text. This meant websites that specified whether IE or Netscape would be best used to view their pages. Web developers would write their pages with one of the browsers in mind for display, while trying to make sure that their pages would display some minimum information if viewed with a different browser. The scientists and engineers responsible for developing standards worried that proprietary tags could divide the web into Netscape and Internet Explorer dominions, neither of which could talk to the other. Once HTML 3 was abandoned, the W3C began to discuss with browser

companies their current practices for incorporation into what became HTML 3.2 (apparently a decision was made to skip right over HTML 3.1). This was achieved through the formation of the HTML Editorial Review Board (ERB), comprising industry representatives including Microsoft and Netscape, as well as scientists and engineers. The goal was to give web page developers and browser companies a common framework, even one that was much less technologically advanced than the abandoned HTML 3. The W3C formally approved HTML 3.2 in 1997.

The ERB had started before the W3C had officially taken over HTML development from the IETF, but after this happened it officially became the HTML Working Group, and started work on project 'Cougar' – which would eventually become HTML 4:

This [HTML 4.0] is being developed by the W3C HTML Working Group, which includes Adobe, Hewlett Packard, IBM, Microsoft, Netscape Communications Corporation, Novell, SoftQuad, and Sun Microsystems.

[...]

What's new in HTML 4.0?

HTML 4.0 builds upon previous work on W3C's Recommendation for HTML 3.2. Among the extensions are style sheets, scripting, improvements for printing, forms and tables, better access to HTML for people with disabilities, the addition of frames, and support for the world's languages including right to left and mixed text.

See: www.w3.org/MarkUp/Cougar/

A working draft of HTML 4.0 was published in 1997, and the standard was formally endorsed by the W3C in 1998. HTML 3.2 and 4.0 introduced features that had originally been proposed for 3.0.

5.5 How the world wide web works: the client-server model

Tim Berners-Lee and others started the World-Wide Web global information initiative in 1990, a way for Berners-Lee to pull other academics and researchers into a discussion about developing the world wide web. Berners-Lee started The World Wide Web Consortium (W3C) in 1994, which develops web standards, see: www.w3.org/

The world wide web operates on a client/server model. Servers wait for requests from clients, and when a valid request is received, carry out the request and send a response to the client. Examples of servers are: file servers, web servers and name servers; examples of client applications are web browsers and email clients. The internet is composed of hosts and routers without overlap between the two (a router cannot be a host, and a host should not be configured to also be a router, as the protocols do not support this). It is possible for a machine to be both a client and a server. RFC 2068 defines 'client' and 'server' as follows:

`client`

`A program that establishes connections for the purpose of sending requests.`

`server`

`An application program that accepts connections in order to service requests by sending back responses.
Any given program may be capable of being both a client`

and a server; our use of these terms refers only to the role being performed by the program for a particular connection, rather than to the program's capabilities in general. Likewise, any server may act as an origin server, proxy, gateway, or tunnel, switching behavior based on the nature of each request.

See: www.ietf.org/rfc/rfc2068.txt

5.5.1 The Hypertext Transfer Protocol

The **Hypertext Transfer Protocol (HTTP)** is a protocol for communicating data in HTML format using the client/server model. HTTP is the Application layer protocol underlying the world wide web. Development was started by Tim Berners-Lee in 1989, the protocol was in use from 1990, with the protocol specification first formally defined as HTTP 1.1 by RFC 2068 in 1997. The HTTP protocol implements the client/server model, in that it defines the actions of a web server in responding to requests from clients. Clients are defined as 'user agents' in RFC 2068 as follows:

```
user agent
    The client which initiates a request. These are often
    browsers, editors, spiders (web-traversing robots), or
    other end user tools.
```

5.5.2 Web browsers

A web browser is a program such as Internet Explorer, Chrome, Opera or Safari that can retrieve files from the world wide web and render text, images and sounds encoded in the files. Files are retrieved via a URL. Browsers can also render HTML files stored on the local machine.

5.5.3 Web servers

A web server processes network requests received via the HTTP protocol. Apache (<http://httpd.apache.org/>), open source software available since 1995, was the most popular HTTP server software on the early web, and played an important part in its growth and development. A typical HTTP server in the early web would run the Linux kernel and Apache server software. Microsoft also released web server software in 1995. Nginx (www.nginx.com/) open source and released in 2004, is another popular web server.

Netcraft's September 2018 Web Server Survey, reported that in September 2018 Apache had 23 per cent of the market share of all sites, with Microsoft on 36 per cent and Nginx on 19 per cent. The rest of the market share was taken by others, with the largest being Google at 1 per cent. However, looked at in terms of market share of active sites (sites that are regularly updated) then Apache had 39 per cent, Microsoft 6 per cent and Nginx 22 per cent. The rest of the market share was taken by others, of whom the largest was again Google, at 8 per cent. See: <https://news.netcraft.com/archives/2018/09/24/september-2018-web-server-survey.html>

The Google web server is proprietary software that runs on Linux, and is only used by Google, on its own sites and on sites that the company operates for third parties. *The Register*, in 2010, called these sites 'a worldwide proprietary infrastructure that amounts to a private internet'; see: www.theregister.co.uk/2010/01/29/google_web_server/. Today Google encourages small businesses and others to use its cloud services for web hosting, which might help to account for 8 per cent of active websites using Google server software in September 2018.

5.5.4 Search engines

The first search engine is believed to be 'Archie', written to search for files on the internet in 1990, just before the invention of the world wide web. At that time files were shared over the internet using FTP. The authors developed a list of FTP servers, and wrote Archie (short for 'archive') to rotate logging in to each one, at night, over a month, automatically listing files on the server and saving the results into text files. Hence results for users could be up to one month old. In a post written in 1990, one of the project's authors noted that Archie currently had no database to save search results in (one was planned), and knew of 210 FTP servers. See: https://groups.google.com/forum/#!msg/comp.archives/LWVA50W8BKk/wyRbF_IDc6cJ

Today, search engines use a web crawler, also known as a web spider, to travel from link to link on the internet, saving the unique URLs that it finds, so that the search engine can index them. The first search engines just indexed web page titles, but soon moved on to indexing the text of web pages too. The results are stored in a database, and it is this database that is queried when a user starts a web search.

Netcraft's September 2018 Web Server Survey, reported more than 1.6 billion host names, and nearly 185 million active sites. See: <https://news.netcraft.com/archives/2018/09/24/september-2018-web-server-survey.html>

5.5.5 The Domain Name System (DNS)

IP addresses, as discussed earlier, are binary strings, and not very easy for humans to remember even when given in dot decimal format. Comer calls IP addresses 'low level names' with users preferring 'high level names'; that is, names that have meaning for them, so the early internet allowed symbolic names:

[W]hen the IT main departmental computer was connected to the Internet in 1980, the Computer Science department at Purdue University chose the name *purdue* to identify the connected machine. At the time, the list of potential conflicts contained only a few dozen names. By mid-1986, the official list of hosts on the Internet contained 3100 officially registered names and 6500 official aliases. Although the list was growing rapidly in the 1980s, most sites had additional machines (typically, personal computers) that were unregistered. In the current Internet, with hundreds of millions of machines, choosing symbolic names is much more difficult.

(Comer 2014, Section 23.2 *Names For Computers*)

The original list of names was what Comer calls a **flat namespace** meaning that the names had no hierarchy. A site called the **Network Information Center (NIC)** kept a list of names and monitored it for any clashes. As Comer notes, this system could not have sustained the massive growth of the internet: 'imagine a central authority trying to handle the current internet where a new computer appears approximately 10 times per second' (Comer 2014, Section 23.3 *Flat Namespace*).

To address this, the Domain Names System (DNS) with a hierarchical name space was introduced. The DNS was specified in 1987 by RFCs 1034 and 1035 (www.ietf.org/rfc/rfc1034.txt; www.ietf.org/rfc/rfc1035.txt) which:

- specified the syntax of names (including that the boundary of each hierarchical unit would be denoted by '.')

- determined how authority for naming should be distributed
- specified the protocol for mapping names to addresses
- specified that names should be distributed among a network of domain name servers, with different parts of the domain space allocated to separate servers, with some redundancy.

Distribution of the domain name space among servers allows for more efficient searching, and prevents having a large single point of failure, and, as Comer (2014, Section 23.25 *Summary*) notes:

Hierarchical naming systems allow delegation of authority for names, making it possible to accommodate an arbitrarily large set of names without overwhelming a central site with administrative duties

The Domain Name System is an Application-layer protocol that coordinates a worldwide network of databases to maintain a mapping between domain names and numerical IP addresses. ICANN is responsible for setting the top-level domain names, which are names under which all domain names are grouped. The domain name system can be viewed as a tree. At the top of the tree is the root name server, run by IANA, which deals with requests for the appropriate name servers for top-level domain names. At the second level are the top-level domain names, with various sub-domains below. The tree can be up to 27 levels deep, although a full 27 levels in any particular branch would be unusual.

5.5.6 Domain names

A domain name is hierarchical, consisting of blocks of text (labels), separated by a dot. At the start of the text string is the label of the local domain, the one with the narrowest scope, and at the end is the label known as the **top-level domain name**. For example, **wired.com** has two labels, **wired** is the local label, and **com** is the top-level domain name. Or in this example: **science.mit.edu**, **science** is the most local (the School of Science at MIT) label, the second level domain label is **MIT**; and the top-level domain name is **edu**. In the hierarchical domain name system, domain names are composed of sub-domains, with the lowest level sub-domain consisting of the entire domain name, and, moving to the left gives the next level sub-domain. For example, **science.mit.edu** is a sub-domain of the domain immediately to the left, which is **mit.edu**. Naturally **mit.edu** is a sub-domain of **edu**, the top-level domain.

The top-level domain names (for example .gov, .com) specify the role or type of the domain. They are divided into country codes (each country has a two-letter code) and type of organisation (for example, .edu for education, .com for commerce). You can find a list of top-level domain names here: <http://data.iana.org/TLD/tlds-alpha-by-domain.txt>; at the time of writing there are 1,531 of them. All other levels of the domain name system are delegated by ICANN; for example, various national authorities are responsible for addresses registered under their particular country code top-level domain.

Registries are responsible for keeping a record of all domain names registered under each top-level domain, and they set the rules for registering names in their name space. **Registrars** are licensed by ICANN and have arrangements with registries to register domain names. **Resellers** are organisations either linked to or contracted by registrars to 'register domain names and offer other services provided by the registrars such as web hosting'; see: www.icann.org/resources/pages/domain-name-industry-2017-06-20-en. Resellers have no agreement with ICANN itself.

ICANN has delegated the top-level domain names to more than 2,000 registrars who are able to sell domain names, either directly or through agreements with resellers. For example, a commercial organisation may apply to a registrar or reseller for the domain name **MyDodgyScheme.com**, and it may be sold to them; or they may be told that the name already exists, and to think again.

Once a customer has registered a name with a registrar or reseller, the customer is then responsible for setting domain names with their sub-domain, and can delegate further. For example, an organisation might register the name **ABCXYZ.org**. They might decide to have sub-domains **right.abcxyz.org** and **left.abcxyz.org**. In theory they could delegate both sub-domains to other organisations to manage.

Hence a domain name is a hierarchical name, with each level in the hierarchy, known as a label, divided from the next by a dot. Domain names progress from the most local and specific label at the start, to the most generic and widest in scope at the end. Each part of the name represents a level of the naming hierarchy (there can be up to 27) and authority for coordinating and assigning names at certain levels can be delegated to various authorities and commercial organisations.

5.5.7 Uniform Resource Locators

Tim Berners-Lee famously invented HTML while working at CERN in 1989, publishing a simple web browser in 1990. When Tim Berners-Lee developed HTML, he brought together markup languages (in development since the 1960s), the internet and hypertext links to documents on remote machines. Of these three things, it was the links to documents on remote machines that Berners-Lee invented himself.

According to the W3C:

The World-Wide Web (W3) initiative is a practical project designed to bring a global information universe into existence using available technology.

Tim Berners-Lee wrote a proposal in 1989 for a system called the World Wide Web. He then wrote the first Web browser, server, and Web page. He wrote the first specifications for URLs, HTTP, and HTML.

See: www.w3.org/Help/

In order to be able to add links to web documents, Tim Berners-Lee needed a way to specify the protocol that was to be used to access the document. At the time FTP was the protocol used to access files on the internet. Once connected to a server the user would see a list of the files on the site, and would then decide which file to download. Hence in 1990 Berners-Lee developed URLs – a way of encoding domain names into a link that specified the access protocol to be used, **and** the file to be accessed, since web pages are simply files. Thus a URL could be used to make a request over the internet, which the receiving machine would understand, and the request did not always have to be the same request. Now a user could ask to FTP and to view files.

In general a URL has the following components:

<scheme> :// <host-name> / <path-name>

Where **<scheme>** tells the client application which protocol to use, **<hostname>** identifies the domain name or IP address of the server and **<path-name>** identifies a file on the server. Website URLs will generally have

<http> as the scheme, specifying the Hypertext Transfer Protocol. Other schemes include <file> which tells the client that the document is on the local machine, and <ftp> specifying the File Transfer Protocol. Some URLs may include optional data; for example, specifying a port number when the protocol associated with the scheme is not using its well-known port.

5.5.8 How a web browser works

When you select a web document to be loaded, either by entering its URL in the location field of your browser, typing in an IP address or by clicking on a hyperlink, the browser will typically carry out the following actions:

1. **Determine the URL requested and extract the domain name.** If an IP address has been used directly, the browser can go straight to step 3.
2. **Use a name server to get the IP address from the domain name.** The domain name will be resolved to an IP address by consulting a name server, either directly or by delegating the request to servers on other machines. Your computer will usually know the addresses corresponding to names on the local network and any remote machines it has recently been in contact with. In order to resolve other names the browser will contact a name server which will resolve the name if it has the information; if not, it will either pass on the request to another server or inform the client that it is unable to resolve the name.
3. **Make a TCP connection (normally to port 80) at the IP address and send a request for web page once the server has accepted the connection.** The request will include HTTP headers including a GET request for the file and information identifying the **user agent**. The server's response will start with headers indicating the version of HTTP in use, the time and date of the transaction, and headers indicating the data format of the document.
4. **The server sends the file, and both sides maintain the connection for subsequent requests, until either the client or the server informs the other it is ready to close the connection.** The client displays the document received.

In the first version of HTTP, the client would open a TCP connection and send a GET request. The server would honour the request (or send an error message in HTML format) and then release the TCP connection. RFC 2068 recommended **persistent connections** as the default setting, meaning that the connection between the client and the server would be maintained until one of them signalled the intention to close it. The RFC noted:

Prior to persistent connections, a separate TCP connection was established to fetch each URL, increasing the load on HTTP servers and causing congestion on the Internet.

See: www.ietf.org/rfc/rfc2068.txt

The above covers the typical course of events, when the browser fetches and displays a web page from a remote site. It is also possible that if the user has viewed the requested page recently, a copy may be retrieved from the local cache, saving network traffic and improving response time.

The server may carry out other functions not listed above, such as performing access control in the case of secure servers. For example, if a web interface is used for internet banking, the server will need to both verify the user's identity and check that they are authorised to access the particular resources they request.

5.5.9 Stateless connections and cookies

HTTP is a stateless protocol: there is no need for the client or server to keep track of information about each other, each HTTP request is intended to be independent of earlier requests, or previous sessions, from the same client. Even though the standard requires developers to implement persistent connection such that the TCP connection is kept open until the client or server requests that the connection be closed, still each request (and its response) is independent of all other request/response pairs.

Web servers do sometimes need to identify a client for the duration of a session; for example, on a shopping site, where a customer is adding items to a virtual shopping bag. Keeping track of the client's IP address is not enough: this identifies the computer rather than the individual, plus some IP addresses are dynamically generated and can change. To coordinate a session with a particular client, servers may request the browser to store a small data file known as a 'cookie' on the user's hard disk, which can serve to identify users uniquely. Cookies have various uses; for example, validating subsequent requests from users who have logged in to a web server using a username and password, keeping track of items in a shopping bag or remembering your time zone and location. Some cookies are saved on your local machine and read the next time you visit the originating web server; others are transient, existing during a particular session and deleted at its end.

There are also 'third-party cookies'; these are cookies that are not generated by the web server that the client is visiting, but are, for example, used by advertisers on websites to track viewing and shopping habits in order to personally target advertisements. Since May 2011, under the EU ePrivacy Directive (https://edps.europa.eu/node/3100#e-privacy_directive2009-136-ec), new visitors to EU websites must be given information about the cookies that the web server will be placing on their machine, the purpose for them and the visitors must explicitly agree. Cookies that are needed for you to successfully use the website, such as those that help you to stay logged in, or that remember items in your shopping bag, are exempt from the notification and consent requirement. All cookies, whether first or third party, that track a user's online habits, must be notified to the user. An EU Directive must be enacted into law in EU member countries; the ePrivacy directive has been law in the UK since 2012. See Chapter 6 for more details of the law relating to data protection and privacy.

5.6 An historical note: Fault tolerance, XHTML and the Semantic Web

HTML is fault tolerant, meaning that a browser will display as much information as it can, by interpreting as much of the code as it can, and ignoring the rest. This makes life easier for less skilled web authors and means that HTML is forward and backward compatible. Forward compatible means that newer HTML code can be displayed by browsers using an older version of HTML. Backward compatible means that browsers using the current version of HTML can read previous versions of HTML.

At the end of the 1990s, HTML was thought by the W3C to have a number of disadvantages:

1. Browsers had to be designed to compensate for authors' mistakes and omissions; making them bigger and as a consequence using more memory and processing power. As W3Schools wrote when commenting on XHTML: 'Today's market consists of different browser technologies. Some browsers run on computers, and some browsers run on mobile phones or

other small devices. Smaller devices often lack the resources or power to interpret “bad” markup. XML is a markup language where documents must be marked up correctly (be “well-formed”). See: www.w3schools.com/html/html_xhtml.asp

2. Developers had introduced proprietary tags which were only recognised by their browsers, meaning that: (1) developers could not always be sure if their pages would be seen the way they wanted them to be; and (2) at one point there was a fear that the web would divide into Netscape and Internet Explorer pages.
3. Fault tolerance makes it harder to debug faulty coding.
4. Web pages are read not just by people but also by software programs such as search engines and ecommerce agents; these programs may struggle to extract data from badly written HTML.

Hence, after HTML 4.01 was released in 1999, the W3C put its development efforts into XHTML, which is HTML rewritten as XML, a restricted form of SGML. XHTML became the official standard in 2000, and was updated in 2002. XHTML 1.0, the first release, was HTML 4.01 but with strict syntax; documents had to be ‘well-formed’ or marked up correctly. XHTML did not have HTML’s fault tolerance; for example, XHTML was case sensitive, while HTML was not. Items that were optional or that could be omitted in HTML, had to be included in an XHTML document.

XHTML addressed the above four disadvantages of HTML:

1. A stricter language needed less memory and processing power to interpret, hence browsers could be written that would be more suitable for smaller devices. In addition, XHTML supported the separation of presentation from mark-up, which made reformatting web pages for smaller devices easier.
2. XHTML allowed for new modules to be easily developed and shared. Modularization was a way of viewing XHTML as a collection of modules providing different functions, making it easier to extend XHTML to provide new functionality. The W3C developed XHTML validators that would check XHTML documents and reject unrecognised tags, see: <https://validator.w3.org/>.
3. Stricter rules made debugging easier.
4. Programs reading web pages would be able to extract information more efficiently if pages were formatted according to strict rules of syntax and with common standards for annotating data.

The stricter standards of XHTML were also perhaps informed by Berners-Lee’s interest in the **Semantic Web**, something that he first publicly discussed in 1994. The Semantic Web is the idea that web pages should be entirely machine-readable, described by Berners-Lee in his book, *Weaving the Web*, as follows:

I have a dream for the Web ... and it has two parts.

In the first part, the Web becomes a much more powerful means for collaboration between people. [...]

In the second part of the dream, collaborations extend to computers. Machines become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A “Semantic Web,” which should make this possible has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be

handled by machines talking to machines, leaving humans to provide the inspiration and intuition. [...]

Once the two-part dream is reached, the Web will be a place where the whim of a human being and the reasoning of a machine coexist in an ideal powerful mixture.

XHTML has much greater interoperability than HTML, allowing a document to be viewed in a browser but also in other devices, making it more possible for machine to talk to machine.

5.6.1 WHATWG, HTML5 and the role of W3C

With XHTML 2.0, first proposed in 2002, the W3C proposed ditching backwards compatibility with previous versions of XHTML and with HTML 4. In addition, the W3C had decided to end development of HTML. This caused some controversy, and even a rebellion with industry representatives forming the Web Hypertext Application Technology Working Group (WHATWG) in 2004 with the aim of continuing the development of HTML abandoned by the W3C. Ian Hickson, who worked for the Opera software company, and represented Opera on various W3C working groups, argued strongly that HTML should be the standard used to address the need for web applications and the desires of developers. Concerned with the direction the W3C was taking, he founded the WHATWG with representatives of Apple and Mozilla. This meant that the browser industry and other interested parties were developing new HTML standards, and testing them in their software.

In 2006 Sir Tim Berners-Lee (he was knighted in 2004) published a blog post admitting that the attempt to move the web into a wonderful new XML future, had failed. In 2007 the W3C formed a new working group to continue HTML development, led by Ian Hickson as editor. Effectively, the W3C had decided to use the standards developed and tested by the WHATWG group, as the basis for a new iteration of HTML, HTML5.

HTML5 was in use from 2008, and published by the W3C as a 'candidate recommendation' in 2014. The current position is that the W3C is responsible for the **in use** standards, while the WHATWG develops new standards, although there does seem to be some overlap.

5.7 HTML5

HTML5 is really a triumvirate, incorporating CSS and with strong support for JavaScript.

HTML:	for the structure of a web page
CSS:	for the look, or presentation, of a web page
JavaScript:	for the interactive and dynamic elements in a web page.

Casad (2017, **Hour 17**, heading *HTML5*):

[HTML5] features simply reflect a further evolution of the web environment, and incorporate functionality once provided by add-ons and extensions directly into the HTML.

[...]

In many ways, HTML5 marks the further evolution of HTML to something more like a development environment for web applications. Developers have long used HTML and its surrounding technologies to build web-based client/server

applications, but much of the functionality required additional components and third party extensions. HTML5 builds more of the functionality directly into the HTML.

If you visit the Google home page, and view the page source code, you will see something that looks terrifyingly complex, but at its heart HTML is still a simple mark up language. The W3C has developed a validator for HTML5, see: <https://validator.w3.org/nu/>. This can be a great help to developers in making sure that the pages display as they expect. HTML5 is fault tolerant, as are all earlier versions of HTML, hence browsers may not display text exactly as the developer wishes it to be seen. Validation ensures that the code is correct, and any presentation issues can be addressed by the developer knowing that the problem is not in the code.

5.7.1 Writing HTML5 code

The following is HTML5 code.

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title></title>
5.   </head>
6.   <body>
7.   </body>
8. </html>
```

If you were to drop this into the text area of the HTML validator on the W3C site, and press the 'check' button, you would see the following messages:

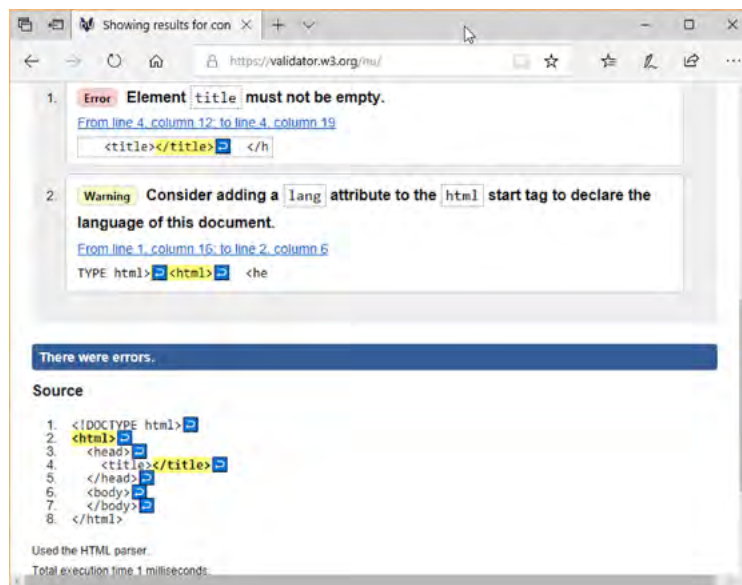


Figure 5.1: HTML validator message.

The validator suggests telling the browser what language you are using, which is optional, and reminds the user that a page must have a title. See: <https://validator.w3.org/nu/#textarea>

Line 1 of the above HTML code tells the validator which version of HTML to use. The DocType declaration is different for different versions of HTML, the one above is for HTML5, and is simple and easy to remember, unlike say those of HTML 4.01. If you are interested you can see a list of DocType declarations here: www.w3.org/QA/2002/04/valid-dtd-list.html The DocType must always

be the first thing in an HTML document; it refers the browser to a list of valid HTML elements for the version of HTML you are using.

The `DocType` is not an HTML tag, although tags are enclosed in '`<`' and '`>`'. Most tags must be closed, as well as opened. The `<html>` tag encloses the entire document and is closed with the same tag repeated but starting with a forward slash: `</html>`. Repeating the tag but with a forward slash is how all tags are closed. Indentation in the above HTML code is done for readability; it is not necessary.

Line 3 has the `<head>` tag, which is where to put the page title. The page does not have a title, which the validator has flagged as an error. The title is what is displayed on the browser's tag, and is used by search engines to index, and by browsers to bookmark pages. Once the head tag is closed on line 5, the body tag signifies the start of the area where the developer can add text, images and interactivity. Everything above the body tag is information for browsers and search engines and is not displayed on the page.

Once a title and the language is added to the page the validator finds no further errors. The language is added to the html tag with `lang = "en"`. Naturally other languages are also possible. `lang = "en"` is an example of an attribute. An attribute gives extra information about an element. Adding a language to the html tag means that the browser should expect the default language of the page to be the language specified, in this case English. I have also added the character encoding, with the meta tag and the `charset="UTF-8"` attribute below. The meta tag does not have separate opening and closing tags, it is known as an empty tag because it has no content, e.g. text, to display. The meta tag gives metadata about the page, see here for more information about the tag: https://www.w3schools.com/tags/tag_meta.asp

Most developers will encode their pages in UTF-8, which has become the standard encoding for the web, because it includes so many characters that ASCII does not include. In fact, UTF-8 is the default character set for HTML5. ASCII was the character encoding at the start of the word wide web. ASCII is based on the English language alphabet, decimal numbers and some commonly used symbols. Anything you can find printed on the keys of a keyboard for an English language device is likely to be an ASCII character. You can find more information about charsets here: www.w3schools.com/html/html_charset.asp

Adding the language, a title and the charset encoding gives the following:

```
1.    <!DOCTYPE html>
2.    <html lang = "en">
3.        <head>
4.            <meta charset="UTF-8" />
5.            <title>A Simple Web Page</title>
6.        </head>
7.        <body>
8.        </body>
9.    </html>
```

Displaying the page, I see this:

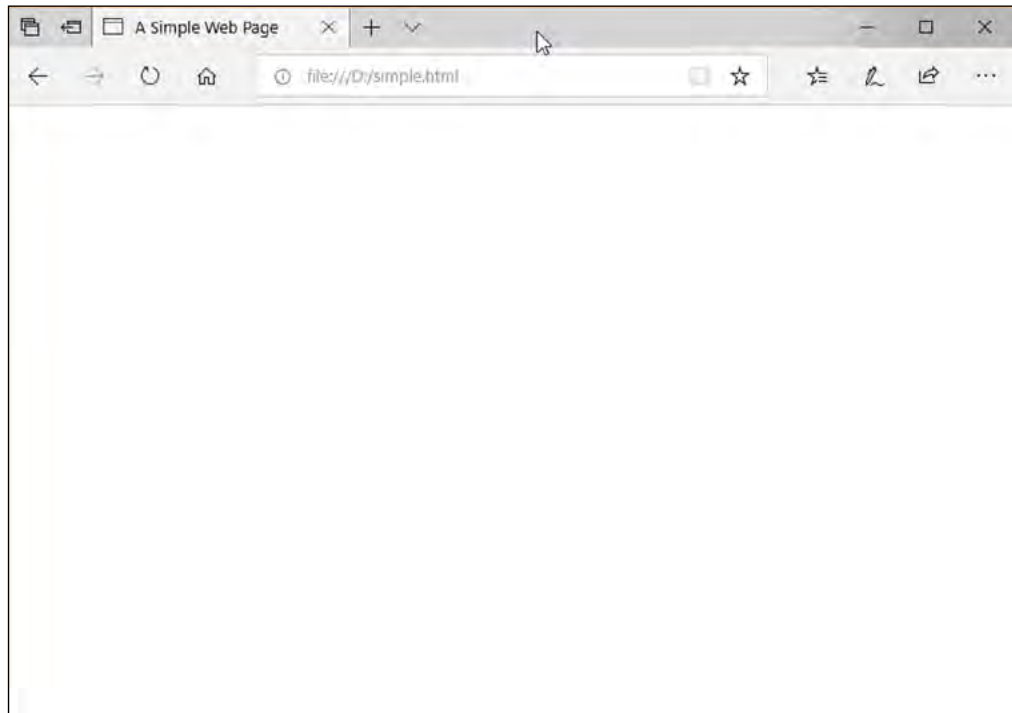


Figure 5.2: A simple web page.

As you can see, the title is on the browser's tab, but nothing is displayed on the page, so the next step is to add some text, as follows:

```
1. <!DOCTYPE html>
2. <html lang="en">
3.     <head>
4.         <meta charset="UTF-8" />
5.         <title>A Simple Web Page</title>
6.     </head>
7.     <body>
8.         Here is a silly joke.
9.
10.        How many surrealists does it take to change a
11.        lightbulb?
12.
13.        Fish!
14.     </body>
15. </html>
```

Saving my changes and refreshing my browser, I now see this:

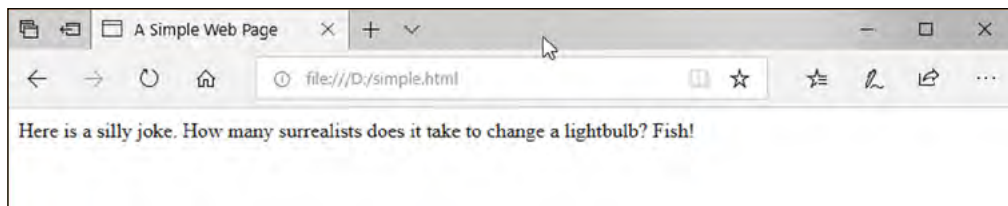


Figure 5.3: A simple web page with text.

As you can see, the spaces and line breaks that I included with my text have been ignored, with all text collapsed onto one line. If I want to structure my text, then I must use tags to do so. In the following I am going to use the

paragraph tag, `<p>`, to structure my text.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>A Simple Web Page</title>
  </head>
  <body>
    <p>How many surrealists does it take to change a lightbulb?</p>
    <p>Fish!</p>
  </body>
</html>
```

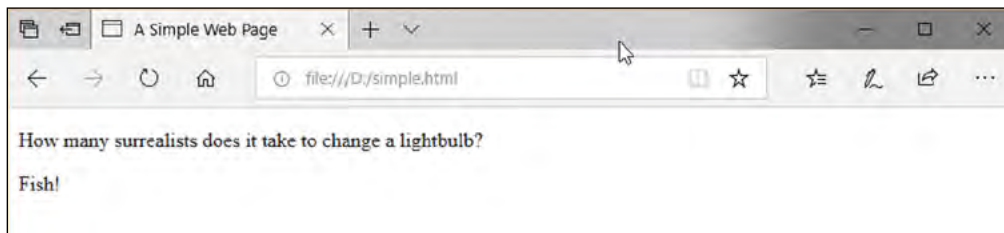


Figure 5.4: A simple web page with text and line breaks.

Now, I'm going to add some headings. HTML has six heading levels, `<h1>` to `<h6>`, but most developers stick with the first three heading levels. I can also structure text as a list, using `` for an ordered (numbered) list and `` for an unordered (or bullet-pointed) list. In both types of list `` specifies a list item. My final page, which I saved in a file called *jokesA1.html*, looks like this:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Types of jokes</title>
    <!-- Page title reflects purpose of page -->
  </head>

  <body>
    <h1>Some jokes</h1>
    <ol>
      <li>Lightbulb jokes</li>
      <li>Knock knock jokes</li>
      <li>Puns</li>
    </ol>
    <h2>Lightbulb joke</h2>
    <p>How many surrealists does it take to change a lightbulb?
    Fish!</p>
```



```

<h2>Knock knock joke</h2>
  <p>Knock, knock.</p>
  <p>Who's there?</p>
  <p>Deja</p>
  <p>Deja who?</p>
  <p>Knock, knock.</p>
<h2>Pun</h2>
  <p>Why don't programmers like nature? It has too many
  bugs.</p>
<h1>Can a machine be funny?</h1>
  <p>Check out this link: <a href= "https://www.theregister.
  co.uk/2018/06/01/japanese_neural_network_jokes_comedy/">Artificial
  jokes aren't funny</a></p>
</body>
</html>

```

I can also add comments to my HTML files with:

```
<!-- My comment here -->
```

Comments are notes for the developer's benefit, and will be ignored by browsers. See: www.w3schools.com/html/html_comments.asp

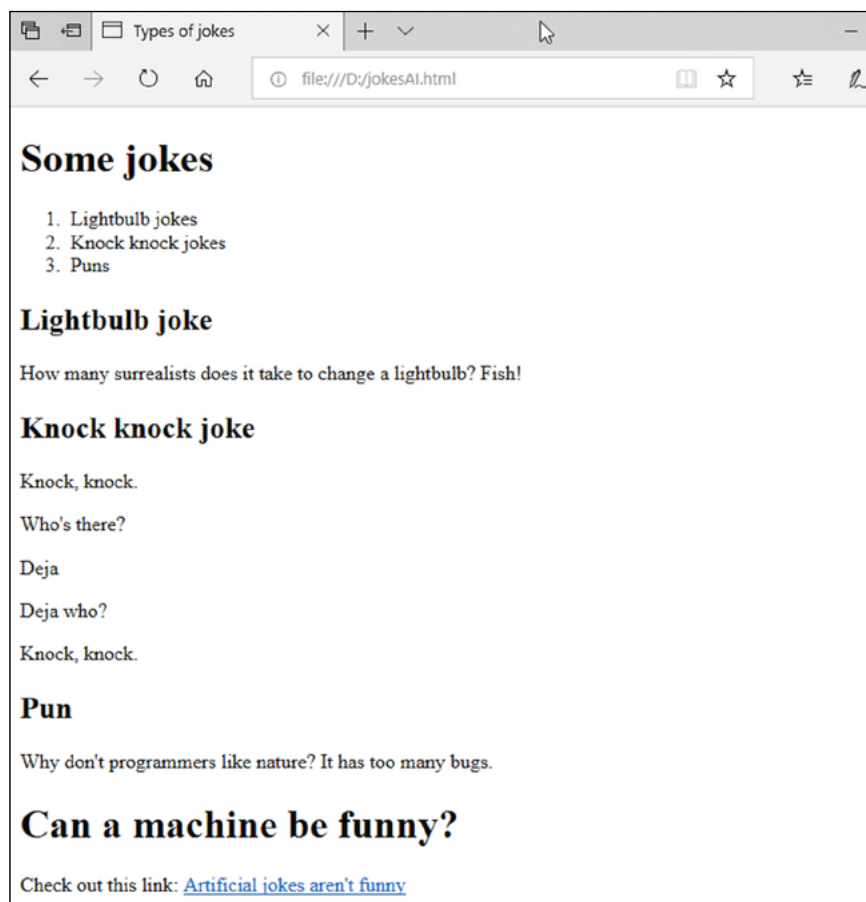


Figure 5.5: Web page with headings, an ordered list and a hyperlink.

So my web page now has a top-level heading, 'Some jokes', followed by a numbered list. Then three second level headings each with some text structured with the paragraph tag, and finally another top level heading with a link. The syntax for adding a link is: `link text`

where `url` should be replaced with your link, and `link text` with the text that the viewer will see (in the above 'Artificial jokes aren't funny').

Closing some tags is optional, and HTML elements will display correctly without it. This is not true of the heading tags, but it is true of the paragraph, `<p>`, tag. The W3C recommends always closing tags as you may see unexpected results with some browsers.

5.7.2 HTML5 template

The following HTML code is a template for a HTML5 page, written in English with the UTF-8 charset:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>This text will be put in the tab, is the
    name for the browser bookmark and used
    to index your page by search engines</title>
</head>
<body>
</body>
</html>
```

Write your web pages as simple text files, and save them with the *html* extension. Note that word processors may put unseen formatting and printing codes into files, and these codes can interfere with the presentation of your page, so it is best to use a text editor such as TextPad for writing your HTML files.

Activity 5.1

1. In the HTML5 template given above, identify the code that tells the web browser that the file is written in HTML5.
 2. In the HTML5 template, identify where the text that will be seen by the viewer begins and ends.
 3. In HTML what is the purpose of the HTML code above the `<body>` tag?
 4. Consider the links given in Section 2.8.3 *Links to other standards and protocols*. Write a simple web page titled 'List of useful links' and in it give an unordered list of the links in Section 2.8.3. Make sure to use the correct DocType for HTML5. Give your list a heading, and give each link some sensible descriptive text for the viewer to see. Test your page by loading it in a browser, this means copying and pasting (for example, from Windows File Explorer), or typing the path to the HTML file on your machine into the browser's address bar.
-

5.7.3 HTML5 fault tolerance and future proofing

Well-formed HTML code is code that is close to the XHTML standards that the W3C at one point wanted all developers to follow. HTML5 is fault tolerant and compatible with previous versions, hence developers do not need to write 'well-formed' HTML code, as browsers can and do interpret HTML5 code that is not well-formed. However, W3Schools recommends writing well-formed HTML5 code, in order to be 'smart and future proof' because:

A consistent use of style makes it easier for others to understand your HTML.

In the future, programs like XML readers may want to read your HTML.

See: www.w3schools.com/html/html5_syntax.asp

Writing well-formed HTML code means keeping to the following rules:

- Always declare the document type as the first line in your document. This can be upper or lower case, `<!DOCTYPE html>` or `<!doctype html>`.
- Use lower case tag names, for example `<table>` not `<TABLE>`. HTML5 is not case sensitive, and allows mixing of upper and lower case letters in tags and attributes.
- Use lower case attribute names, for example `<html lang = "en">` **not** `<html lang = "EN">`
- Always quote attribute values. HTML5 allows attribute values to be set without quotes. While `<html lang = en>` would work, it should not be attempted.
- Close all elements. For example, the paragraph tag, `<p>`, does not need to be closed, but close it anyway.
- Close all empty elements, that is elements with no content. For example: `<h2>This is content</h2>` The meta tag has no content, which is not to say that it does not have a purpose. HTML5 does not require empty elements to be closed, as in `<meta charset="UTF-8">`, but close them anyway by adding the forward slash, for example `<meta charset="UTF-8" />`.
- When adding an image (done by providing a link to an image file) always use the alt text. This is the text that describes the image and will be shown to viewers if for some reason the image file does not load. It is also the text that will be read to visually impaired users who are running assistive software. For example: ``

Here the image file is **birthday.jpg** and the alt text that describes the file is *A photo of a birthday cake with lit candles*.

- The `<html>` tag can be omitted. However, this tag is the preferred place to specify the page language, which is important for accessibility software and search engines.
- The `<body>` tag can be omitted. The browser will infer where the text to display starts; however, this is not recommended as it is not XML compliant, and can confuse some older browsers.
- The `<head>` tag can be omitted; however, this would mean including the `<body>` tag, so that the browser can infer that anything before the body is the head.

See: www.w3schools.com/html/html5_syntax.asp

5.7.4 HTML5 and progressive enhancement

Casad (2017, **Hour 17**, heading *HTML5*), notes how HTML5 supports the web on smaller devices:

One who looks at the HTML5 feature list will see that many of its traits reflect the new role of HTML as a tool for building the mobile revolution. The HTML5 standard includes many features designed to support web browsing from mobile devices.

Frames are deprecated in HTML5. Frames were used to divide a web page up into different areas, with each area able to have its own attributes. This was known as a frameset. Instead HTML5 has various different layouts available for developers to use, with the advantage that the width of each element can be specified as a percentage of the total width of the page, allowing web pages to be **responsive**; that is, to realign themselves on smaller devices. This means that one web page can be written and viewed on phones, tablets and desktop computers rather than the developer having to have three different websites, and to direct users to the correct page for their device.

Progressive enhancement is an approach to building web pages that allows browsers to display as much as they can, whether the browser is limited in its abilities by advanced age, or by being displayed on a smaller device. The idea is to first build the page with core HTML content that is accessible to all visitors, whatever device or browser they are using. After that add the design of the page with CSS, and any interactivity with JavaScript. Hence the developer builds the core of the page with simple features that are accessible to all, and then adds enhancements that may not be accessible for all users. Page layouts with percentage widths are an example of one of the many features of HTML5 that support progressive enhancement.

5.8 CSS and HTML5

Separating the structure of HTML documents from their presentation had been a goal of Berners-Lee from the first development of HTML, but in fact presentation was left to web browsers, which at first had very limited ways for developers to control the appearance of their pages. When HTML 3.2 introduced support for specifying fonts and colours, this solved one problem for developers, but introduced another one, in large websites adding fonts and colours to every page could take some time. Cascading Style Sheets (CSS) were developed by the W3C in order to remove formatting from HTML.

When tags like , and colour attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

See: www.w3schools.com/css/css_intro.asp

In HTML5 structural elements such as paragraphs, headings, lists and list entries are handled by tags, sometimes with attributes to give more information about the element. In previous versions of HTML attributes could give information about presentation, such as the colour of text, or the font to be used, or whether text was centered, and so on. Using CSS was a choice.

Many presentational attributes are deprecated in HTML5, and developers are now expected to use CSS to add formatting to their text.

5.8.1 Three ways to format a web page with CSS

Using CSS for formatting a web page can be done in three ways. For example, I have edited the *jokesAl.html* file and saved it as *jokesAlwCSS.html*, to demonstrate the three ways. In the file I have made the some additions as follows:

- **Inline styles:** adding a style to a single element directly, where allowed, using the style attribute. For example, I could have made the text for the first heading of the jokes page purple, as follows:

```
<h1 style="color:purple;">Some jokes</h1>
```

- **Document level:** adding style information within the head of the HTML file. In the jokes page I could have made each level two heading (<h2>) red by adding the bold section below to the head:

```
<head>
  <style>
    h2{color: red;}
  </style>
  <meta charset="UTF-8">
  <title> Types of jokes </title>
</head>
```

- **External style sheets:** I can put the styling that I want to apply to elements into a .css file, and add a link in the head of my HTML document as follows:

```
<head>
  <link rel="stylesheet" href="blueheadings.css">
  <meta charset="UTF-8">
  <title> Types of jokes </title>
</head>
```

In the same web page, I can do all three.

In the above example the CSS file, *jokesAlwCSS.html*, is in the same directory as the HTML file, hence the link to the CSS file is just the file name (*blueheadings.css*), effectively the relative path to the file when the file is in the same location as the HTML file. A relative path means that the browser looks for the file relative to the location of the page it is displaying. Particular syntax can be used to indicate if the file is to be found in a directory above or below the current one, and how far above or below in the directory structure. The file path can also be given as an absolute path, although this is not recommended. An absolute path includes the root directory and all sub directories and ends with the file name. See here for more information about file paths in HTML: www.w3schools.com/html/html_filepaths.asp

The *blueheadings.css* file contains just this text:

```
h1{
  color: blue;
}
```

The effect of the CSS file in any HTML file that it is linked to, will be to make all the top level headings blue, except for any top level headings that were formatted using document level or inline styling.

5.8.2 CSS precedence

The three different ways of adding presentation with CSS have precedence levels. Precedence is needed so that when the developer has used more than one CSS method to format a particular piece of text, the browser knows which format to follow.

- Highest level of precedence: inline styles – the browser will always implement them, if present.
- Document level styles have the middle level of precedence, they are ignored if an inline style is available, but take precedence over external style sheets.
- External style sheets have the lowest precedence level, meaning that formatting contained in an external style sheet is only applied if no inline or document level formatting is specified.

Precedence levels are designed so that developers can have very fine-grained control over the appearance of a website. All web pages can link to the same external style sheet in order to apply a house style to all pages. Document level styles can be used if a particular page needs a different look, and inline styles can be used just for particular elements. For example, perhaps the developer wishes to add emphasis to a particular piece of text, with colour or by making the font bigger. Because the inline style has the highest precedence, it is possible to use it to make small changes to web pages that in all other respects have the house style applied by linking to a CSS file.

5.8.3 Syntax for CSS

Within an **external CSS file** HTML structural elements are formatted by using the name of the element, followed by the desired attributes in curly brackets. Attributes are ended with a semi-colon, and more than one can be applied to an element. For example, the *blueheadings.css* file contains only:

```
h1{  
    color: blue;  
}
```

This means that the content of all `<h1>` headings will be in blue, unless the web page has inline or document level styles for top-level headings that take precedence. Suppose that I also wanted my top-level headings to be centred? I would add the attribute `text-align`, followed by a colon, and after the colon the style I want the attribute to take, followed by a semi-colon to indicate the end of the attribute and value pair. Changing my CSS file, and saving the result as *jokes.css*, the file would contain:

```
h1{  
    color: blue;  
    text-align: center;  
}
```

The above curly braces and their contents are known as a **declaration block**, and each attribute and style pair is known as a **declaration**. Hence my *jokes.css* file has one declaration block, and inside the block are two declarations.

In general, the element or elements selected for formatting in the CSS file is known as the **selector**. In the above example `h1` is the selector. A selector and its declaration block are together known as CSS rule-sets. Hence each CSS file consists of one or more CSS **rule-sets**. The *jokes.css* file contains one rule-set. The rule-set has the selector `h1`, and a declaration block that formats text as blue and centred.

Document level styles must be placed in the head section, and enclosed within style tags. Inside the style tag is one or more CSS rule-set, as described above. For example, in the *jokesAlwCSS.html* file, the following code in the head section, makes the second level headings red:

```
<style>
  h2{color: red;}
</style>
```

I have decided to make all second level headings red and centered (saved in the file *jokesAlwCSSv2.html*) with:

```
<style>
  h2{color: red;
    text-align: center;}
</style>
```

Centering the level two headings looked odd, so I decided to make all paragraphs centred, as well:

```
<style>
  h2{color: red;
    text-align: center;}
  p{text-align: center;}
</style>
```

Inline styles are applied to single elements using the style attribute, and enclosing the declaration in quotes, as the following example from the *jokesAlwCSS.html* file shows:

```
<h1 style="color:purple;">Some jokes</h1>
```

If more than one formatting is to be applied, then just as for document level and external style sheets, they are included inside the quotation marks. For example, perhaps in the *jokesAlwCSSv2.html* page I want the font used for the first top level heading to be Courier, so I add `font-family:courier;` to the style attribute:

```
<h1 style="color:purple; font-family:courier;">Some jokes</h1>
```

You can read more about CSS syntax here:

www.w3schools.com/css/css_syntax.asp

5.8.4 Fonts

Before HTML5 the font tag could be used to set the size, colour and type of font. Hence to change the colour of the first level one heading in the *jokesAl.html* file, I might have put:

```
<h1> <font color="purple">Some jokes</font></h1>
```

Use of the font tag is deprecated in HTML5. CSS should be used for all formatting, including font type, colour and size. For example, using inline CSS I might style the size, colour and font of my first level one heading in my *jokesAlwCSSv2.html* file with:

```
<h1 style=" font-size:60px; color:purple; font-
family:courier;">Some jokes</h1>
```

There is one issue with fonts, which is that the local machine may not have the necessary fonts to display a web page as the designer conceived it. This is why fonts can be specified as a list using the `font-family` property, such that the browser can choose the first font from the list that it can display. If the browser cannot find a font in the list then it will use the operating system's default font for display. In order to prevent this, the font list can be terminated

with the type of font required, for instance `monospace` means a font where each character has the same width, such as Courier.

For example, I might add the following rule-set to my *jokes.css* file:

```
h1{
    color: blue;
    font-family: Courier, "Lucida Console", monospace;
}
```

In the above rule-set, the second declaration says to make all text in the top level headings (`<h1>`) Courier, if that is not available then use Lucida Console (enclosed in quotes to make sure both words are treated as a single font name) and if that is not available then find a monospaced font and use that.

Other options are `serif`, `sans-serif`, `cursive` and `fantasy`; these options are known as `generic-family` types. Names for particular fonts, such as Courier, Calibri, etc. are known as `family-name` types. You can read more about fonts and CSS here: www.w3schools.com/css/css_font.asp and you can read about the CSS `font-family` property here: www.w3schools.com/cssref/pr_font_font-family.asp

5.8.5 Colours

When adding colour formatting in the examples given above we have used colour names, such as **purple**. HTML supports 140 colour names; a complete list can be found here: www.w3schools.com/colors/colors_names.asp

Many developers, wanting access to the full range of colours available, will use hexadecimal values to specify the desired colour rather than names. There are other ways, but hexadecimal and colour names are the only ones this guide will consider. As we know, a colour is made by specifying the amount of red, green and blue it contains in that order. Colour values are specified with a number, and the higher the number the more of that particular colour to include. For example, xyz means give colour values as follows: red: x; green: y; and blue: z. Each colour value can be between 0 and 255 inclusive ($2^8 - 1$). With the inclusion of 0, we have 256 numbers. Hence we have $256 \times 256 \times 256$ colours, or more than 16 million.

The hex colour value then obviously consists of three distinct hexadecimal numbers. Each hexadecimal number in the triple has two digits, giving $16^2 - 1$ values for each, or 255 (256 numbers including zero). In HTML hex colour values are prefixed with '#'; for example, red would be #FF0000 (red: 255, green: 0 and blue: 0) and blue would be #0000FF. Black would be #000000, or just #000 since when a hex colour value consists of 3 repeated pairs, it is possible to use just one member of the pair to represent both numbers. This cannot be done for hex colour values with just one or two repeated pairs, such as #15FF0E, which must remain as #15FF0E.

What colour is #15FF0E? 15FF0E means there are 15_{16} for red, FF_{16} for green and $0E_{16}$ (or just E_{16}) for blue. This means we have red: 21; green: 255; and blue 14; giving a very bright green colour.

Activity 5.2

Using the *jokes.css* file, experiment with adding a background colour to the page described by the *jokesAlwCSSv2.html* file, in hex colour notation. You can do this by adding a new rule-set to the file. You will need to choose the selector, and find the attribute name.

You can find the *jokes.css* file and the *jokesAlwCSSv2.html* file on the VLE.

5.8.6 Images

Images are added to your web pages by using the `` tag. The image tag has two important attributes: `src`, which tells the browser where to find the image file and `alt`, which is text that describes the image if it should fail to load for some reason. The `alt` text is also read to viewers who are using assistive technology, and should always be completed with some sensible descriptive text.

Hence in order to add an image, at its most basic, you should have the following:

```

```

For example:

```

```

In the above example, the browser will expect to find the image in the same directory as the web page, since the name of the file alone implies both files are in the same folder; see: www.w3schools.com/html/html_filepaths.asp for more about file paths in HTML. If the image file is uploaded to a different folder, as is usual, the relative (preferred) or absolute file path should be given.

Images can also be given specific dimensions for their width and height. This can be done with the `width` and `height` attributes or with the `style` attribute. It is recommended that it is always done with the `style` attribute. Images do not have to be given dimensions, but if they are then the browser knows how much space to reserve for them, and will load more smoothly, as it can work out exactly where to place the image and any accompanying text.

Since digital images are measured in pixels, it is usual to give the image size in pixels. This can be done by finding the size of the original image and using those dimensions, although if the original image is very large you may wish to scale it with an image editor first. In Windows you can find the dimensions of an image by right clicking while your mouse hovers over it in Windows File Explorer. From the menu that appears choose **Properties**, and within that, choose the **Details** tab. There you will find the exact dimensions of the image in pixels. However, the space given to an image in a browser does not have to reflect its actual size, it can be smaller or larger. Normally you would not want image sizes larger than a few hundred pixels in width and height. Taking, choosing, editing and sizing images is beyond the scope of this course.

Whatever size you decide on for your image, you should set the size as follows:

```

with X and Y replaced by pixel values.
```

For example:

```

```

See: www.w3schools.com/html/html_images.asp for more about images.

The CSS background-image property allows you to set an image as the background to your entire web page, see here for more details:

www.w3schools.com/cssref/pr_background-image.asp

Activity 5.3

Why should the width and height of an image be specified by the `style` attribute rather than with the `width` and `height` attributes?

5.8.7 Tables in HTML5 with CSS

Tables in HTML5 come with powerful new properties: tables can be divided into header, body and footer sections with the tags `<thead>`, `<tbody>` and `<tfoot>`. Some browsers can recognise these sections and treat them differently; for example, by allowing the viewer to scroll through the body section of large tables, while keeping the header and footer sections constant at the top and bottom of the page. Alternatively, when a table is viewed in successive pages due to its large number of entries, the browser can insert the header and footer sections at the top and bottom of each page. The following is a simple template for a HTML5 page containing a table with header, body and footer sections (*HTML5-table.html*):

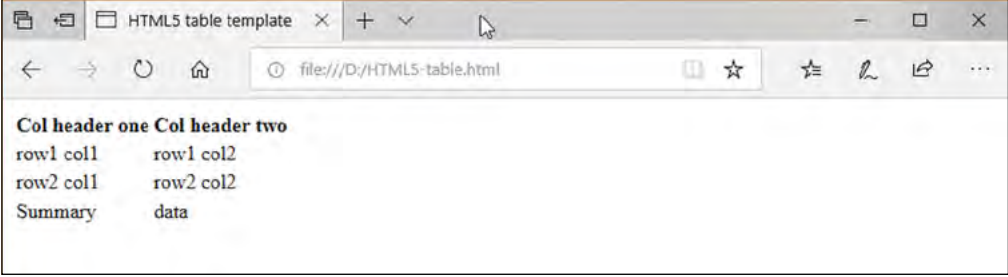
```
<!DOCTYPE html>
<html lang = "en">
  <head>
    <meta charset="UTF-8" />
    <title>HTML5 table template</title>
  </head>

  <table>
    <thead> <!-- table header start-->
      <tr>
        <th>Col header one</th>
        <th>Col header two</th>
      </tr>
    </thead> <!-- table header end-->
    <tbody> <!-- table body start -->
      <tr>
        <td>row1 col1</td>
        <td>row1 col2</td>
      </tr>
      <tr>
        <td>row2 col1</td>
        <td>row2 col2</td>
      </tr>
    </tbody> <!-- table body end-->
    <tfoot> <!-- table footer start -->
      <tr>
        <td>Summary</td>
        <td>data</td>
      </tr>
    </tfoot> <!-- table footer end -->
  </table>
```

The `<thead>` section contains one row (`<tr>`) and the row has two cells, given with the `<th>` tag, which means the cell is a header cell, designed to

label a row or column. The body section has two rows, and each row has two data cells, described with the `<td>` tag. The footer section (`<tfoot>`) has one row, with two data cells.

If loaded into a web browser, the table should look something like this:



Col header one	Col header two
row1 col1	row1 col2
row2 col1	row2 col2
Summary	data

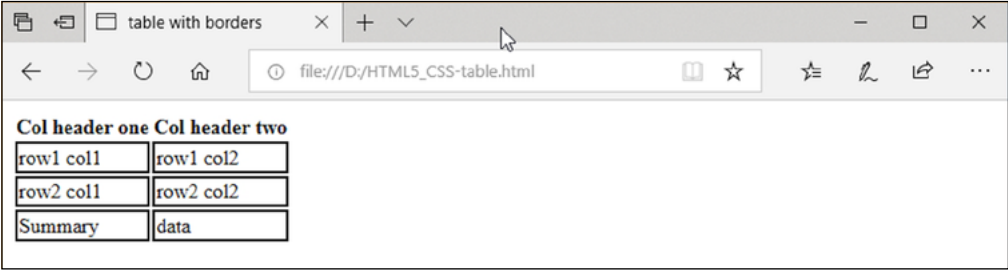
Figure 5.6: HTML5 table template.

Browsers will display tables only as wide as necessary. In Figure 5.6, the header row decides how wide the table needs to be. While the table has borders, these cannot be seen as they do not appear by default, but have to be added with CSS. Hence I will link an external CSS file to the page, and add borders to the data cells, saving the revised file as *HTML_CSS-table.html*.

The CSS file (tableformatting.css) contains the rule-set:

```
td{
    border: 2px solid #000;
}
```

The table now looks like this:



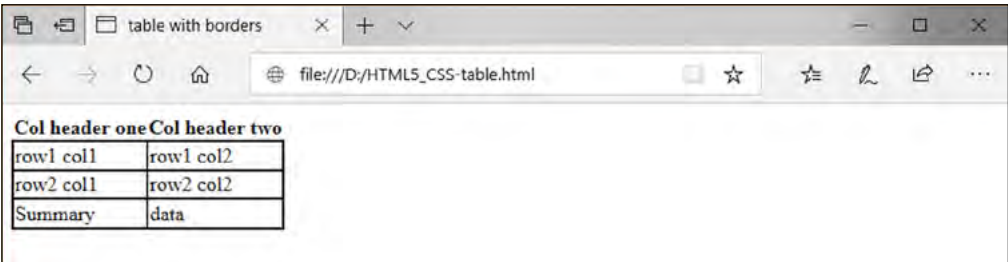
Col header one	Col header two
row1 col1	row1 col2
row2 col1	row2 col2
Summary	data

Figure 5.7: Table with borders.

I can add the `border-collapse` property to the CSS file in order to remove the space between borders (the default setting) as follows:

```
table {
    border-collapse: collapse;
}
td{
    border: 2px solid #000;
}
```

So that now I will see:



Col header one	Col header two
row1 col1	row1 col2
row2 col1	row2 col2
Summary	data

Figure 5.8: Table with space between borders removed.

5.8.8 Advantages of CSS

The advantages of using CSS are as follows.

- The tasks of providing content for a web document and controlling its appearance can be split between different individuals or departments. This makes for easier management of a house style, as the liaison for web page styling will be a single person or department. It also simplifies things for departments and individuals who do not want or need to concern themselves with the detail of design, but wish to concentrate on content.
- If a similar appearance is required for a number of pages, the work of specifying or changing the style only has to be done once, rather than over and over again in each file.
- If the house style should change, then only the CSS file needs to be edited, not every web page.
- By using external style sheets, it is easy for organisations to maintain a consistent 'house style' among their web pages, as co-workers can be instructed to link to a designated CSS file.
- CSS offers very fine-grained control over the layout of a web page.

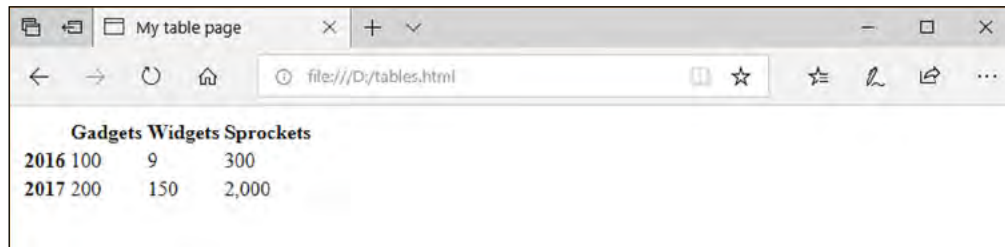
Activity 5.4

```
<!DOCTYPE html>
<html lang = "en">
  <head>
    <meta charset="UTF-8"/>
    <title>My table page</title>
  </head>
  <body>
    <table>
      <tr>
        <th></th>
        <th>Gadgets</th>
        <th>Widgets</th>
        <th>Sprockets</th>
      </tr>
      <tr>
        <th>2016</th>
        <td>100</td>
        <td>9</td>
        <td>300</td>
      </tr>
      <tr>
        <th>2017</th>
        <td>200</td>
        <td>150</td>
        <td>2,000</td>
      </tr>
    </table>
  </body>
</html>
```

Consider the above HTML5 code that lays out a simple table. The code is saved in the *tables.html* file.

Download the *tables.html* file from the VLE. Make sure to use a text editor such as TextPad when editing your file so that there are no hidden word processing codes in your document that may cause problems for your browser.

Load the document in a web browser, by typing its path into the browser's address bar, and you should see this:



	Gadgets	Widgets	Sprockets
2016	100	9	300
2017	200	150	2,000

Figure 5.9: *tables.html* in a web browser.

Note that the years 2016 and 2017 are not contained in data cells, but are in `<th>` cells; that is, they are header cells, designed to be used to label rows and columns.

- (a) Save the file as *tables4.html*, and make the following changes to the HTML code:
1. Add a heading to the page, 'A close look at gizmos'.
 2. Divide the table up into `<thead>`, `<tbody>` sections.
 3. Add a `<tfoot>` section. Label this section 'TOTAL' and put the sum of each column into a data cell. Figures for each data cell respectively are 300, 159 and 2,300.

When you have made the above changes, you should see something similar to the following when loading your *tables4.html* file into a web browser:

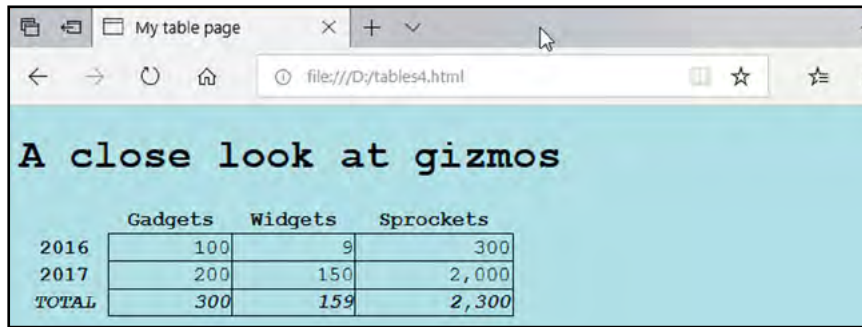


	Gadgets	Widgets	Sprockets
2016	100	9	300
2017	200	150	2,000
TOTAL	300	159	2,300

Figure 5.10: *tables4.html* in a web browser.

- (b) Write a CSS file called *activity4.css* and link the *tables4.html* file to it. Write rule-sets in the CSS file to do the following:
1. Set the font for the page to 'Courier New'.
 2. Set the background colour of the page to blue.
 3. Make the table footer contents bold and italic.
 4. Put a border around each data cell.
 5. Align text in data cells to the right.
 6. Set the width of the table to be 50%.

Once your CSS file is complete, your web page should look something like this:



The screenshot shows a web browser window with the title 'My table page' and the address bar displaying 'file:///D:/tables4.html'. The page content has a light blue background and features the heading 'A close look at gizmos' in a bold, black, monospace font. Below the heading is a table with three columns: 'Gadgets', 'Widgets', and 'Sprockets'. The table contains three rows of data: '2016', '2017', and 'TOTAL'. The values in the table are: Gadgets (100, 200, 300), Widgets (9, 150, 159), and Sprockets (300, 2,000, 2,300).

	Gadgets	Widgets	Sprockets
2016	100	9	300
2017	200	150	2,000
TOTAL	300	159	2,300

Figure 5.11: Final *tables4.html*.

You may find the following links helpful:

- www.w3schools.com/tags/tag_tfoot.asp – example of a table with a header, footer and body sections.
- www.w3schools.com/html/html_styles.asp – background colour
- www.w3schools.com/css/css_font.asp – using CSS to format fonts
- www.w3schools.com/css/css_table.asp – border-collapse, width of a table, aligning text.

5.9 HTML5 and JavaScript

Historically, developers used plug-ins for interactivity that HTML did not provide, although plug-ins often brought with them security issues (see Chapter 7). HTML5 reduces the need for plug-ins in two ways. One is by allowing developers to add audio and video directly to their pages, known as ‘native multimedia’. Not all browsers support this, and the file formats have not currently been standardised, plus some developers may see it as an issue that source files may be available to anyone, since users can read the embedded links and navigate to the source file.

The second is the integration of JavaScript with HTML5. This is done through the **Canvas**, an area that JavaScript uses to draw graphics. Developers would include a Canvas on a web page with HTML code, for example:

```
<canvas id="topCanvas" width="150" height="120">
</canvas>
```

Developers use a scripting language to add graphics, usually JavaScript, but if writing code directly onto the Canvas (rather than importing a file) then they must use JavaScript. Graphics can be simple or interactive, and can include games. The developer specifies the size of the Canvas, with `width` and `height`, and can have more than one Canvas area on a page, using the `id` to distinguish them. In the example above the `id` is `topCanvas`.

The Canvas area with JavaScript can be used for such things as:

- Creating forms for the user to fill in, in order to capture data.
- Creating text boxes for the user to enter data, that is then manipulated and fed back to the user; for example, a calculator to convert a monthly interest rate into a yearly one.
- Creating games.
- Adding an area for the user to drag and drop files to upload.
- Writing complete web applications (using advanced features of JavaScript not covered here).

You are not required to know any of the details of how to implement JavaScript in web pages for this course, nor are you required to know how to code with JavaScript. However, if you are interested in learning more then you may find the following sources useful:

- The W3Schools Canvas tutorial is here: www.w3schools.com/graphics/canvas_intro.asp
- JQuery is a large library of JavaScript files, with a community of online users and good online documentation. There are other JavaScript libraries; however, W3Schools recommend *JQuery*, see: www.w3schools.com/jquery/default.asp

5.10 Dynamic web pages

Dynamic web pages are those that use scripting languages to interact with users. Applications generating dynamic web pages are classified as client-side or server-side according to where the script actually runs. Server-side scripts run on the web server, which then sends results to the client's machine. Client-side scripts run within the client's web browser; no interaction with a web server is required, although once the script has completed information gained (for example, the user might have been asked to input their address) may be returned to the server.

Typical e-commerce applications will employ scripting both on the client system, to validate input, and on the server system, to process the input.

5.10.1 Client-side scripting

This can be used to generate web pages or perform other actions on the user's machine under the control of the browser, in response to user actions. With HTML5 this is likely to be done with JavaScript, but other scripting languages are available. Some typical applications are as follows.

- Validating forms input: for example, checking that a user has entered obligatory fields such as their name and address, or that their credit card number has the requisite number of digits. This saves on time compared with transmitting invalid or incomplete data to the server.
- Performing simple calculations on the client machine rather than transmitting data to the server and waiting for a reply: for instance, giving mortgage estimates based on the user's salary, deposit available and postal code (zip code) of the property.
- 'Rollovers', which display text or images when the cursor moves over an item on the page. These can be used to display additional information about a particular field or link on the page.
- A server can save on transactions by downloading a script which contains the ingredients of a number of different pages, which can be displayed in turn as selected by the user, either by moving the cursor over a field or by clicking on it. From the user's point of view there appears to be no difference between this technique and fetching each page separately, apart from the speed.

5.10.2 Server-side scripting

The advantage of client-side technologies is that they can save on transaction time by executing functions on the client's machine that do not require interaction with the server. The user's waiting time is cut down, and the website owner can reallocate the CPU cycles which have been offloaded to the client, to run some other application.

A scripting language is a language that can tell other programs and apps to perform certain tasks. On a server, one very typical use for a scripting language is to act as a bridge between the back end database, and the front end web page. The scripting language will send the query to the database in a form the database understands, and return it as HTML that the server can send to the client machine for display. Typically PHP would be the server-side scripting language, which many authorities claim is used by 80 per cent of web servers.

PHP (Hypertext Preprocessor) is open source general-purpose scripting language that is free to download and use. PHP is supported by the W3C for integration with HTML5, and W3Schools has the following to say about it:

PHP is an amazing and popular language!

It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!

It is deep enough to run the largest social network (Facebook)!

It is also easy enough to be a beginner's first server side language!

What is a PHP file?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP codes are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

See: www.w3schools.com/php/php_intro.asp

Other popular languages used for scripting on the client side are Ruby and Python, both object-oriented scripting languages. Also used are C#, Scala and even JavaScript. Python (and JavaScript) tutorials are offered by W3Schools, see: www.w3schools.com/default.asp

Although you are expected to be aware of client and server-side scripting as part of the learning outcomes for this course, you are not required to be familiar with the details of specific applications.

5.11 Electronic mail and its protocols

Electronic mail is one of the oldest applications for the internet and still one of the most widely used. SMTP, the most common protocol for sending email, dates back to the early 1980s. An email client allows users to both compose and send messages and to retrieve and read email messages. The email client will most likely send messages using SMTP, and retrieve them using either POP3 or IMAP. Following the client/server model, the email client sends to, and receives email messages from, a server.

5.11.1 SMTP

SMTP (Simple Mail Transfer Protocol) is the most commonly used protocol for sending email. When the client sends an email message it will normally be received at the local SMTP mail server, which will either place the email into one of its user mailboxes for later retrieval, or forward the message to another server if it has been sent to a recipient in a different domain. In either case, server software stores the email message for retrieval by the recipient.

Casad notes two advantages of using a server for email: (1) the user does not have to be on the network in order for the server to accept mail for them; and (2) the recipient can retrieve their email from any device with an email client that can connect to the server (Casad, 2017, **Hour 20: Email**, in the section *Retrieving the Mail*).

An SMTP transmission is initiated by connecting to port 25 on the receiving server. Following that the client sends a sequence of commands to the server, followed by a series of header and data records that are destined for the recipient. It is important to distinguish between the functions of the commands and the header records.

5.11.2 SMTP commands

SMTP clients use a series of short, mostly four-letter, commands to communicate with SMTP servers. Some important commands are as follows.

- **HELO hostname** which is the sign-on command and identifies the originating system.
- **MAIL FROM: username** indicates the address of the sender (the individual who is sending the message).
- **RCPT-TO: <address>** is a list of addresses, each enclosed in angle brackets. The list may only contain one address.
- **DATA** indicates that what follows is data, including header records. The end-of-file is indicated by a line consisting only of a full stop (.).
- **QUIT** closes the session.

Note that all commands are sent as ASCII text strings making it easier for administrators to debug a trace of the session.

5.11.3 SMTP headers

Email headers are what you usually see at the beginning of a message, giving basic information about the message. Note that headers are mostly for the user's information and the header is not linked to SMTP commands. Email headers contain a great deal of information that is not routinely displayed to users. Email clients normally display only the header records of most interest, such as subject, date, who the message is from, who the message is to, what address to reply-to and what encryption has been used. Email clients have a facility to display the entire header, for example, if using Gmail, the user would need to access the command **Show original** from the hamburger menu to see the full header of a particular email.

Some examples of header record types are:

- **Subject:** as filled in by the sender
- **From:** email address of sender
- **To:** email address of recipient
- **Date:** time and date sent
- **Received:** domain name of each SMTP server that has processed this mail

5.11.4 SMTP and email spoofing

The original SMTP specification (RFC 821) does not require the **From:** and **To:** fields to contain the same addresses as were given in the **MAIL FROM** and **RCPT TO** commands. In general there is no link between SMTP commands and the header fields that allows method authentication. This makes it easy to construct 'spoof' emails that appear to come from someone other than the actual sender. In recent years there has been a massive increase in spoofed emails, often sent by spammers wanting to trick the recipient into opening the email because they believe it has come from a source in their address book. Typically spammers will use web crawlers to construct lists with millions of email addresses, knowing that only a few people will respond to their messages. The most recent RFC addressing this security issue, RFC 5321 from 2008, notes in Section 7 *Security Considerations*:

7.1. Mail Security and Spoofing

SMTP mail is inherently insecure in that it is feasible for even fairly casual users to negotiate directly with receiving and relaying SMTP servers and create messages that will trick a naive recipient into believing that they came from somewhere else. Constructing such a message so that the "spoofed" behavior cannot be detected by an expert is somewhat more difficult, but not sufficiently so as to be a deterrent to someone who is determined and knowledgeable. Consequently, as knowledge of Internet mail increases, so does the knowledge that SMTP mail inherently cannot be authenticated, or integrity checks provided, at the transport level. Real mail security lies only in end-to-end methods involving the message bodies, such as those that use digital signatures (see RFC 1847 [43] and, e.g., Pretty Good Privacy (PGP) in RFC 4880 [44] or Secure/Multipurpose Internet Mail Extensions (S/MIME) in RFC 3851 [45]).

Various protocol extensions and configuration options that provide authentication at the transport level (e.g., from an SMTP client to an SMTP server) improve somewhat on the traditional situation [but] in general, they only authenticate one server to another [...]

Efforts to make it more difficult for users to set envelope return path and header "From" fields to point to valid addresses other than their own are largely misguided: they frustrate legitimate applications in which mail is sent by one user on behalf of another, in which error (or normal) replies should be directed to a special address, or in which a single message is sent to multiple recipients on different hosts.

See: <https://tools.ietf.org/html/rfc5321>

ISPs now generally use a variety of methods to attempt to detect and reject spoofed and spam emails; these methods are not supported by the SMTP protocol.

5.11.5 MIME

Internet applications such as email were originally set up to process ASCII text records up to a specified maximum length, hence the first emails were in a simple text format written in a character set only suitable for English and could not handle attachments. Multipurpose Internet Mail Extensions (MIME) is the protocol that extends email to be able to transmit other data in addition to ASCII text. MIME allows email headers and messages to contain other character sets (for example, UTF-8) and specifies how non-ASCII files should be transmitted as email attachments. MIME works by encoding attachments and other data as ASCII text so that it can be transmitted in a standard email message. MIME also supports multipart email messages, which this course does not consider.

MIME adds the following header types, among others:

- **MIME-Version**
- **Content-description:** human-readable description of message
- **Content-type:** such as text/enriched, application/post script, image/jpeg
- **Content-transfer encoding:** how the object is encoded for transmission

Note that these header types are among those that most email clients do not routinely display to users.

MIME is also used in web protocols for specifying and wrapping different types of downloadable objects.

5.11.6 POP3 and IMAP

These are the two most widely-used protocols for retrieving email from servers. POP3 is the third version of the Post Office Protocol and is used to transfer email from a server to a local device. The protocol assumes that only one device is being used to access email messages, hence messages are usually deleted once downloaded, although some implementations allow mail to be saved for a short time. Email clients using POP3 will communicate with a POP3 server. The SMTP server accepts incoming email and groups it into user mailboxes. The POP3 server allows the recipient to view their mailbox and download messages. When the user connects to the server, all new messages are downloaded. Email clients must use the POP3 protocol to communicate with the POP3 server, and the POP3 and SMTP servers must communicate so that while the user is downloading their new messages with POP3 the SMTP server can still accept new messages for them and keep the mailbox current and free from errors.

IMAP, the Internet Message Access Protocol, that is now on its fourth version, allows users to view their messages on a server, and to forward, delete, write and send them from there. Users can also organise messages into folders

on the server, and search messages with a search string. The protocol allows messages to stay in the server's mailbox, and be viewed from any device with an email client that can make a connection. The protocol will ensure that any local copies of messages are synchronised with messages on the server.

The most important difference between the two protocols is that POP3 mail clients will download mail from the server onto the user's computer before it can be viewed, followed by optional deletion of the server copy of the message; whereas IMAP clients leave messages on the server's filespace and allow the user to view the message contents remotely. Some advantages and disadvantages are as follows.

- POP3 advantage: messages can be viewed offline, without having to stay connected to the server.
- POP3 disadvantage: once messages are downloaded and deleted from the server they are no longer accessible from other hosts.
- POP3 disadvantage: the user can only view a list of messages, download and delete messages from the server. In order to view and edit messages they must be downloaded, which can increase network traffic.
- IMAP advantage: messages on the server can be accessed from any host computer connected to the Internet.
- IMAP advantage: less network traffic as messages can be viewed and managed on the server.
- IMAP disadvantage: messages are only available while connected to the server.

The first and last considerations are much less important now than they were earlier in the internet's history: most users have 'always-on' broadband connections for their home computers, and have prepaid data packages for their mobile phones where they pay a fixed price for web access.

Email clients support both POP3 and IMAP; some may also implement proprietary protocols.

5.12 Overview of the chapter

In this chapter we looked at how the world wide web was developed and how it works. We considered the history of HTML in the context of the historical development of web browsers, and then we looked at HTML5 in some detail. We concluded by looking at the protocols necessary for successful email transmission and receipt.

5.13 Reminder of learning outcomes

Having completed this chapter, and the Essential reading and activities, you should be able to:

- explain in general terms how web documents are transferred across the Internet and what processes are triggered when you click on a hyperlink
- code web pages using style sheets
- describe some of the technologies available for dynamic web pages, and be able to select the appropriate type of technology for your requirements
- describe the most widely used email protocols and summarise the important differences between them.

5.14 Test your knowledge and understanding

5.14.1 Sample examination questions

(a) (i) Which of the following are protocols used for electronic mail? [2 marks]

1. SMTP
2. POP3
3. IMAP
4. All of the above

(ii) Which one of the following is a scripting language? [2 marks]

1. HTTP
2. PHP
3. FTP
4. None of the above

(iii) DNS is: [2 marks]

1. A protocol to map domain names to numerical addresses
2. A protocol for communicating data in HTML format
3. A form of SGML that is now deprecated
4. None of the above

(b) (i) What does it mean to say that HTTP is a **stateless** protocol? [3marks]

(ii) Explain what is mean by **fault tolerance** in HTML. What is the advantage of fault tolerance? [6 marks]

(c) (i) What is the advantage of using the <thead>, <tbody> and <tfoot> tags to divide an HTML table into sections? [5 marks]

(ii) The *partc.css* file has the following content:

```
table {
    border-collapse: collapse;
}

td {
    border: 1px solid #000;
}
```

The contents of the HTML file *Ch5ExamQ.html* are given below:

```
<!DOCTYPE html>
<html lang = "en">
<head>
    <meta charset="UTF-8"/>
    <title>Ch 5 Exam Q</title>
</head>

<body>
    <table>
        <thead>
            <tr>
                <th>Year</th>
                <th>2016</th>
```

```
        <th>2017</th>
        <th>2018</th>
    </tr>
</thead>
<tbody>
    <tr>
        <th>widgets</th>
        <td>11</td>
        <td>72</td>
        <td>917</td>
    </tr>
    <tr>
        <th>wosnames</th>
        <td>5</td>
        <td>7,000</td>
        <td>964</td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <th>TOTAL</th>
        <td>16</td>
        <td>7,072</td>
        <td>1,881</td>
    </tr>
</tfoot>
</table>
</body>
</html>
```

When viewed in a web browser the file produces a table similar to Table 5.1. In your answer book draw the table as you would expect to see it in a web browser after the *Ch5ExamQ.html* file is linked to the *partc.css* file.

[5 marks]

Year	2016	2017	2018
widgets	11	72	917
wosnames	5	7,000	964
TOTAL	16	7,072	1,881

Table 5.1: *Ch5ExamQ.html*.

Chapter 6: Legal framework

6.1 Introduction

As an IT professional you will need to know what rights you have over software you have developed, what responsibilities you have to users of your systems and what your obligations are under national law. As a private individual it is in your interest to be informed of your rights and responsibilities when you use networked systems, for instance sending emails, publishing web documents or downloading material from the web.

This chapter focuses on UK legislation, and EU law. EU laws that place restrictions on the transfer of personal data from the EU to other jurisdictions are particularly relevant to IT professionals. We also discuss the **Council of Europe Convention on Cybercrime**, which has influenced the law relating to computer crime in the EU and worldwide, with 61 countries ratifying the Convention (or the legal equivalent, meaning 61 countries have accepted an obligation to bring their laws into accord with the treaty). Countries outside of Europe who have ratified the Convention include the USA, Japan, Australia and Argentina. You can find a list of countries that have signed the Convention here: www.coe.int/en/web/conventions/full-list/-/conventions/treaty/185/signatures

As with just about all topics dealt with in this subject guide, computer law is in a state of development, to cope with new technological and social realities. You should try and keep up-to-date about new technology, and political, social and legal issues with the technology, by reading magazines, web pages and technology sections in newspapers.

6.1.1 Aims of the chapter

This chapter aims to help you to develop a good understanding of the major issues in law that computing professionals need to be aware of including intellectual property, privacy and unauthorised access to computers and computer networks. We will discuss how the law has responded to the challenges of the computer age and the internet, including defining new offences, defining new responsibilities for individuals and organisations plus providing new rights and protections for individuals. We will consider the law in the UK and the EU, as well as the **Council of Europe Convention on Cybercrime**, which has international applications.

This chapter will concentrate on the areas of intellectual property, computer misuse (hacking and virus writing) and data protection and privacy. We will also briefly consider issues of defamation, and it is important to be aware that computer files and electronic communications are just as much subject to laws on blackmail, libel and obscenity laws as are the traditional print-based media.

6.1.2 Learning outcomes

By the end of this chapter, and having completed the Essential reading and activities, you should be able to:

- explain the difference between patent and copyright, describe some ways in which internet technology has made it more difficult to enforce copyright or monitor copyright infringements and discuss arguments for and against extending patent protection to computer software
- list the principles of data protection

- explain what rights you have as a data subject in relation to persons or organisations holding your details on file
- explain what companies must do to keep within the law if they keep records of individuals on manual or electronic files (assuming UK jurisdiction)
- explain the legal implications of computer hacking, virus writing and unauthorised access to computer systems.

6.1.3 Essential reading

This chapter in the subject guide gives a complete account of the topics covered and is Essential reading.

6.1.4 Further reading

- Gillespie, A. A. *Cybercrime: Key Issues and Debates*. (Abingdon, Oxon; New York, NY: Routledge, 2015) [ISBN 9780415712217 (hbk); 9780415712200 (pbk); 9781315884201 (ebk)]. Chapter 1 *What is cybercrime?*; Chapter 2 *Jurisdiction*; Chapter 3 *Targeting the Technology*; Chapter 4 *Offences relating to data*; Chapter 5 *Cyberterrorism and Cyberwarfare* (discusses when hacking becomes terrorism); Chapter 6 *Fraud* (discusses the types of fraud committed online); and Chapter 7 *Virtual property* (includes IP).
- Holt, J. and J. Newton (eds) *A Manager's Guide to IT Law*. (London: British Computer Society, The Chartered Institute for IT, 2011) 2nd edition [ISBN 9781906124755 (pbk); 9781780170039 (ebk)].
- Bott, F. *Professional Issues in Information Technology*. (Swindon: BCS Learning and Development, The Chartered Institute for IT, 2014) 2nd edition [ISBN 9781780171807 (pbk); 9781780171821 (ebk)].

6.1.5 References cited

- www.wipo.int/wipolex/en/treaties/text.jsp?file_id=283698 – text of the Berne Convention
- <https://ico.org.uk/> – The UK Information Commissioner's Office
- www.wto.org/english/tratop_e/trips_e/intel2_e.htm – overview of the TRIPS agreement
- www.wto.org/english/docs_e/legal_e/31bis_trips_04_e.htm – text of the TRIPS agreement
- www.wipo.int/edocs/pubdocs/en/intproperty/611/wipo_pub_611.pdf – text of the Paris Convention
- www.wipo.int/treaties/en/ip/washington/ – text of the Washington Treaty
- www.wipo.int/wipolex/en/treaties/text.jsp?file_id=295166 – text of the WIPO Copyright Treaty (WCT)
- www.wipo.int/treaties/en/ip/wct/summary_wct.html – summary of the WCT
- www.wipo.int/treaties/en/ip/wppt/ – summary of the The WIPO Performances and Phonograms Treaty (WPPT)
- www.wipo.int/wipolex/en/treaties/text.jsp?file_id=295578 – text of the WPPT
- <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31996L0009:EN:HTML> – the EU Database Directive
- <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32001L0029:EN:HTML> – the EU Copyright Directive
- www.gov.uk/government/news/quashing-of-private-copying-exception
- <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31991L0250:EN:HTML> the EU Computer Programs Directive

- www.wipo.int/sme/en/documents/software_patents_fulltext.html – WIPO statement on patenting software
- www.epo.org/law-practice/legal-texts/html/epc/2016/e/ar52.html – the European Patent Convention
- www.wipo.int/patents/en/faq_patents.html – WIPO patent FAQ
- www.eff.org/files/annual-report/2017/index.html#InnovationAlice

6.2 Glossary of key terms

- **Devops:** an approach to software development combining software development with information technology operations.
- **Patent:** the legal right to exploit a new invention or discovery, granted to individuals and organisations for a time-limited period. Patents have to be applied for; there is normally a fee and a checking process. Patents also have to be renewed or they will lapse. While the patent is in force the owner of the patent has the sole right to exploit the invention.
- **Registered design:** Similar to a patent, in the UK designs can be registered to protect the appearance, physical shape, configuration and decoration.
- **Registered trademark:** a trademark is a design and/or words that are identified with a particular company or product. Trademarks can be registered to prevent competitors from using any part of them.
- **Copyright:** given for original creative work, which can be written, musical or visual, or a combination thereof. Copyright is automatically conferred on the originator of the new material, and means that they have the exclusive right to make and licence copies for a fixed period of time.
- **Design right:** There is also an automatically conferred 'design right' which, in the UK, protects the 'shape and configuration' of a design, see: www.gov.uk/design-right
- **Intellectual property (IP):** intangible property, where ownership rights can be conferred by copyright, design rights, patents, registered designs and registered trademarks.
- **Defamation:** publishing something that: (1) is damaging to the reputation of an individual or organisation; and (2) is untrue. In the UK, the **Defamation Act** of 2013, re-defined defamation as it applies in law as follows:
 - (1) A statement is not defamatory unless its publication has caused or is likely to cause serious harm to the reputation of the claimant.
 - (2) For the purposes of this section, harm to the reputation of a body that trades for profit is not "serious harm" unless it has caused or is likely to cause the body serious financial loss.
- **Publishing:** In terms of the law regarding defamation, publishing means making known to a third party. 'Publication' includes speech, broadcast, books, newspapers and magazines, emails and web pages. UK law distinguishes between publication in permanent form (for example, radio and television broadcasts) and publication in temporary form; for example, words privately spoken to another person.
- **Libel:** defamation that is published in permanent form.
- **Slander:** defamation that is published in a temporary form.
- **Council of Europe:** The Council was founded in 1949, and now has 47 members, 28 of which are also members of the EU. Members have signed up to the **European Convention on Human Rights**. It describes itself

as 'the continent's leading human rights organisation' and oversees the European Court of Human Rights, which enforces the European Convention on Human Rights.

- **European Union (EU):** the body overseeing the framework and agreements that provide for the political and economic union of member states, of which there are 28 at the time of writing (2019), including the UK.
- **EU directive:** an instruction from the EU to member states to enact the measures contained in the directive into national law. EU members must implement directives.
- **Convention on Cybercrime:** also known as the **Budapest Convention** after the tradition of naming conventions after the city where they are signed. This convention of the **Council of Europe** lays out a framework for the law on cybercrime for each signatory to implement within its own jurisdiction.
- **WTO (World Trade Organization):** 'the only global international organization dealing with the rules of trade between nations. At its heart are the WTO agreements, negotiated and signed by the bulk of the world's trading nations and ratified in their parliaments.' See: www.wto.org/english/thewto_e/thewto_e.htm
- **Paris Convention:** The 1883 Paris Convention for the Protection of Industrial Property was an agreement to establish a union between 11 countries to protect IP including patents, but did not include copyright. The Convention is now administered by WIPO who describe it as 'the first major step taken to help creators ensure that their intellectual works were protected in other countries.'
- **Berne Convention:** Formally known as the Berne Convention for the Protection of Literary and Artistic Works, it attempted to codify and standardise copyright law in all ratifying countries, see: www.wipo.int/treaties/en/ip/berne/
- **TRIPS (Agreement on Trade-Related Aspects of Intellectual Property Rights):** Many of the provisions of the Berne and Paris Conventions were incorporated into TRIPS, negotiated in 1994. TRIPS is about countries respecting each other's IP rights, hence it introduced IP rights into international trade, and must be followed by all members of the WTO.
- **The three-step test:** This test was added to the Berne convention in 1967, and was incorporated into TRIPS. In a very non-specific way it grants countries the right to enact laws that 'permit the reproduction of [copyright protected work] in certain special cases, provided that such reproduction does not conflict with a normal exploitation of the work and does not unreasonably prejudice the legitimate interests of the author'.
- **Fair use:** Fair use or **fair dealing** law derives from the three-step test in order to permit limited reproduction of copyright protected works without the permission of the author or rights holder.
- **World Intellectual Property Organization (WIPO):** a self-funding agency of the United Nations established in 1967, with the aim of developing and promoting a global system for the protection of IP. At the time of writing (August 2019), WIPO had 192 members, www.wipo.int/about-wipo/en/ and administered 26 treaties: www.wipo.int/treaties/en/index.html
- **WCT (WIPO Copyright Treaty):** established under the Berne Convention to address copyright law in the digital domain, see: www.wipo.int/treaties/en/ip/wct/

- **Contracting party:** a person or organisation entering into an agreement with another or others. In terms of international treaties, contracting parties are nation states. Signing a treaty does not commit the nation to being bound by its terms, but only to further consider the treaty. Once a nation state has ratified a treaty it is bound by its terms, accepting the rights, advantages and obligations conferred. Accession to a treaty, means that a state agrees to an offer to become a contracting party; it has the same effect as ratification.
- ***Sui generis*:** if a thing is described to be *sui generis* it means that nothing like it already exists. So a *sui generis* legal right would be one without precedent in law or practice.
- **SMEs:** Small and medium-sized enterprises. These are precisely defined with metrics by the EU, who note that they represent 99 per cent of EU businesses. See: http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition_en
- **Patent trolls:** individuals and companies that buy up and register patents with the business model of suing individuals and organisations for patent violation. Since patent litigation is expensive most individuals and SMEs will settle early, whatever the merits of their case.
- **Net neutrality:** Net neutrality means that ISPs provide the same access to all websites for all of their customers. Without net neutrality ISPs may be able to slow down (or even block) your access to certain sites, unless those sites pay for premium access.
- **General Data Protection Regulation (GDPR):** Approved by the EU in 2016, and in force across member states of the EU from 25 May 2018, the Regulation represents, at the time of writing, the most comprehensive and powerful data protection law in the world.
- **Take down request:** A formal procedure to ask an ISP or a search engine to remove web content, or to block access to such content. The grounds may be that the content is illegal; for example, in breaching copyright. In some jurisdictions individuals have the right to ask for outdated information to be taken down, known as the right to be forgotten; for example, the EU's GDPR grants individuals this right.
- **Traffic data:** 'any data processed for the purpose of the conveyance of a communication on an electronic communications network or for the billing in respect of that communication and includes data relating to the routing, duration or time of a communication'. See: <https://ico.org.uk/for-organisations/guide-to-pecr/communications-networks-and-services/traffic-data/>

6.3 Magazines and web pages of interest to computing professionals

Note this section is not examinable.

Listed below are magazines and web pages that will help you to understand what the current and emerging issues of particular interest to IT professionals are:

- www.acm.org/ – The Association for Computing Machinery, based in the USA, describes itself as follows:

ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and a profession. ACM provides the computing

field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

The ACM publishes several magazines; you can find a list here, <https://dl.acm.org/mags.cfm>, including:

- <https://cacm.acm.org/> – ACM's flagship magazine, *Communications of the ACM*, 'is the premier chronicler of computing technologies, covering the latest discoveries, innovations, and research that inspire and influence the field.'
- <https://xrds.acm.org/> – *XRDS*, is the official magazine for student members. Published quarterly, with an online presence, each issue has a theme and includes career advice linked to the theme.
- <https://arstechnica.com> – A site founded in 1998, aimed at 'technologists and IT professionals'. Jon Stokes is a co-founder and was CPU Editor until 2010 and Deputy Editor from 2008–2011. The site describes itself with 'we specialize in news and reviews, analysis of technology trends, and expert advice on topics ranging from the most fundamental aspects of technology to the many ways technology is helping us discover our world.'
- www.computerweekly.com/ – This is a website that was formerly a print magazine. Based in the UK it is aimed at European IT professionals, although they claim an international readership and 'in-depth coverage of the issues, challenges and trends facing today's IT leaders'.
- www.computerworld.com/about/about.html – Based in the USA, *Computerworld* focuses on business computing, and describes itself as follows:

Computerworld: The Voice of Business Technology.
Computerworld focuses on empowering enterprise users and their managers, helping them create business advantage by skilfully exploiting today's abundantly powerful web, mobile, and desktop applications. Computerworld also offers guidance to IT managers tasked with optimizing client systems—and helps businesses revolutionize the customer and employee experience with new collaboration platforms.

- <https://thehackernews.com/> – *The Hacker News*, unsurprisingly, focuses on network security. It sums up its mission as follows:

Since almost every organization in the present world is connected to the Internet in some or the other way, steps must be taken to ensure their networks remain safe and secure, and that's exactly what our mission is about. The Hacker News (THN) is a leading, trusted, widely-acknowledged dedicated cybersecurity news platform, attracting over 8 million monthly readers including IT professionals, researchers, hackers, technologists, and enthusiasts. The Hacker News features latest cyber security news and in-depth coverage of current as well as future trends in Infosec and how they are shaping the cyber world.

The *Hacker News* also organises events for the 'best IT professionals and hackers', including the yearly Hackers Conference in Delhi, which attracts 'ignited minds in the field of Internet security'.

- www.theregister.co.uk/ – slogan ‘Biting the hand that feeds IT’. Founded in the UK in 1994, it now has offices in London, San Francisco and Sydney. *The Register* describes itself as ‘a leading global online tech publication, with more than nine million monthly unique browsers worldwide’ that provides ‘[i]ndependent news and views for the tech community’. Somewhat more irreverent than other sources (an example of a 2018 headline: ‘Grumbling about wobbly Windows 10? Microsoft can’t hear you over the clanging cash register’), with Business, Devops, Software, Security and Emergent Tech sections, among others. *The Register* describes its readership as follows:

Many Register readers are technology professionals but we’re also read by technology enthusiasts, shed boffins, policy wonks and science fans around the globe.

- www.technologyreview.com/ – Web page of the *MIT Technology Review* magazine. Founded at the Massachusetts Institute of Technology in 1899, today it describes its mission as:

Every day, we provide an intelligent, lucid, and authoritative filter for the overwhelming flood of information about technology. We do this with serious journalism, written in clear, simple language, by a knowledgeable editorial staff, governed by a policy of accuracy and independence. We do this in features, news analysis, business reports, photo essays, reviews, and interactive digital experiences that invite our readers to probe deeper, examine data, and get to know experts and their opinions to see, explore, and understand new technologies and their impact.

The magazine publishes a yearly list of the most influential technologies, and is also known for promoting innovators under 35.

- www.wired.co.uk/ and www.wired.com/. *Wired* magazine, founded in the USA in 1993, is aimed at a general readership, and now has subsidiaries in Europe and Japan. *Wired* describes itself as being ‘about what’s next – bringing you the people, the trends and the big ideas that will change our lives’. Of interest to computing professionals are the features analysing and exploring the impacts of new technology in social, political and business terms.
- www.2600.com/ – *The Hacker Quarterly*. Based in the USA, the magazine has an interesting history, and now particularly focuses on online freedom, while maintaining its historic interest in telephony. It actively solicits articles from readers with its web page stating:

Want to write for 2600? If you have an intelligent article on something interesting you think we’d like to publish, send it to articles@2600.com. If one of your articles gets printed, you’ll get either a free shirt or a one year subscription (or a year of back issues). If you get more articles printed, you’ll get even more stuff and will eventually be listed as a “known associate” of 2600 in various government files!

- www.eff.org/ – The Electronic Freedom Foundation (EFF) is very much focused on the impact of technological, legal and political changes on free speech issues. They describe themselves as the: ‘leading nonprofit organization defending civil liberties in the digital world’. Because of their focus their website may be one of the first to pick up on proposed legal changes worldwide and their impact on issues of interest to IT professionals. Their 2017 Annual Report describes, *inter alia*, their part in

the fight to protect software developers and other innovators from patent trolls (very much an issue in the USA where software is much more likely to be patented than in the rest of the world) and to protect net neutrality.

- <https://www.schneier.com/> Blog of computer security guru Bruce Schneier.

6.4 Intellectual property

Intellectual property is property that is intangible, but has potential commercial and industrial application and value. Intellectual property can be divided into two camps, firstly industrial property such as patents, trademarks and designs; and secondly copyright, given to artistic and literary works. During the late 19th century, efforts were made to globalise the protection of IP rights provided by individual states to businesses and citizens.

6.4.1 An historical note: The Berne Convention

The **Berne Convention for the Protection of Literary and Artistic Works** addressed the issue that copyright in one country could not be enforced in another. It was agreed in Berne in 1886. The Berne Convention was initially signed by 10 countries, including the UK and six other European states. Under the Convention, creators receive copyright automatically, and members are required to recognise copyright granted in all jurisdictions that are parties to the Convention. The Berne Convention also stated that protection must be given not just to the right to make copies, but also to related rights, such as to authorise translations, make adaptations (for example, adapting a book into a stage play) or to give performances and recitations.

The Berne Convention has been revised several times, most recently in 1971, see: www.wipo.int/wipolex/en/treaties/text.jsp?file_id=283698 The Convention had 177 contracting parties at the time of writing in 2019, and is now administered by World Intellectual Property Organisation (WIPO).

6.4.2 An historical note: The Paris Convention

The 1883 Paris Convention for the Protection of Industrial Property was an agreement to establish a union between 11 countries (not including the UK) to protect IP including patents, industrial designs, trademarks and trade names. The Paris Convention also intended to protect business from unfair competition, and to enforce that geographical locations given by goods must be correct. The Convention is now administered by WIPO. In 2017 there were 177 contracting parties, see: www.wipo.int/treaties/en/ShowResults.jsp?lang=en&treaty_id=2 for a list of contracting parties.

WIPO describes the Paris Convention as ‘the first major step taken to help creators ensure that their intellectual works were protected in other countries.’ See: www.wipo.int/treaties/en/ip/paris/ The Paris Convention was most recently revised in 1967. Also see: www.wipo.int/wipolex/en/treaties/text.jsp?file_id=288514

6.4.3 TRIPS

Historically, the Berne Convention of 1886 addressed the issue of international protection for copyright holders, while the Paris Convention of 1883 did the same thing but for patents and other forms of ‘industrial property’. Many of the provisions of the Berne and Paris Conventions were incorporated into the Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS), negotiated in 1994. The Agreement stipulates minimum legal standards for IP rights, which all members of the World Trade Organization must implement. TRIPS is about countries respecting each other’s IP rights, hence it introduced IP rights into international trade. TRIPS specifies how to enforce IP rights

between members, as well as procedures for dispute resolution. The Agreement attempts to balance rights and obligations with promoting innovation and economic growth. It has been criticised for placing an economic burden on poorer countries, and in fact facilitating, at least in the short-term, a transfer of wealth from poorer countries (who were also IP rights poor) to the world's richest countries. Similarly TRIPS has been criticised for failing to stimulate the economies of poorer countries, despite claims that it would do so.

Since all members of the WTO must implement the protections outlined in TRIPS, your own country must have enacted these minimum standards into law if it is a member of the WTO. You can find a list of WTO members here: www.wto.org/english/thewto_e/whatis_e/tif_e/org6_e.htm

6.5 IP protection and TRIPS

The TRIPS Agreement sets out:

the minimum standards of protection to be provided by each Member. Each of the main elements of protection is defined, namely the subject-matter to be protected, the rights to be conferred and permissible exceptions to those rights, and the minimum duration of protection.

See: www.wto.org/english/tratop_e/trips_e/intel2_e.htm

Under TRIPS, members can give more extensive protection than the Agreement outlines, but must give the minimum as defined in the agreement, and must respect and enforce the rights of citizens of all contracting parties.

See: www.wto.org/english/docs_e/legal_e/31bis_trips_03_e.htm for the text of the agreement.

The agreement provides for the protection of seven kinds of IP, which are described below. Only copyright, patents and trade secrets are applicable to this course.

6.5.1 Copyright

TRIPS states that 'Copyright protection shall extend to expressions and not to ideas, procedures, methods of operation or mathematical concepts as such', and adopts Articles 1 through 21 of the Berne Convention (1971), which, among other things, defines works that will attract copyright and the rights of the copyright holder. The Berne Convention states that its purpose is to protect the rights of authors of 'literary and artistic works'. The Convention defines such works as:

The expression "literary and artistic works" shall include every production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression, such as books, pamphlets and other writings; lectures, addresses, sermons and other works of the same nature; dramatic or dramatico-musical works; choreographic works and entertainments in dumb show; musical compositions with or without words; cinematographic works to which are assimilated works expressed by a process analogous to cinematography; works of drawing, painting, architecture, sculpture, engraving and lithography; photographic works to which are assimilated works expressed by a process analogous to photography; works of applied art; illustrations, maps, plans, sketches and three-dimensional works relative to geography, topography, architecture or science.

See: www.wipo.int/wipolex/en/treaties/text.jsp?file_id=283698

Articles 8 to 14 specify the rights that authors have over their work, such as the right to authorise translations or to authorise a broadcast of their work. In addition to Articles 1–19, TRIPS extends the protections given by the Berne Convention to software and databases, adds the right to authorise rentals to the rights of copyright holders, and gives some additional rights to performers to allow and prohibit the recording of their work.

6.5.2 Copyright in software and databases

Article 10 of TRIPS extended copyright protection to computer programs and databases:

Article 10

Computer Programs and Compilations of Data

1. Computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention (1971).
2. Compilations of data or other material, whether in machine readable or other form, which by reason of the selection or arrangement of their contents constitute intellectual creations shall be protected as such. Such protection, which shall not extend to the data or material itself, shall be without prejudice to any copyright subsisting in the data or material itself.

See: www.wto.org/english/docs_e/legal_e/31bis_trips_04_e.htm

Note that databases can be digital or in ‘other form’, which would include any form that imposes structure on data; for example, a card index system might qualify for protection.

Article 11 deals with rental rights for ‘computer programs and cinematographic works’, giving authors the rights to permit or forbid ‘the commercial rental to the public of originals or copies of their copyright works’. Note that computer programs are not sold to the public, but licensed for their use.

6.5.3 Trademarks

Trademarks were protected under the Paris Convention, and that protection is extended by TRIPS. Under the TRIPS agreement trademarks are defined as:

Any sign, or any combination of signs, capable of distinguishing the goods or services of one undertaking from those of other undertakings, shall be capable of constituting a trademark.

See: www.wto.org/english/docs_e/legal_e/31bis_trips_04_e.htm

6.5.4 Geographical indications

Geographical indications were protected under the Paris Convention, and that protection is extended by TRIPS. Under the TRIPS agreement they are defined in Article 22 as:

Geographical indications are, for the purposes of this Agreement, indications which identify a good as originating in the territory of a Member, or a region or locality in that territory, where a given quality, reputation or other characteristic of the good is essentially attributable to its geographical origin.

See: www.wto.org/english/docs_e/legal_e/31bis_trips_04_e.htm

Perhaps the most famous geographical indicator is that of Champagne, defined to be a sparkling wine from the Champagne region in France. In fact, Article 23 of TRIPS is entirely about wines and spirits.

6.5.5 Industrial designs

Industrial designs were defined by the Paris Convention to be:

(e) Industrial designs may be described as consisting of those ornamental aspects or elements of a useful article, including its two dimensional or three-dimensional features of shape and surface, which make up the appearance of the article. The proprietor of such industrial design will usually, under existing legislations, have the exclusive right to make, sell and use articles embodying such design.

See: www.wipo.int/edocs/pubdocs/en/intproperty/611/wipo_pub_611.pdf

Articles 25 and 26 of TRIPS deal with the requirements for protection, and the protection, of industrial designs.

6.5.6 Patents

TRIPS defines patentable subject matter as:

Subject to the provisions of paragraphs 2 and 3, patents shall be available for any inventions, whether products or processes, in all fields of technology, provided that they are new, involve an inventive step and are capable of industrial application.

See: www.wto.org/english/docs_e/legal_e/31bis_trips_04c_e.htm

Note that paragraphs 2 and 3 describe certain exemptions to patentable subject matter that members **may** apply. Computer programs and software are not included in the allowed exceptions, so it is arguable that patents could be applied to software under the TRIPS Agreement.

6.5.7 Layout-designs (topographies) of integrated circuits

Articles 35–38 allow for the protection of the layout of integrated circuits. Of course, processes and components may have their own patents, but the layout of a circuit, while commercially valuable information, does not meet the definitions of industrial design, patent or copyright attracting material. For obvious reasons the layout cannot be kept as a trade secret. Given their commercial value, TRIPS requires that members extend protection to integrated circuits by incorporating certain aspects of the Washington Treaty. This treaty is formally known as the **Treaty on Intellectual Property in Respect of Integrated Circuits**, referred to in the TRIPS Agreement as IPIC. The Washington Treaty was negotiated and signed in 1989, see: www.wipo.int/treaties/en/ip/washington/

6.5.8 Trade secrets

Trade secrets are a form of intellectual property, which the TRIPS agreement requires member states to give legally enforceable protection to. Trade secrets are information that is commercially sensitive, and can be protected provided that the information:

- (a) is secret in the sense that it is not, as a body or in the precise configuration and assembly of its components, generally known among or readily accessible to persons within the circles that normally deal with the kind of information in question;
- (b) has commercial value because it is secret; and

- (c) has been subject to reasonable steps under the circumstances, by the person lawfully in control of the information, to keep it secret.

See **Section 7: protection of undisclosed information**

www.wto.org/english/docs_e/legal_e/31bis_trips_04d_e.htm

6.6 IP protection in international treaties post-TRIPS

The important copyright treaties after TRIPS are **The WIPO Copyright Treaty of 1996 (WCT)** and the **WIPO Performances and Phonograms Treaty of 1996 (WPPT)**, which together are known informally as the **Internet Treaties**.

6.6.1 The WIPO Copyright Treaty of 1996

The WIPO Copyright Treaty of 1996 (known as the WCT) was about protecting copyright in the digital age, expanding on issues addressed by TRIPS. The preamble to the Treaty stated that contracting parties recognised 'the profound impact of the development and convergence of information and communication technologies on the creation and use of literary and artistic works'. See: www.wipo.int/wipolex/en/treaties/text.jsp?file_id=295166

The Treaty followed the TRIPS agreement in stating that copyright applies to:

- (i) computer programs, whatever the mode or form of their expression; and (ii) compilations of data or other material ("databases"), in any form, which, by reason of the selection or arrangement of their contents, constitute intellectual creations.

See: www.wipo.int/treaties/en/ip/wct/summary_wct.html

The Treaty goes on to outline three rights available to authors:

1. The right to distribute original and copies of their work.
2. The right to authorise or forbid rentals.
3. The right of communication to the public.

The right of rental is a new right that mostly mirrors that introduced in the TRIPS agreement. Like the TRIPS agreement the right is to be enforced for computer programs, and for cinematographic works but only in those cases where copying has been so widespread that it has had the effect of 'materially impairing the exclusive right of reproduction'. Added to this is the right of 'works embodied in phonograms as determined in the national law of Contracting Parties (except for countries which, since 15 April, 1994, have had a system in force for equitable remuneration of such rental).'

Reflecting the digital age the rights of authors are broken into the right to make and distribute copies, and the right to communicate to the public, which is defined to be:

the right to authorize any communication to the public, by wire or wireless means, including "the making available to the public of works in a way that the members of the public may access the work from a place and at a time individually chosen by them". The quoted expression covers, in particular, on-demand, interactive communication through the Internet.

See: www.wipo.int/treaties/en/ip/wct/summary_wct.html

This allows authors to allocate digital rights to different parties than those given the right to make analogue copies.

6.6.2 The WIPO Performances and Phonograms Treaty of 1996

The WIPO Performances and Phonograms Treaty (WPPT) deals with the rights of two kinds of beneficiaries, particularly in the digital environment: (i) performers (actors, singers, musicians, etc.); and (ii) producers of phonograms (persons or legal entities that take the initiative and have the responsibility for the fixation of sounds).

See: www.wipo.int/treaties/en/ip/wppt/

The preamble to the WIPO Performances and Phonograms Treaty stated that contracting parties recognised 'the profound impact of the development and convergence of information and communication technologies on the production and use of performances and phonograms', see: www.wipo.int/wipolex/en/treaties/text.jsp?file_id=295578

The WPPT, similarly to the WCT, separated the right of making available to the public 'by wire or wireless means' from the right of reproduction. The effect of both treaties was to make the internet a separate domain in terms of copyright law.

6.6.3 Moral rights

The Berne Convention was revised in 1928 to include **moral rights**, the right of the author to be identified as the author separate from their 'economic rights' including the right to 'object to any distortion, mutilation or other modification of, or other derogatory action in relation to, the said work, which would be prejudicial to his honor or reputation.' See Article 6*bis*. Hence even if an author has assigned copyright to another party and no longer derives revenue from a work, they still have the right to be recognised as the author and to object to any changes to their work that could damage their reputation. Moral rights were not included in TRIPS, which excluded 6*bis* from the Articles of the Berne Convention covered by TRIPS (Articles 1–21 and the appendix, hence all articles from the Berne Convention concerning copyright except for 6*bis* were included in TRIPS). Hence any country that is a member of the WTO but has not signed the Berne Convention is not obliged to protect the moral rights of authors unless they have signed the Internet Treaties.

6.6.4 Moral rights in the Internet Treaties

The WCT stated that articles 1 through 21 (and the appendix) of the Berne Convention as revised in 1971 were to be complied with by contracting parties. The WCT includes Article 6*bis*, hence moral rights of authors are to be protected by contracting parties. The WPPT grants moral rights to performers in Article 5; that is to say a performer has the right to be identified as taking part in a performance, and the right to 'object to any distortion, mutilation or other modification of his performances that would be prejudicial to his reputation.'

Activity 6.1

Answer **true** or **false** to each of the following statements. Where you think that the answer is false, state your reasons in no more than three sentences:

1. Trade Secrets are a form of intellectual property.
2. Arguably, a computer program could be patented under TRIPS, if it included an *inventive step*.
3. Since copyright applies to literary and artistic works, it cannot be applied to computer programs.

4. TRIPS is an international convention that introduced IP rights into international trade.
 5. Unlike copyright, patent law is not covered by any international treaties.
 6. Every country ratifying TRIPS must give all authors the same copyright rights and protections as all other countries.
 7. Authors and originators have both moral and economic rights over their work. They can sell both rights separately.
-

6.7 Database right in the EU

Individuals and companies within the EU have a 'database right', a right applying exclusively to databases, which introduces a new form of intellectual property: the work of compiling the database.

In introducing the database right, the EU was concerned that the contents of a database may be copied and rearranged in such a way that the content is identical, but technically no copyright or other laws have been broken. However, the originator of the database may have lost the financial and/or professional investment made in setting up the database and sourcing its contents. Hence database originators were given a *sui generis* (without precedent) right to prevent, or give remedy against, electronic copying. Provided that the database owner can show that their investment has been substantial, they have the database right. The investment can be financial or in terms of the work undertaken. This right only applies to databases created inside the EU, and is independent of the copyright status of the material in the database.

The TRIPS Agreement, as well as the WCT, agree that the right to copyright must be extended to databases, but only so far as 'the selection or arrangement of their contents constitute intellectual creations' (TRIPS article 10 and WCT article 5). The effect of the protection of databases in terms of selection or arrangement means that some creativity or element of choice must be made in selecting or arranging the database in order for copyright protection to apply. So a database is not protected by copyright where the content is supposed to comprehensively include all data that meets an objective definition, since in this case there is no creativity in arrangement of data, or element of choice in the selection of data. In the USA copyright in databases has been enacted in the Digital Millennium Copyright Act of 1998. The Supreme Court famously ruled that a telephone directory cannot have the copyright protection afforded to databases as arranging names alphabetically is expected of a telephone directory and shows no creativity or originality in arrangement. Furthermore since all subscribers in a certain geographical area are included in a telephone directory there is no element of choice in the selection.

A person might collate a database of, for example, favourite Mark Twain quotes, and this could be protected, since there would be some choice in the selection. This would mean that the entire database would be protected from copying, as well as protection from copying substantial parts of the database, whatever the copyright status of the material contained in the database.

This is not the case in the European Union, where, to the rights given by TRIPS and the WCT have been added a *sui generis* right of database protection. This was introduced by EU Directive 96/9/EC in 1996, known as the Database Directive.

See: <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31996L0009:EN:HTML>

The Database Directive introduces the copyright protection of databases defined in TRIPS and the WCT, in paragraphs 15 and 16 of the introduction:

- (15) Whereas the criteria used to determine whether a database should be protected by copyright should be defined to the fact that the selection or the arrangement of the contents of the database is the author's own intellectual creation; whereas such protection should cover the structure of the database;
- (16) Whereas no criterion other than originality in the sense of the author's intellectual creation should be applied to determine the eligibility of the database for copyright protection, and in particular no aesthetic or qualitative criteria should be applied;

In addition to this, the EU was concerned that the contents of a database 'may be copied and rearranged electronically [...] to produce a database of identical content which, however, does not infringe any copyright' (paragraph 38 of the introduction to the directive). This copying was thought to amount to 'misappropriation of the results of the financial and professional investment made in obtaining and collection the contents' (paragraph 39 of the introduction). Hence a *sui generis* (without precedent) right was introduced to prevent, or give remedy against, electronic copying including over the internet. Similarly to the WCT and WPPT treaties, the Database Directive distinguishes between the right to make or authorise copies (the right to prevent substantial copying), and the right to display 'on-screen' the database. Where this display involves temporary or permanent transfer of the database, the rights holder should authorise this (paragraph 44 of the introduction).

In order to have this *sui generis* database right, there must be 'qualitatively and/or quantitatively a substantial investment' in the database (Article 7 of the directive). This means that provided that the database owner can show that their investment has been substantial, they have the *sui generis* database right. The investment can be financial or in terms of the work undertaken, as defined in paragraph 40 of the introduction:

- (40) Whereas the object of this *sui generis* right is to ensure protection of any investment in obtaining, verifying or presenting the contents of a database for the limited duration of the right; whereas such investment may consist in the deployment of financial resources and/or the expending of time, effort and energy;

Hence the database owner can protect the work of compiling the database, precisely the right that the Supreme Court of the United States has denied to US businesses. This right is not copyright, it is a new right, defined in paragraph 41 of the introduction as follows:

- (41) Whereas the objective of the *sui generis* right is to give the maker of a database the option of preventing the unauthorized extraction and/or re-utilization of all or a substantial part of the contents of that database; whereas the maker of a database is the person who takes the initiative and the risk of investing; [...]

Note that:

- The owner of the database is the person or organisation providing the investment, which is either the maker of the database, or their employer if undertaken as part of contractual duties.

- The Directive does not define what is meant by 'substantial part' of the database contents.
- The copyright status of the material contained in the database is independent of the right to prevent substantial or complete copying, whether that right is copyright of 'the selection or arrangement' of the database, or the EU database right.
- The database right does not extend to databases created outside of the EU.

6.8 The three-step test and fair use

Article 9(2) of the Berne Convention gives members the right to provide exceptions in order to allow copies to be made in certain circumstances:

(2) It shall be a matter for legislation in the countries of the Union to permit the reproduction of such works in certain special cases, provided that such reproduction does not conflict with a normal exploitation of the work and does not unreasonably prejudice the legitimate interests of the author.

Article 10 specifies that members may legislate to allow quotations from works, without the rights holder's permission, provided that the author is credited. This right is particularly suggested for the purposes of teaching and news reporting.

Article 9(2) has become known as the 'three-step test', and was incorporated into TRIPS as follows:

Article 13

Limitations and Exceptions

Members shall confine limitations or exceptions to exclusive rights to certain special cases which do not conflict with a normal exploitation of the work and do not unreasonably prejudice the legitimate interests of the right holder.

This is also incorporated into the WCT and can be extended to digital rights, as WIPO notes:

As to limitations and exceptions, Article 10 of the WCT incorporates the so-called "three-step" test to determine limitations and exceptions, as provided for in Article 9(2) of the Berne Convention, extending its application to all rights. The Agreed Statement accompanying the WCT provides that such limitations and exceptions, as established in national law in compliance with the Berne Convention, may be extended to the digital environment. Contracting States may devise new exceptions and limitations appropriate to the digital environment. The extension of existing or the creation of new limitations and exceptions is allowed if the conditions of the "three-step" test are met.

See: www.wipo.int/treaties/en/ip/wct/summary_wct.html

6.8.1 Fair dealing

Most countries will have enacted exemptions in their copyright laws under the three-step test. The EU and the USA have copyright law allowing for a certain amount of a work to be quoted with attribution and without permission from the copyright holder for certain purposes, including educational, research and news reporting. In the UK, the **Copyright, Designs and Patents Act 1988** defines 'fair dealing' for many purposes including educational, research and

news reporting. However, fair dealing is not defined in UK law very closely. A rule of thumb is 10 per cent of the original work, but care must be taken when quoting poetry and song lyrics, as they are so short that any amount of quotation, even only a line or two, amounts to a significant part of the work. In fact, publishers will not quote song lyrics that are under copyright without getting explicit permission from the copyright holder, as most music companies proactively protect their copyrights and can, and do, claim damages for quoting without permission.

British law was amended by the **Copyright and Related Rights Regulations 2003**, which implemented the EU Copyright Directive.

6.8.2 EU Copyright Directive of 2001

In the EU, Directive 2001/29/EC incorporated the provision of the WCT, as well as seeking to harmonise copyright law across the EU partly by specifying the exceptions that states may apply to copyright and related rights. The Directive gives member states the rights to keep exceptions already in their national law, but new exceptions can only be taken from the list in the Directive in Article 5. Members do not have to apply the exceptions given in the directive, and how they are applied may vary from member to member.

The only exception that states are required to enact, is the right to make a temporary copy as part of a 'technological process' to either enable lawful use or 'transmission in a network between third parties by an intermediary.' See: <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32001L0029:EN:HTML> The UK **Copyright, Designs and Patents Act 1988** was amended in 2003 to include the right to make temporary copies in both circumstances, and to explicitly allow for 'time-shifting' – recording a broadcast in order to watch or listen to it at a later, and more convenient, time.

The Directive stated that:

Member States should be allowed to provide for an exception or limitation to the reproduction right for certain types of reproduction of audio, visual and audio-visual material for private use, accompanied by fair compensation.

As a consequence most EU countries have allowed for private copies to be made for format shifting (making a copy in order to change the format of a recording) applying the instruction to provide 'fair compensation' to rights holders by, for example, taxing the technology that might be used to make such copies and distributing the income gained to rights holders. In the UK, a more narrow attempt at introducing the right to make private copies of digital material without compensation (not including computer programs – **Copyright and Rights in Performance (Personal Copies for Private Use) Regulations 2014** – was withdrawn in 2015 after a legal challenge from recording companies. The government has said that it has no plans to reintroduce it, see: www.gov.uk/government/news/quashing-of-private-copying-exception

6.8.3 Back-up copies of software and EU law

The right to make a private back-up copy of a computer program was given in the EU Software Directive of 2009 (2009/24/EC), see: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32009L0024&from=EN> This Directive followed on from 91/250/EEC in 1991, requiring a computer program to be protected under copyright as literary works. This implemented the requirements of TRIPS and the WCT.

6.9 Patent and copyright law with respect to software and applications

It is important to distinguish between **copyright** and **patents**. Copyright automatically belongs to the author of any original or creative work, or in certain cases to their employer, and no one else may derive revenue from the work without the copyright holder's permission. Copyright is held in every country that is a member of the WTO, or that has signed the Berne Convention.

A patent on the other hand protects the right to exploit inventions, such as innovative computer hardware: it does not exist automatically but has to be granted by a government patent office, following a registration process which may take a number of years. In addition, patents have to be separately applied for in each country, although WIPO runs a process where a single application covers many countries, see: www.wipo.int/pct/en/; and the European Patent Office runs a process where applicants can apply for a patent valid in several European countries.

There is an essential difference between copyright and patenting with respect to software:

- **Patenting** concerns the technical content of a program rather than the particular way it is coded. So a patent on an algorithm might be infringed by any implementation of this algorithm, not just by a copy of the patent holder's original program.
- **Copyright** treats a program as a document or literary work. Copyright would be infringed by making a copy of the program itself but it is not clear that a re-implementation of the program design, perhaps in a different programming language, would still be an infringement.

In law and in international treaties copyright has covered the way ideas are expressed rather than the ideas themselves, but this distinction is not always clear when it comes to computer programs. In fact, copyright and patenting may both be appropriate ways to protect different aspects of a programmer's work. However, the idea of patenting software is controversial, as many IT professionals believe that algorithms should not be anyone's private property.

For the purposes of this course you should be aware that there are different ways of protecting IP rights in computer software and you should be able to explain the differences in general terms. However, you are not studying for a law degree and will not be expected to say which form of protection is more suitable in particular cases.

6.9.1 Copyright in computer software

Under TRIPS (Article 10) and the WCT (Article 4) computer programs are classed as literary works. Copyright protection extends to the design material for a computer program and any documentation provided with the program.

The major problem with copyright protection for computer programs, is that the ideas and algorithms contained in, and implemented by the program are not protected. The law and international treaty do not prohibit 'reverse engineering', analysing a system in order to understand how it has been composed or constructed, provided that the product has been legitimately sourced. Reverse engineering does not involve copying, so it is not prohibited. In the USA it may be allowed even if the intention is to produce a similar product, whereas the EU Article 6 of the EU's Computer Programs Directive

(originally issued in 1993, and last updated in 2009) restricts reverse engineering ('decompiling') to the purpose of interoperability with another program, and forbids the information gained being used to produce a 'substantially similar' program. See: <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31991L0250:EN:HTML>

6.9.2 Can software be patented?

The summary of the TRIPS Agreement states that computer programs 'shall be protected as literary works under the Berne Convention (1971)', see: www.wto.org/english/tratop_e/trips_e/intel2_e.htm However, under patents, the summary states that Member countries should 'make patents available for any inventions, whether products or processes, in all fields of technology without discrimination, subject to the normal tests of novelty, inventiveness and industrial applicability'. Three possible exceptions are then listed:

1. Inventions damaging of public order or morality; including inventions dangerous to life or to the environment.
2. Diagnostic, therapeutic and surgical methods.
3. Plants and animals.

Since computer programs do not come under any of these exceptions, it is arguable that they could be patented.

WIPO also has a web page about patenting software, which makes it clear that software may be patentable in certain jurisdictions:

To be eligible for patent protection, an invention must meet several criteria. Among those, five are most significant in determining patentability: (i) the invention must consist of patentable subject matter; (ii) the invention must be capable of industrial application (or, in certain countries, be useful); (iii) it must be new (novel); (iv) it must involve an inventive step (be non-obvious); and (v) the disclosure of the invention in the patent application must meet certain formal and substantive standards. Since patent law is applicable to inventions in any field of technology without discrimination, to be patentable, software-related inventions and business method-related inventions must also comply with those requirements.

In connection with software-related innovation, particular attention should be paid to the requirements concerning patentable subject matter and inventive step (non-obviousness). Firstly, a patent is granted for an "invention", which may be described, in general, as a solution to a technical problem. So far, there is no international definition of "invention", and indeed, each national law would give you a different answer to the question as to which subject matter falls under the term patentable "invention". In many countries, "inventions" are required to have a technical character, or to provide a solution using laws of nature. Thus, mere economic theories, methods of doing business, mathematical methods or computer programs as such are not patentable "inventions". Since this requirement varies from one country to another, as explained further in TIP 4, you should pay special attention as to whether your software-related innovation is covered by patentable subject matter under the relevant patent law.

Secondly, in order to obtain a patent, an invention must not be obvious to a person skilled in the art having regard to the prior art. It is not enough that the claimed invention is new; namely, that it is different from what exists in the state of the art. But the difference between the claimed invention and the existing state of the art should be significant and essential to the invention. Therefore, it is most likely that it will not be possible to obtain a patent for a software-related innovation that simply replaces existing technical and physical solutions with the same solutions using software and a computer, insofar as such a replacement would be obvious to an average engineer in the relevant technical field.

See: www.wipo.int/sme/en/documents/software_patents_fulltext.html

The European Patent Convention (EPC) was signed in 1973, with the European Patent Office set up to administer it in 1977. At the time of writing, 38 European states (including Turkey) have signed the convention, including every member of the EU. The EPO exists in order to simplify the process of applying for a patent in Europe, and the Treaty harmonised patent standards and procedures in order to facilitate this.

The ECT defines a patent in Article 52 as follows:

European patents shall be granted for any inventions, in all fields of technology, provided that they are new, involve an inventive step and are susceptible of industrial application.

See: www.epo.org/law-practice/legal-texts/html/epc/2016/e/ar52.html

Article 52 then has a list of exemptions – items that will not be patentable; one of these is ‘programs for computers’. While this seems to be clear, in fact the Article goes on to say that the exclusion from patentability applies ‘only to the extent to which a European patent application or European patent relates to such subject-matter or activities as such’. This has been interpreted to mean that software can be patentable if it has novel aspects that make a significant technical contribution.

In general, patenting software might be possible, but any application will almost certainly need significant legal advice, increasing the cost, particularly if an application is refused and an appeal is possible in law.

The situation is somewhat different in some other countries, including the USA, where patents are much more readily granted for computer software by comparison with the UK and EU.

6.9.3 Protecting software as a trade secret

In their FAQ section, WIPO explains that:

In general, computer programs are protected under copyright as literary works. [...]

However, according to a well-established principle, copyright protection extends only to expressions, not to ideas, procedures, methods of operation, or mathematical concepts as such. Thus many companies protect the object code of computer programs by copyright, while the source code is kept as a trade secret.

See: www.wipo.int/patents/en/faq_patents.html

Note that the benefit of a trade secret is that its lifetime is not limited as a patent is; a trade secret lasts for as long as the conditions outlined in Section 6.5.8 apply. Trade secret status does not have to be applied for hence there are no costly legal fees in establishing a trade secret, and no delays due to the

bureaucratic nature of the application process. In addition, trade secret status does not have to be applied for in every country that the business operates in.

6.9.4 Patenting software

In general, patents are said provide economic benefits, directly in terms of gaining revenue from the patented item, but also in terms of incentives to create and research, fostering further innovation. A strong patent system is claimed to give businesses confidence, leading to the sort of risk taking that fosters growth. You should be aware that patenting software is complex, the law is capable of interpretation, and the idea itself is the subject of much debate. In many countries patenting software is possibly, but difficult.

In the USA the courts are much more willing to give patents for software, and this has led organisations such as the Electronic Freedom Foundation (EFF) to campaign against software patents. Among other things the EFF claims that software patents do not work well as the US Patent Office accepts patents that use 'vague' or 'broad' language. Not having to precisely specify an invention can lead to the same idea being patented more than once if it is described in different terms, and to patents being issued for obvious extensions of existing technology. This encourages patent trolls, and makes it hard for US developers to write new software that does not infringe on someone's patent. The EFF, in its annual report for 2017, focuses on the case of a woman who ran a web page as a hobby, being sued for infringing US Patent No. 8,209,618, a patent granted for the idea of online competitions. See: www.eff.org/files/annual-report/2017/index.html#InnovationAlice

However, issues with the US Patent Office are specific to the USA, where thousands of software patents have been issued. Other arguments against software patenting include:

- Software is ultimately mathematics, and mathematics cannot be patented.
- Copyright protection is sufficient for software.
- Patenting reduces the sharing of new ideas, as companies and individuals do not wish to share their ideas in advance of patenting them. Without such sharing innovation is stifled.
- Patenting takes a long time, and in a field as fast moving as computing, the software may be redundant by the time the patent is approved.
- Allowing software patents could lead to defensive patenting, seen in the USA, where tech companies proactively patent with the intention of preventing other companies from suing them. Patent litigation is expensive, and only very large companies can usually afford it. Small and medium sized enterprises will normally settle in order to reduce costs, whatever the rights and wrongs of their case.
- Patents would reduce interoperability, a fundamental standard for the internet.
- Costs of patent litigation could easily exceed the economic benefits of patents.
- Patent offices do not have a good standard for software patents, meaning that officials may not be able to understand technical details from the application, and applications may be approved that should not be.
- Patent applications take time and money, and are a drain on the resources of small and medium-sized enterprises.

Many developers and computer scientists simply feel that patenting software goes against the fundamental development principles of the internet (shared standards, interoperability) and that innovation and progress depends upon the free exchange of ideas. The open source software movement promotes the sharing and collaborative development of software with developers publishing their source code and others free to update it.

See: <https://opensource.org/> for more information.

Activity 6.2

Explain why companies may choose to protect their proprietary software as a trade secret, rather than relying on copyright law. Give the advantages of protecting software as a trade secret over copyright and patent law.

6.10 ISPs and the law

ISPs are not publishers, in the eyes of UK and EU law, and so they have certain defences against accusations of illegality, that they would not have if they were. The EU's Directive 2000/31/EC, known as the e-Commerce Directive, first addressed the need for a market framework for online services:

The purpose of the Directive is to remove obstacles to cross-border online services in the European Union and provide legal certainty to business and citizens in cross-border online transactions.

The Electronic Commerce Directive (e-Commerce Directive 2000/31/EC), adopted in 2000, sets up an Internal Market framework for electronic commerce, which provides legal certainty for business and consumers alike.

The Directive establishes harmonised rules on issues such as the transparency and information requirements for online service providers, commercial communications, electronic contracts and limitations of liability of intermediary service providers.

See: <https://ec.europa.eu/digital-single-market/en/e-commerce-directive>

This directive complies with the WCT, which recognised that countries would need to make new laws to allow for technological changes in the digital age, and that 'limitations and exceptions, as established in national law in compliance with the Berne Convention, may be extended to the digital environment'; see: www.wipo.int/treaties/en/ip/wct/summary_wct.html

Below is a summary of the three protections given by the e-Commerce directive. While the directive is concerned with copyright, the protections listed below do not just apply to copyright, but to other laws, such as those relating to obscenity and defamation.

- **Mere conduit**

Under Article 12, ISPs are considered to be the 'mere conduit' of information transmitted and viewed, provided that they do not initiate the transmission, do not target it at any particular receiver, and do not select or change any of the information transmitted. This means that they are not liable for any breaches of law in the data. This article also gives ISPs the right to make copies of data provided that they are automatic, temporary and needed for the purpose of transmission. Caching, which means longer term storage of web pages, is dealt with in Article 13.

- **Hosting**

Article 14 concerns hosting, and notes that 'the service provider is not liable for the information stored at the request of a recipient of the service' provided that the provider is unaware of any illegality, but on being made aware of it acts quickly to remove the data or deny access to it.

- **Caching**

Article 13 applies to caching, defined as 'information provided by a recipient of the service' that is stored in a way that is 'automatic, intermediate and temporary' and for the 'sole purpose of making more efficient the information's onward transmission to other recipients of the service upon their request'. This is legal provided that certain conditions are met. These conditions are largely to do with making sure of the integrity of the information by such things as not modifying it, updating it appropriately and removing from the cache data that has been removed from the network for any reason, and also with complying with any restrictions or conditions for access to the information. This allows ISPs to make copies that are not needed for technological reasons by, for example, caching copies of popular websites in order to increase access time for users.

In addition service providers in the EU have no obligation to monitor material for illegality. However, the above protections for regarding hosting and the **mere conduit** defence only apply if service providers comply quickly with requests to remove illegal material that is brought to their attention via take down notices.

6.10.1 ISPs and copyright

Copying music, video, images and text from the internet is illegal under copyright laws in most countries, including the UK and the EU. Graphic works or photographs used to create screen images would be treated as artistic works and animated graphics are classed as films. Email messages are also copyright and in theory cannot be forwarded without the sender's permission. The layout and design of web pages themselves may also be protected by copyright. Despite copyright law applying to the internet, it would be expensive for copyright holders to take civil enforcement action against every infringement, which is why the internet can seem like a copyright-free zone, when it is not.

Any computer processing of documents, images or other objects may include the creation of transient copies in cache memory, a process that occurs outside the user's direct control. Browsing a document through the web in fact causes multiple copies to be made as the document travels through various routers until it is displayed in the browser window, with a copy likely to be kept in the browser's cache for a limited period. ISPs are given explicit protection for caching under the e-Commerce Directive, and for other copies made, provided that they are automatic, temporary and needed for the purpose of transmission.

Thus while merely viewing a web document on a screen may not be a breach of copyright, because of the legal protections afforded to ISPs, it could still be an infringement for the user to print off a copy or save it.

6.10.2 ISPs and defamation

Defamation means publishing a statement that harms someone's reputation, or is likely to harm it. 'Publishing' means simply that the statement is communicated to at least one third party. A defamatory statement that is untrue falls under the law of either libel or slander.

Libel is defamation made in a permanent form (for example, written or printed).

Slander is defamation made in a temporary form (e.g. spoken).

Defamation via electronic communication is generally classed as libel:

- email
- social media
- webpages.

Different countries have different policies on 'innocent dissemination', when a libellous statement may be posted on a website or bulletin board without the prior knowledge or active involvement of the ISP or website owner. In the USA, under the Communications Decency Act 1995, only the original poster of the libel can be prosecuted.

Internet service providers in the EU have the defence of **mere conduit**, for private communications (e.g. emails), and of **hosting** for material posted online that is illegal in some way (defamatory, infringes copyright).

Employers may be liable for any defamatory content in email messages sent by employees.

Jurisdiction: jurisdiction in cases of libel concerning the internet can be complicated. In the UK it once was the case that a libel action could be brought against an individual or organisation if the libellous material could be downloaded in the UK. This has been amended by the Defamation Act of 2013 to say that if the individual is not a UK or EU resident, then a UK court does not have jurisdiction unless 'the court is satisfied that, of all the places in which the statement complained of has been published, England and Wales is clearly the most appropriate place in which to bring an action in respect of the statement.'

The Defamation Act 2013 was brought in partly to address that the law of defamation in the UK was being interpreted in light of the Human Rights Act of 1998 (based on the European Convention on Human Rights, drafted by the Council of Europe), which stated that other laws had to be interpreted in the light of the new law. This led, for example, to a judgement of the Law Lords in the UK in 2005, who held that pursuing a claim for defamation in the case of a hyperlink followed and read by just five people (*Jameel v Dow Jones & Co Inc*), was an abuse of process since the damage done was considered to be trivial even if the claim was technically actionable. The Law Lords considered that Section 6 of the Human Rights Act required them to consider a 'proper balance between the Article 10 right of freedom of expression and the protection of individual reputation' and that:

Keeping a proper balance [...] must [...] require the court to bring to a stop as an abuse of process defamation proceedings that are not serving the legitimate purpose of protecting the claimant's reputation, which includes compensating the claimant only if that reputation has been unlawfully damaged [...] on the premise that there have only been the five individual publications within this jurisdiction, we would dismiss this action as an abuse of process.

The Defamation Act of 2013 changed the law to give more weight to freedom of expression, following on from the Human Rights Act 1998. The law tries to balance freedom of expression with protection of reputation, and codifies case law established by interpreting the law in terms of the Human Rights Act, such as the case quoted above (addressed by changing that law so that a statement is not defamatory unless it causes 'serious harm').

In terms of the internet, as well as changing the law on jurisdiction for defendants outside the EU, the Act also introduced:

- **Anonymous posts (Section 5):** this section is entirely directed at 'operators of web sites'. It states that 'It is a defence for the operator to show that it was not the operator who posted the statement on the website'; that is, a website operator cannot be sued for libel if they are not the originator of the defamatory material. This defence only holds if they have complied with any take down notice. However, in the case of anonymous material, defined as content where it is impossible to identify the originator of the material, then the website operator may be sued for libel, and the defence does not apply.
- **The single publication rule** means that if a libel that is 'substantially the same' is published more than once, then the date of the first publication counts for time limiting the right to take legal action. This should apply to internet archives, caches and reposting. (Section 8)
- **Defamation actions** cannot be taken against secondary publishers 'unless the court is satisfied that it is not reasonably practicable for an action to be brought against the author, editor or publisher'. This clause was presented as a defence for booksellers, but has obvious applications to posting and reposting on the internet. (Section 10)
- **Taking down defamatory statements:** Once a case has been decided, the court can order the 'operator of a website on which the defamatory statement is posted to remove the statement'. (Section 13)

6.10.3 ISPs and data protection

While ISPs and other web service providers have the same responsibilities as any other organisation for protecting the data of their customers, they also have some particular responsibilities to guard their data, and to inform customers of any risks. See: <https://ico.org.uk/for-organisations/guide-to-pecr/communications-networks-and-services/security-of-services/> for more information about data protection law and the particular responsibilities of web service providers. Note that the UK is largely implementing EU directives and following EU regulations. See Section 6.13 for more on data protection law in general.

6.10.4 Surveillance and monitoring

Article 15 of the e-Commerce Directive established that ISPs had no general obligation to monitor users. When providing world wide web services covered by the Directive, service providers have no obligation to monitor, or at least member states cannot impose on them a duty to actively monitor information for illegal activity. Member states may require service providers to respond to any illegality brought to their attention, and/or oblige service providers to provide identifying information of particular clients when so requested by public authorities.

ISPs who did monitor data would be likely to be considered as publishers under the law, putting them at much greater risk for breaching the law, including that of copyright.

Article 15 of the Directive may be the most significant in terms of stopping member states from policing and controlling the flow of information on the Internet. It has never been implemented into UK law, however, and would require special legislation to be enacted to implement it once the UK leaves the EU.

In fact, within the UK civil liberties organisations and the government have been involved in a long-running dispute regarding the Investigatory Powers Act of 2016, which requires, among other things, that ISPs store their user's browsing history for a year and gives the security services the legal right to hack into citizens devices and networks. In addition, in 2017 *The Register* reported on regulations proposed under the 2016 Act that would 'provide the government with the legal authority to monitor anyone in the UK in real time, as well as effectively make strong and unbreakable encryption illegal.' See: www.theregister.co.uk/2017/05/04/uk_bulk_surveillance_powers_draft/

The Act was successfully fought in the courts on the grounds that it conflicted with UK law, and the government was required to issue new regulations to amend this. At the time of writing it is now UK law, with legal challenges ongoing, see: www.theguardian.com/technology/2019/jun/17/liberty-mounts-latest-court-challenge-to-snoopers-charter-mi5-gchq

The concern many organisations have, in opposing the law, is that UK law could become a model for illiberal regimes around the world. The law relating to ISPs, monitoring and surveillance is clearly an area of active change, and you will not be expected to be aware of the details of the law in the UK, EU, or elsewhere.

6.11 An historical note: The Cybercrime Convention

Gillespie (2015, Chapter 1, section *Cybercrime Convention*) states that the **Council of Europe Convention on Cybercrime** is 'an important international instrument' and was the first international treaty to consider cybercrime, and 'arguably retains its position as the leading international instrument tackling cybercrime'. The Cybercrime Convention was signed in Budapest in 2001, so is also known as the Budapest Convention. The Council of Europe intended it to be applied worldwide, and so the USA, Japan and other countries were given observer status as it was drafted. At the time of writing 61 countries had ratified the treaty or equivalent, with four having signed but not yet ratified it. See: <https://www.coe.int/en/web/conventions/full-list/-/conventions/treaty/185/signatures>

All members of the Council of Europe have signed the treaty, with the exceptions of San Marino, and more significantly, Russia. At the time of writing all 28 members of the EU were also members of the Council of Europe and had signed the Cybercrime Convention, with all except Ireland ratifying. The USA, Canada, Chile, Israel, Japan, Australia, the Philippines and Argentina are among non-member states signing and ratifying the Convention.

Gillespie (2015, Chapter 1, section *The Convention today*) notes that:

the Convention has been used as a model for other international instruments, most notably by the Commonwealth (formerly known as the 'British Commonwealth') who developed model cybercrime laws that were based on the Cybercrime Convention.

This dramatically increases its reach as the Commonwealth countries include a number of African, Asian and Caribbean countries. Other geo-political groupings have taken the Cybercrime Convention and then built upon it. A good example of this is the African Union whose 'Draft African Union Convention on the Establishment of a Credible Legal Framework for Cybersecurity in Africa' has the potential to become one of the most comprehensive international documents.

Gillespie also notes in this section that Russia and China both 'want more control over the governance of the internet' and would like a new

international treaty to be drafted by the UN. Gillespie notes that ‘this dispute is likely to continue’ and suggests that ‘if some geo-political areas do refuse to become involved and extend this to countries within their sphere of influence then it may be difficult to get true global reach, which may cause complications in the fight against cybercrime’.

6.11.1 The Cybercrime Convention and substantive criminal law

Countries that have ratified (or equivalent) the CyberCrime Convention will have laws that criminalise computer hacking and writing and deploying viruses and other malicious code. Gillespie (2015, Chapter 1, section *Measures to be taken at the domestic level*) notes that Chapter II of the Convention contains Articles 2–13, which specify crimes that are to be enacted into criminal law by contracting parties, with the rest of the Chapter dealing with procedures (e.g. for evidence gathering) and jurisdiction issues.

Articles 2–13 are summarised briefly below:

Title 1 – Offences against the confidentiality, integrity and availability of computer data and systems

Article 2: **Illegal access** to ‘the whole or any part of a computer system without right’

Article 3: **Illegal interception** ‘when committed intentionally, the interception without right, made by technical means, of non-public transmissions of computer data to, from or within a computer system.’

Article 4: **Data interference** when done intentionally, ‘the damaging, deletion, deterioration, alteration or suppression of computer data without right.’

Article 5: **System interference** when done intentionally, ‘the serious hindering without right of the functioning of a computer system by inputting, transmitting, damaging, deleting, deteriorating, altering or suppressing computer data.’

Article 6: **Misuse of devices** making available, procuring or possessing, such things as computers, computer programs, passwords and access codes ‘with intent that it be used for the purpose of committing any of the offences established in Articles 2 through 5’.

Title 2 – Computer-related offences

Article 7: **Computer-related forgery** ‘when committed intentionally and without right, the input, alteration, deletion, or suppression of computer data, resulting in inauthentic data with the intent that it be considered or acted upon for legal purposes as if it were authentic.’

Article 8: **Computer-related fraud** ‘when committed intentionally and without right, the causing of a loss of property to another person’ by ‘any input, alteration, deletion or suppression of computer data’ and/or an ‘interference’ with a computer system with the intention of fraudulently ‘procuring, without right, an economic benefit for oneself or for another person’.

Title 3 – Content-related offences

Article 9: **Child pornography**: the article defines child pornography and offers a list of actions to be criminalised.

Title 4 – Offences related to infringements of copyright and related rights

Article 10: **Offences related to infringements of copyright and related rights.** This article specifies that infringing copyright 'wilfully, on a commercial scale and by means of a computer system' must be a criminal offence unless other remedies exist. Moral rights under the Berne Convention, TRIPS or the WCT are excluded.

Title 5 – Ancillary liability and sanctions

Article 11: **Attempt and aiding or abetting:**

- 'when committed intentionally, aiding or abetting the commission of any of the offences' from Articles 2–10 'with intent that such offence be committed'.
- With intent attempting 'to commit any of the offences established in accordance with Articles 3 through 5, 7, 8, and 9.1.a and c of this Convention'.

Article 12: **Corporate liability** does not define new crimes, but is about how a corporation can be held to account for crimes committed by a person acting under their authority.

Article 13: **Sanctions and measures** against the person for articles 2–11 should be 'effective, proportionate and dissuasive sanctions, which include deprivation of liberty' and sanctions 'in accordance with Article 12 shall be subject to effective, proportionate and dissuasive criminal or non-criminal sanctions or measures, including monetary sanctions'.

See: <https://www.coe.int/en/web/conventions/full-list/-/conventions/rms/0900001680081561>

6.12 Computer misuse

Computer misuse concerns the criminalisation of hacking, virus writing and using a computer to commit further crimes such as fraud or theft. **The Council of Europe Convention on Cybercrime** was the first international treaty relating to cybercrime. It was signed in Budapest in 2001, so is also known as the Budapest Convention.

6.12.1 Definition of cybercrime

Relevant legal issues come under two main headings:

1. 'Traditional' legal topics that may require new interpretations or new legislation to take account of new technology: this includes issues of copyright, defamation, obscenity, fraud, blackmail and theft.
2. New areas of law addressing issues that are unique to a computer environment such as data protection, hacking and viruses, use of the internet in the workplace, intellectual property rights in computer databases.

An alternative distinction is between crimes committed using a computer (the computer as a tool) – and crimes committed on a computer (the computer as the target).

Gillespie (2015, Chapter 1, section *Classifying cybercrime*) gives the following four point definition of cybercrime:

1. **Crimes against computers.** That is to say, crimes that were not in existence before the internet and where the computer is the target of the crime

2. **Crimes against property.** This is where the object of the cybercrime is to obtain property (be that financial or intellectual) from another.
3. **Crimes involving illicit content.** This is where the crime relates to the posting, hosting or accessing of objectionable content.
4. **Crimes against the person.** This is where the technology is used as a 'weapon' against an individual, with the potential of causing harm to that person.

Making a broad comparison to the above 4 definitions of cybercrime to Articles 2–13 of the Cybercrime Convention **crimes against computers** are covered by title 1 of the Convention (Articles 2–6), **crimes against property** by title 2 (Articles 7 and 8) and **crimes involving illicit content** by title 3 (Article 9) and title 4 (Article 10). The Convention does not cover **crimes against the person**, for example, 'cyberstalking' or 'cyberbullying'; neither of which has a clear agreed definition at the time of writing, but is understood to mean using the internet to cause fear and distress to a target by various means, whether the target be a person or an organisation.

The crimes of interest to this course are the title 1 crimes, Articles 2–6, since they concern hacking, or unauthorised access to a computer or computer system. Article 3, illegal interception, is arguably different since the interception of data has historically been understood to be espionage, rather than hacking. In fact interception was covered in the UK in the **Regulation of Investigatory Powers Act 2000**, and the introductory text to the act makes it clear that interception and the security services are closely linked:

An Act to make provision for and about the interception of communications, the acquisition and disclosure of data relating to communications, the carrying out of surveillance, the use of covert human intelligence sources and the acquisition of the means by which electronic data protected by encryption or passwords may be decrypted or accessed; to provide for Commissioners and a tribunal with functions and jurisdiction in relation to those matters, to entries on and interferences with property or with wireless telegraphy and to the carrying out of their functions by the Security Service, the Secret Intelligence Service and the Government Communications Headquarters; and for connected purposes.

See: www.legislation.gov.uk/ukpga/2000/23/introduction

6.12.2 Additional Protocols to the Cybercrime Convention

In 2003 an **Additional Protocol to the Convention on Cybercrime**, concerning the criminalisation of acts of a racist and xenophobic nature committed through computer systems was opened for signing. Gillespie (2015, Chapter 1, heading *Protocol to the Convention on Cybercrime*) notes that this was not included in the original convention because some countries had concerns about free speech. Therefore it was published as an additional Protocol that countries could sign or not. In fact the UK has not signed the Protocol, but on the grounds that the UK already has laws governing hate speech and racially motivated crime.

In 2017 the Council of Europe agreed to consider a second Additional Protocol, concerning access for law enforcement to data stored in other countries' jurisdictions, see: www.coe.int/en/web/cybercrime/-/cybercrime-towards-a-protocol-on-evidence-in-the-cloud. Also see: www.eff.org/deeplinks/2017/09/cybercrime-conventions-new-protocol-needs-uphold-human-rights

6.12.3 The Computer Misuse Act 1990 (CMA) as amended by the Police and Justice Act 2006 (PJA) and the Serious Crimes Act 2015 (the 2015 Act)

The Computer Misuse Act (CMA) was introduced in the UK in 1990, 11 years before the Cybercrime Convention was opened for signing, and 21 years before the UK signed it in 2011. The early genesis of the Act was the result of actions by two journalists who, in 1985, hacked into systems owned by British Telecom (now the BT Group) and left a message in the Duke of Edinburgh's message box. At the time there was rising concern about computer hacking, even if most people only had a very vague idea of what hacking was. Although the journalists were attempting to highlight security flaws and caused no damage, they were prosecuted under the Forgery and Counterfeiting Act 1981, a prosecution that ultimately failed, and led to the introduction of new law to criminalise hacking, the CMA. See: www.theregister.co.uk/2015/03/26/prestel_hack_anniversary_prince_philip_computer_misuse for a full account of the story.

The CMA originally created three categories of offence, very much aimed at criminalising hacking: (1) unauthorised access to computer material; (2) unauthorised access with intent to commit a further offence (or to facilitate the commission of a further offence); and (3) unauthorised modifications of programs or data. When the Act was passed in 1990 the maximum sentence for the first crime was six months in prison, and for the second two, considered to be more serious offences, it was five years' imprisonment. Since then the CMA has been amended in 2006 by the Police and Justice Act (PJA) to: (1) add to the first offence that of modifying a computer either to allow access at a later date or to allow another person to make unauthorised access, for example, by placing a trojan on the machine; (2) make significant changes to the third offence to make it clear that denial of service attacks are included; and (3) to add a fourth offence of making, supplying or obtaining articles for use in computer misuse offences.

With respect to the third point above, Gillespie (2015, Chapter 3, Section 3) argues that the CMA 1990 did cover denial of service attacks, but that there was some doubt in law that the amendments in the PJA removed. The PJA also increased the maximum penalty to 10 years imprisonment.

The fourth offence added by the PJA was that of making, adapting, supplying or offering to supply items ('hacker tools') intended to be used in an offence under the CMA, or making, adapting, supplying or offering to supply items believing that the items would be so used; this implemented Article 6 of the Cybercrime Convention. However, Article 6 suggested that procuring hacker tools could be an offence, but since most such tools are dual use and may be used (and thus procured) by security professionals and system administrators this optional part of the Article was not implemented.

The CMA was amended again by the Serious Crimes Act 2015 to include the offence of procuring items to be used in an offence. This was to update the CMA to comply with EU Directive 2013/40/EU on attacks against information systems, a Directive which built on the Cybercrime Convention, but unlike the Convention required that procurement of hacker tools be criminalised. The CMA was believed to comply with the Directive in other respects except for some jurisdiction issues, also amended in the CMA by the 2015 Act.

The most significant change introduced to the CMA by the 2015 Act, was that of 'Unauthorised acts causing, or creating risk of, serious damage'. This new section introduced the term of life imprisonment if such acts resulted, or could result, in 'serious damage to human welfare' (including loss of life) or 'serious damage to national security'. This section was introduced as it was

felt by the UK government that there was a gap in the law where a cyber attack could cause serious damage to national infrastructure. Clearly this is a step change from the original CMA with its maximum sentences of five years' imprisonment, and reflects the growth of the internet and its intertwining with critical national infrastructure in a way unforeseen in 1990, and, arguably, a certain amount of paranoia.

6.12.4 The five categories of offence under the CMA 1990 as amended

The UK Computer Misuse Act (as amended) has five categories of offence:

1. Section 1: **Unauthorised access to computer material.** Intentional access to a computer, knowing that such access is unauthorised, or actions enabling access at a later date.
2. Section 2: **Unauthorised access with intent to commit or facilitate the commission of further offences:** the further offence may be one that is already established as a crime in other UK laws; for example, accessing personal files or company records in order to commit fraud or blackmail.
3. Section 3: **Unauthorised acts with intent to impair, or with recklessness as to impairing, operation of computer, etc.** Section 3 has been used to convict authors of malicious code such as viruses, and for denial of service, and distributed denial of service attacks.
4. Section 3ZA: **Unauthorised acts causing, or creating risk of, serious damage.** Damage comprises:
 - (a) damage to human welfare in any place;
 - (b) damage to the environment of any place;
 - (c) damage to the economy of any country; or
 - (d) damage to the national security of any country.

The maximum sentence is 14 years, except where damage or potential damage is: (1) to human welfare that includes loss of life, illness or injury; or (2) 'serious' damage to national security. In such cases, the sentence is life imprisonment.

This section would apply not just to authors of malicious code, or those launching denial of service attacks, but also to attacks where national infrastructure is targeted through the computer systems that support it.

5. Section 3A: **Making, adapting, supplying, offering to supply or procuring articles for use in offences under Sections 1, 3 or 3ZA.**

6.12.5 What the CMA does not cover

Article 3 of the Cybercrime Convention is the sole article from title 1, **offences against computer data and systems**, which the CMA does not cover. However, this was covered in the **Regulation of Investigatory Powers Act of 2000**.

The CMA does not cover crimes against the person such as cyberbullying (using the internet to cause fear and distress to a target by various means, whether the target be a person or an organisation), just as the Cybercrime Convention does not. This is not to suggest that cyberbullying, cyberstalking or cyber harassment is legal, although as Gillespie notes in Chapter 11, it is not covered by any of the international conventions on cybercrime. Gillespie notes that UK law does cover computer crimes against the person, through provisions in the Malicious Communications Act 1988 and the Communications Act 2003. These two acts both define offences carried out by means of 'electronic communication'. In addition the Protection from Harassment Act 1997 addresses harassment in general.

Click farms are defined by <https://whatis.techtarget.com/definition/click-farm> as 'a business that pays employees to click on website elements to artificially boost the status of a client's website or a product'. The site notes that social media likes (e.g. Facebook, Instagram) are a particular target. In fact try searching 'instagram likes' to find companies offering to sell likes to you. Click farms are places where organised click fraud is committed. Click fraud is defined by <https://en.oxforddictionaries.com/> to be:

The practice of repeatedly clicking on an advertisement hosted on a website with the intention of generating revenue for the host website or draining revenue from the advertiser.

This exploits the fact that many advertisements on the web are only paid for when a user clicks on the advert and is then sent to the advertiser's web page.

Click fraud would seem to be covered by Article 8 of the Cybercrime Convention but is not covered explicitly by the CMA, just as cyberbullying is not, because the CMA focuses on unauthorised access, and further actions taken once unauthorised access has been achieved. This is because the intention of the CMA is to criminalise hacking and malicious code. Click fraud and cyber harassment or cyberbullying do not need unauthorised access, and are not malicious code.

Click fraud links

- www.theregister.co.uk/2018/06/29/ad_fraud_bad/ – looks at the law on click fraud, with reference to the United States.
- www.actionfraud.police.uk/a-z-of-fraud/click-fraud – the UK's national fraud reporting service encourages the public to report click fraud.
- www.theukdomain.uk/what-is-ad-fraud-and-what-should-smes-be-doing-about-it/ – contains surprising statistics about the extent of click fraud.

Activity 6.3

Answer the following questions.

- (i) What are the three kinds of legal defence available to ISPs regarding the content of webpages they allow their customers to access?
 1. Hosting; caching; mere conduit
 2. Hosting, network availability; data privacy
 3. Hosting; transferable data, the telephone company defence
 4. None of the above
- (ii) What is a **take down request**?
 1. A law enforcement request to prevent an individual's web and internet access
 2. An official request to remove illegal web content, or to block access to it
 3. An official request for the web browsing histories of an ISP's customers
 4. None of the above
- (iii) Which one of the following is true?
 1. **Libel** is a form of defamation, while **slander** is not
 2. The target of a libel can take court action, while the target of slander cannot
 3. **Slander** and **libel** are both forms of defamation
 4. None of the above

(iv) Which one of the following is **false**?

1. The Computer Misuse Act 1990 as amended, has three categories of offence
2. The Computer Misuse Act 1990 was brought in by the UK government to criminalise hacking
3. The Computer Misuse Act 1990 has been amended to include making and supplying 'hacker tools', as these were not included in the 1990 act
4. None of the above

6.13 Data protection and the General Data Protection Regulation (GDPR)

EU countries, including the UK, had data protection law based on the EU Data Protection Directive, 95/46/EC. Each country in the EU had implemented the Directive into law, in the UK with the Data Protection Act 1998. Each implementation was different, given the variability allowed by the Directive, which specified minimum standards for data protection. When the EU issues a Regulation, rather than a Directive, it has the force of law in member countries and no legislation to implement it is necessary. Hence, with the coming into force of the General Data Protection Regulation (GDPR) on 25 May 2018, all EU countries have the same data protection law regarding the personal information of their citizens. Not only that, but as *Wired* notes:

Europe is now covered by the world's strongest data protection rules. The mutually agreed General Data Protection Regulation (GDPR) [...] was designed to modernise laws that protect the personal information of individuals.

See: www.wired.co.uk/article/what-is-gdpr-uk-eu-legislation-compliance-summary-fines-2018

Data that is covered by the GDPR is data that could be used to identify an individual that is processed 'wholly or partly by automated means' and data that is processed 'other than by automated means' that 'forms part of a filing system'. Hence the Regulation applies to electronic data, and to hard copy data provided that it is stored in a structured way and can be retrieved according to specific criteria. The Regulation does not apply to data processed for the purposes of law enforcement or public security.

6.13.1 Application of the GDPR outside the EU

The GDPR applies to the processing of the personal data of EU citizens, including when that data is processed outside of the EU. This processing may be done by EU organisations processing data at centres outside the EU, or EU organisations transferring data to a third party to process. However, in two circumstances the Regulation applies to organisations that are established outside the EU. The first is when an organisation offers goods and services to EU data subjects. Since this includes goods and services where no payment is required, this would cover social media companies such as Facebook. The second is when an organisation outside the EU is monitoring the behaviour of EU data subjects, provided that behaviour takes place inside the EU. This would apply to companies that track the browsing habits of individuals in order to target advertising at them.

6.13.2 The GDPR in the UK

In the UK the GDPR applies while the UK remains a member of the EU, and:

When the UK leaves the EU, the GDPR will be incorporated into the UK's domestic law under the European Union (Withdrawal) Bill, currently before Parliament.

See: www.legislation.gov.uk/ukpga/2018/12/notes/division/2/index.htm

In addition the Data Processing Act 2018 repeals the 1998 Data Protection Act, and 'applies a broadly equivalent regime to certain types of processing to which the GDPR does not apply' which includes processing for law enforcement purposes and by the intelligence services. The Act also lists cases where certain GDPR provisions do not apply, based on provisions in the GDPR (see later, **Restrictions and derogations**).

In the UK data protection, including the monitoring and enforcement of the Act and of the GDPR is overseen by the Information Commissioner's Office (ICO): <https://ico.org.uk/>

So at the time of writing, the GDPR and the Data Protection Act 2018 together comprise the UK's Data Protection law.

6.13.3 Key definitions under the GDPR

Personal data: The GDPR defines personal data as follows:

'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person

See: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:02016R0679-20160504&from=EN> for the text of the GDPR.

This definition specifically includes genetic data, and also covers biometric data, that is data about the 'physical, physiological or behavioural characteristics of a natural person' produced by a technical process, such as a retinal scan or gait analysis, that can be used to uniquely identify them, or to confirm identity. In addition an 'online identifier' could be an IP address or a username for an online account. Personal data includes data points that on their own could not identify an individual, but data controllers must consider whether, in combination with other data they have, the individual could be identified. Thus what is personal data for one data controller, may not be personal data to another.

Personal data can be both facts and opinions about the individual. Individuals can, and do, make data subject access requests to organisations to see emails that were written about them.

Data processing: is defined in the GDPR as 'any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available,

alignment or combination, restriction, erasure or destruction'. That seems to cover everything.

Data subject: an individual who is the subject of personal data. Note that the GDPR applies only to 'natural persons' (private individuals) and not to organisations such as companies or trade unions. Hence a data subject is a person who can be uniquely identified by the data.

Data subject access request: a request by the data subject to see their personal data. This can be made verbally or in writing. The data controller has one month to respond.

Data controller: the person or organisation that decides how, and for what purposes, the data shall be processed. They are responsible for complying with data protection law.

Data processor: the person or organisation that processes the data on behalf of the controller.

Data Protection Officer (DPO): public bodies and organisations that: (1) perform data processing on a large scale; or (2) who process on a large scale 'special category' data (see later) or data on criminal convictions; are required by the GDPR to appoint a DPO. The DPO will be the individual within the organisation who monitors and advises on compliance with data protection law, and acts as a point of contact for the supervisory authority.

Supervisory authority: Article 51 of the GDPR states that 'Each Member State shall provide for one or more independent public authorities to be responsible for monitoring the application of this Regulation'. In the UK the Information Commissioner takes this role. The supervisory authority is charged with monitoring and enforcing data protection laws, and has various powers, including the power to investigate, fine and prosecute data controllers.

Personal data breach: The data controller and the processor must have in place 'appropriate technical and organisational measures' to ensure the security of data being transmitted, processed and stored. A personal data breach 'means a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed.' The data controller must notify breaches to the supervisory authority within 72 hours, unless the breach is trivial; namely, is 'unlikely to result in a risk to the rights and freedoms of natural persons'.

6.13.4 The seven principles of data protection

The EU Data Protection Directive, 95/46/EC, laid out the principles of data protection on which EU member states based their laws. These have now been amended to the following seven key principles, which underpin the GDPR:

1. **Lawfulness:** Personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject.
2. **Purpose limitation:** Personal data shall be collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes. Further processing for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes shall (subject to certain safeguards), not be considered to be incompatible with the initial purposes.
3. **Data minimisation:** Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed.

4. **Accuracy:** Personal data shall be accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay.
5. **Storage limitation:** Personal data shall be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed; personal data may be stored for longer periods insofar as the personal data will be processed solely for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes.
6. **Integrity and confidentiality (security):** Personal data shall be processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures.
7. **Accountability:** The controller shall be responsible for, and be able to demonstrate, compliance with the principles.

6.13.5 Lawful data processing

Article 6(1) of the GDPR lists six points where data processing 'shall be lawful only if and to the extent that at least one of the following applies':

- (a) the data subject has given consent to the processing of his or her personal data for one or more specific purposes;
- (b) processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract;
- (c) processing is necessary for compliance with a legal obligation to which the controller is subject;
- (d) processing is necessary in order to protect the vital interests of the data subject or of another natural person;
- (e) processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller;
- (f) processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child.

Point (f) [...] shall not apply to processing carried out by public authorities in the performance of their tasks.

See: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:02016R0679-20160504&from=EN>

6.13.6 Rights of individuals with respect to personal data under the GDPR

'Data protection' is an inaccurate term: it is not the data that is protected as such, but the rights of individuals to whom the data relates. The GDPR sets out the rights of data subjects, which include new rights to **data portability**, and the **right to be forgotten**, which arose from a case decided by the European

Court of Justice in 2014. Other rights, already in existence because of earlier EU Directives, have been strengthened.

Under the GDPR data subjects have the following rights, described in Articles 12 to 22:

- **Right to information** (see Section 6.13.7)
- **Right of access** (see Section 6.13.8)
- **Right to rectification**: the right to have inaccurate information corrected in a timely manner
- **Right to erasure (the 'right to be forgotten')**: the data subject has a right to have their data erased under certain conditions, including unlawful processing, withdrawal of consent and that the data is no longer needed for the original purpose
- **Right to restriction**: data subjects can restrict the processing of their data under certain conditions such as that the data is inaccurate or the processing is unlawful
- **Right to data portability**: the data subject has the right to apply for their personal data and to receive it in a structured, commonly used and machine-readable format, provided that: (1) their data is being processed based on their consent, or processed because of a contract the data subject has chosen to enter into; and (2) their data is being processed automatically
- **Right to object**: the data subject can object to the processing of their data for a number of reasons, including that they can object at any time to data processing for direct marketing
- **Automated decision-making**: the right not to be subject to a decision based solely on automated processing, including profiling*, unless the data subject has specifically given permission, or it is necessary to enter into a contract with the data controller, or it is authorised in the law of the EU or the member state, with safeguards.

*Profiling is the automated processing of personal data that comes to conclusions about individuals on which decisions will be based, for example, credit scoring.

6.13.7 Right to information

Data subjects have the right to information about the use of their data, and their rights over it. Where data is collected directly from the data subject they must be given the following information at the time; where data is sourced from a third party the data subject must receive the following information in a timely way:

- the identity and contact details of the controller or their representative
- the contact details of the data protection officer, if appointed
- the purposes of the processing and the legal basis for it;
- the categories of personal data (only when data has not been collected directly from the data subject)
- where processing is necessary under Article 6(1)(f) the legitimate interests in question
- the recipients of the personal data (if any beyond the controller)
- if the data is to be processed in another country, or by an international organisation, the data subject is to be told of this and the safeguards in place

- the retention period or the criteria used to determine the retention period
- the existence of automated decision-making, including profiling, and information about the logic and consequences of such processing
- if the controller intends to process the data subject's personal data for a further purpose other than the original purpose, then, before the processing can take place, the data subject must be told the reason for the additional processing and given further information as appropriate, including information about their rights as a data subject.

The data subject also has the following rights.

- The right to request:
 - access to personal data
 - rectification or erasure of personal data
 - restriction of processing of personal data
 - right to object to processing of personal data
 - a copy of their data in a portable format.
- Where the data subject has given consent to the processing of their data, they have the right to withdraw consent at any time, and must be informed of this.
- The right to complain to a supervisory authority (in the UK, the Information Commissioner).

Where the data has not been sourced from the data subject, there are certain cases where the data controller does not have to give some or all of the above information, for example, where doing so would involve a disproportionate cost (see Article 14(5) for more).

6.13.8 The right of access

The data subject has the right to ask whether or not their personal data is being processed. If it is, they have the right to freely receive the following:

- their personal data
- the purpose of the processing
- the categories of personal data collected
- the recipients of personal data, in particular if those recipients are overseas or international organisations
- the retention period, or if one has not been set, the criteria to determine the retention period
- the right to ask the controller to rectify or erase personal data, the right to restrict or object to the processing of their data
- the right to complain
- the source of the personal data, if not directly collected from the data subject
- information about automated decision making, including logic used and the significant consequences of such
- if data is to be transferred to another country or an international organisation, information about the safeguards in place.

Data transfer rules have implications for companies and institutions which take advantage of differential labour costs by routinely sending customer data to non-European countries for processing, for instance, to support customer helplines.

Data controllers may not charge data subjects for exercising their right of access, unless:

Where requests are manifestly unfounded or excessive, in particular because of their repetitive character, the supervisory authority may charge a reasonable fee based on administrative costs, or refuse to act on the request. The supervisory authority shall bear the burden of demonstrating the manifestly unfounded or excessive character of the request.

See: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:02016R0679-20160504&from=EN>

6.13.9 Other rights of data subjects under the GDPR

In addition to their rights over their personal data, data subjects have **the right to be informed of personal data breaches** (Article 34) in a timely way, where such breach 'is likely to result in a high risk to the rights and freedoms of natural persons'. The communication must be in plain language and include the details of a contact point for more information, the likely consequences of the breach, and any measures taken by the controller with respect to the breach.

Data subjects have **the right to complain to the supervisory authority** (Article 77) and in relation to that right have other rights including the **right to receive compensation** from the controller or processor (or both as appropriate) where they have suffered damage (material or otherwise) as a result of the infringement of the Regulation (Article 82).

6.13.10 Special categories of personal data

Under the GDPR additional protections are in place before **special categories** of personal data can be processed. Article 9 states:

Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited.

See: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=en#d1e1489-1-1>

This means that genetic data is not just personal data, it is sensitive personal data, as is biometric data when processed for the purposes of identifying an individual.

Personal data relating to criminal convictions is not included, but the law in member states must regulate processing of such data when not done by an official authority (see Article 10).

Special categories of personal data can be processed if one of the following applies:

- (a) the data subject has given explicit consent (except where member state law forbids the lifting of the prohibition by the data subject);
- (b) processing is necessary for the purposes of employment, social security and social protection law (subject to certain safeguards);
- (c) processing is necessary to protect the vital interests of the data subject or of another natural person where the data subject is physically or legally incapable of giving consent;

- (d) processing is necessary in the course of the legitimate activities of a not-for-profit body (this includes trade unions and religions) and relates to members, former members and others who have regular contact with the organisation;
- (e) the data was made public by the data subject;
- (f) processing is necessary for the establishment, exercise or defence of legal claims or whenever courts are acting in their judicial capacity;
- (g) processing is necessary for reasons of substantial public interest (subject to certain safeguards);
- (h) processing is necessary for medical, including occupational medical purposes, or the provision of social care, subject to certain safeguards including that the data are processed under the authority of a professional subject to the professional confidentiality standards of the member state;
- (i) processing is necessary for reasons of public interest in the area of public health, subject to certain safeguards including member state standards for professional confidentiality;
- (j) processing is necessary for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes subject to certain safeguards.

6.13.11 Restrictions and derogations (exemptions)

There are circumstances when member states can legally restrict the application of the Regulation. There are 10 possible grounds for this, given in Article 23, which are, briefly: national security; defence; public security; law enforcement; other 'important objectives [...] of the Union or of a Member State' in particular where these objectives are economic or financial; the protection of judicial independence and proceedings; taking action regarding ethical breaches in 'regulated professions'; some regulatory functions carried out by the state; the protection of the data subject or others; and the enforcement of civil law claims.

In addition articles 85–91 specify certain conditions where member states may apply derogations (exemptions) from some of the rights of the data subject. Member states are required in Article 85 to adopt laws that 'reconcile the right to the protection of personal data pursuant to this Regulation with the right to freedom of expression and information, including processing for journalistic purposes and the purposes of academic, artistic or literary expression.' Article 89 applies to '[p]rocessing for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes'. Data can be processed for such reasons with exemptions from certain rights of the data subject provided that certain safeguards are in place.

The other articles cover the following issues:

- Public access to official documents (Article 86)
- National Identification Numbers (Article 87)
- Employment (personal data of employees) (Article 88)
- Obligations of professional (or equivalent) secrecy (Article 90)
- Existing data protection rules of churches and religious associations (Article 91).

6.13.12 International data transfers

Articles 44–50 lay out in some detail the conditions relating to international data transfers, that is transfers to a third country outside the EU, and transfers to international organisations. This includes onward transfers. Data transfers could happen if, for instance, a European company opens a call centre in an Asian country, where operators will be able to call up personal data on their computer screens to deal with customer enquiries.

Article 45: Transfers on the basis of an adequacy decision. Where the EU has decided that a country or international organisation has ‘an adequate level of protection’. At the time of writing 12 countries, including the USA, have been recognised, see: https://ec.europa.eu/info/law/law-topic/data-protection/data-transfers-outside-eu/adequacy-protection-personal-data-non-eu-countries_en There is also provision in 45(3) for the EU to make a case-by-case determination, which has to be reviewed regularly, and at least every four years.

Articles 46 and 47: Transfers subject to appropriate safeguards. Transfers can be made where the receiving organisation has appropriate safeguards, and where data subject rights are enforceable with ‘effective legal remedies’ for infringement. The Article lists ways in which such safeguards can be provided, such as by including clauses in a contract. These also include ‘binding corporate rules’ which are specified in some detail in Article 47.

Article 48: Transfers or disclosures not authorised by Union law. This relates to legal judgements in a third country requiring personal data to be transferred. In order for the transfer to take place there must be an international agreement in force between the member state and the third country requesting the transfer or disclosure of personal data.

Article 49: Derogations for specific situations. This Article lists specific cases where transfers may take place without an adequacy decision (Article 45) or appropriate safeguards (Articles 46 and 47). These include:

- (a) the data subject has explicitly consented, knows that their data will not be protected in the same way it would be in the EU, and is aware of the risks
- (b) the transfer relates to a contract agreed with the data subject
- (c) the transfer relates to a contract that is in the interest of the data subject, and is between the controller and another natural or legal person
- (d) public interest
- (e) the transfer is necessary to pursue or defend a legal claim
- (f) the transfer is necessary in order to protect the vital interests of the data subject or of other persons, where the data subject is physically or legally incapable of giving consent
- (g) the transfer relates to a public register.

Where the above conditions do not apply, an international transfer without an adequacy decision or appropriate safeguards may still be possible under certain limited conditions provided that it ‘is necessary for the purposes of compelling legitimate interests pursued by the controller’.

Article 50: International cooperation for the protection of personal data. This commits the EU to developing international cooperation and to provide ‘international mutual assistance’ in data protection law.

Activity 6.4

- (i) The General Data Protection Regulation (GDPR) came into force in 2018. Which one of the following is true?
1. The GDPR does not apply to biometric data
 2. The GDPR does not apply, in any circumstance, to companies based outside the EU
 3. The GDPR cannot be applied to hard copy data, but only to data in electronic form
 4. None of the above
- (ii) Under the General Data Protection Regulation (GDPR) **data subjects** can be:
1. Private individuals
 2. Private individuals and certain non-profit making bodies such as trade unions
 3. Private individuals, certain non-profit making bodies such as trade unions and all government departments
 4. None of the above
- (iii) What does a **personal data breach** mean?
1. A loss of security meaning that personal data has been disclosed to an unauthorised person or organisation, or has been stolen
 2. A loss of security meaning that personal data has been altered
 3. A loss of security meaning that personal data has been destroyed
 4. All of the above
- (iv) Under the GDPR a personal data breach must be reported to the appropriate authority:
1. Within 24 hours of the breach
 2. Within 48 hours of the breach
 3. Within 72 hours of the breach
 4. None of the above
- (v) The GDPR has added certain rights to data subjects, while strengthening those already existing. Which of the following best describes the two new rights granted to data subjects by the GDPR?
1. The **right to be forgotten** and the **right to rectify incorrect data**
 2. The **right to be forgotten** and the **right to data portability**
 3. The **right to be forgotten** and the **right to withdraw consent for data processing within 7 days of granting it**
 4. None of the above
-

6.13.13 Higher profile for data protection under GDPR

Under the GDPR the supervisory authority can fine organisations for infringements of the Regulation. Article 83 specifies fines for business of €10 million or ‘up to 2 per cent of the total worldwide annual turnover of the preceding financial year’, whichever is greater, for certain infringements, and €20 million or up to 4 per cent of worldwide turnover (whichever is greater) for more serious infringements. In addition, failure to comply with an order of the supervisory authority can also mean a fine of up to 4 per cent of annual

worldwide turnover. The fines have been described as ‘eye watering’ by more than one commentator. In contrast, *The Register* reported in July 2018 that if, as expected, the ICO fined Facebook £500,000 (the maximum fine under the Data Protection Act 1998, which was in force at the time of the data breach) for allowing the personal data of 87 million people to be harvested and shared with Cambridge Analytica, this would represent a mere 18 minutes of Facebook profits (see: www.theregister.co.uk/2018/07/11/ico_fine_facebook_cambridge_analytica/). The ICO imposed the £500,000 fine as expected in October 2018. Facebook’s turnover was \$28 billion in 2016. This means that they dodged a fine of more than \$560,000,000, or more than half a billion, based on 2 per cent of 2016 turnover, under the GDPR. If the fine had been based on 4 per cent of turnover, possible considering the serious nature of the breach, then the fine would have been more than a billion US dollars (approximately three quarters of a billion pounds) under the GDPR.

In addition to the new fines, organisations that do significant personal data processing and public bodies must appoint a Data Protection Officer (DPO) to monitor compliance and provide information to employees and customers. The DPO should report at the ‘highest management level of the controller’ (Article 38(3)). This gives data protection much more strategic importance, and more visibility at senior levels. Or, as *Wired* reports the UK Information Commissioner, Elizabeth Denham, as saying, ‘It means the data protection will be a boardroom issue in a way it hasn’t in the past’. See: www.wired.co.uk/article/what-is-gdpr-uk-eu-legislation-compliance-summary-fines-2018

Note that in January 2019 Google was fined €50 million by the regulators in France, for breaching the GDPR. See: www.wired.com/story/eu-privacy-law-snares-first-tech-giant-google/ for more information.

6.13.14 The ePrivacy Directive of 2002

Before the GDPR, data protection in the EU was covered by two directives: the 1995 EU Directive (95/46/EC) and the ePrivacy Directive (2002/58/EC), which was about the protection of privacy and the free movement of data in telecommunications and online:

The E-privacy Directive covers processing of personal data and the protection of privacy including provisions on:

- the security of networks and services;
- the confidentiality of communications;
- access to stored data;
- processing of traffic and location data;
- calling line identification;
- public subscriber directories; and
- unsolicited commercial communications (‘spam’).

See: https://edps.europa.eu/data-protection/our-work/subjects/eprivacy-directive_en

The ePrivacy Directive of 2002 had implications for website operators; and users. The Directive states that data protection law (specifically Directive 95/46/EC) ‘covers any form of processing of personal data regardless of the technology used’ and adds to this specific rules for electronic communications. These rules include such things as that traffic data should be erased or anonymised once no longer needed, location data (apart from traffic data which may be processed for billing purposes) ‘may only be processed when

they are made anonymous, or with the consent of the users or subscribers to the extent and for the duration necessary for the provision of a value added service' and subscribers should be informed of particular risks to network security which could lead to data breaches.

In general, data subjects should be aware of the identity of the person or organisation operating the website and/or using it to collect personal data, and the purposes for which they intended to process the data. Website operators are responsible for the security of personal data when stored or in transit, which should be proportionate to the risk, hence credit card numbers should be encrypted for transmission. If a company uses a web hosting service the data controller is responsible for data protection and should have a written contract with the service regarding security measures.

Users should be able to refuse having 'a cookie or similar device' placed on their machine. Email addresses are personal data, hence data subjects can ask to be removed from mailing lists at any time. Organisations should provide users with the ability to opt out of mailing lists in a simple and obvious way. If data subjects request organisations to stop sending them unsolicited emails, or spam, then organisations in the EU must comply. However, if they are outside the EU the risk is that any contact lets them know that the email address is 'live', and encourages more spam. Since 2018, in theory the provisions of the GDPR would apply to such non-EU organisations (since they would be offering goods and services the provisions of Article 3 may apply); however, in practice enforcement may be an issue.

See: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32002L0058&from=en> for the text of the Directive.

6.13.15 The Cookie Directive

The original ePrivacy Directive, 2002/58/EC, was amended in 2009 (2009/136/EC) to:

include a rule requiring the notification of data breaches (for instance someone whose personal data are lost, modified or accessed unlawfully while being treated by its electronic communications provider should be notified if this breach is likely to affect him/her negatively) and an extension of the Directive to also cover various electronic tags, strengthened enforcement rules, etc.

Among other things, the amended Directive extended the protection against unsolicited marketing via email (spam) to text messages, and had a great deal more to say about data security and breaches than the 2002 Directive. However, it became known as the 'Cookie Directive' as it brought into force a requirement for much greater transparency regarding cookies ('electronic tags'). Under the 2002 Directive, users had to have the opportunity to opt out, but under the amended Directive the user had to opt in by giving informed consent before cookies can be placed on their machine.

Under the amended Directive, visitors to EU websites must be given information about the cookies that the web server will be placing on their machine, the purpose for them, and must explicitly agree. Cookies that are needed in order to facilitate a service requested by the user, such as those to remember items in your shopping bag, are exempt from the notification and consent requirement (paragraph 66). Also exempt are cookies necessary for carrying out a transmission requested by the user (Article 3(5)), such as those that help you to stay logged in. All cookies, whether first or third party,

that track a user's online habits, must be notified to the user. In the UK the Information Commissioner's Office is responsible for monitoring the law implementing the Directive, which took effect in May 2012.

Full text of the 2009 amended directive is here: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:02009L0136-20091219&qid=1542067110597&from=EN>

There is overlap between the GDPR and the ePrivacy Directive, in particular with respect to data breaches and tracking (online monitoring) of individuals. The Directive focuses on data breaches from the point of view of network security, with a requirement on 'competent national authorities' to monitor security risks and best practice to contain them, while the Regulation is focused on the notification of data breaches to the supervisory authority and the rights of data subjects to legal redress including compensation.

The ePrivacy Directive is being revised, but this time as the ePrivacy Regulation. This again will mean that EU member states will not have to implement the law as the ePrivacy Regulation will directly apply. It also addresses the somewhat uneven implementation into law of the ePrivacy Directive. A draft version has been published, but will not be in force for some time, as the draft was contentious and the subject of much lobbying. There is likely to be overlap again between the GDPR and the new ePrivacy Regulation. For example, the GDPR specifies online identifiers as personal data, and this could include RFIDs, used in the Internet of Things. The new ePrivacy Regulation specifically mentions the Internet of Things, and the draft states that:

In order to ensure full protection of the rights to privacy and communications, and to promote a trusted and secure Internet of Things in the digital single market, it is necessary to clarify that this Regulation should apply to the transmission of machine-to-machine communications. Therefore, the principle of confidentiality enshrined in this Regulation should also apply to the transmission of machine-to-machine communications'.
See: <https://ec.europa.eu/digital-single-market/en/news/proposal-regulation-privacy-and-electronic-communications>

In the case of conflict with the GDPR, it may be likely that the ePrivacy Regulation will prevail; however, at the time of writing the Regulation had not been finalised by the EU and was unlikely to be brought into force until at least 2020.

6.13.16 Workplace monitoring of communications

The ICO advises 'where monitoring [at work] involves the manual recording or any automated processing of personal information, it must be done in a way that is both lawful and fair to workers'. The 'law' referred to is data protection law, in this case the GDPR, as the UK's Data Protection Act 2018 does not, as Article 88 of the GDPR allows, 'provide for more specific rules to ensure the protection of the rights and freedoms in respect of the processing of employees' personal data in the employment context'.

Monitoring of employees by their employer can be legal under the GDPR provided that employees have been informed of the monitoring. However, it must also be justified; that is the employer needs to be able to show that any harm to employees from monitoring is offset by the benefits to the employer and others. This judgement is made with the help of an 'impact assessment'. If the monitoring cannot be justified then the employer will need the permission

of employees for the monitoring. Monitoring can include such things as CCTV cameras used for health and safety purposes, using software to scan emails for inappropriate content, checking telephone logs for calls to inappropriate numbers and website logs for visits to inappropriate sites.

Employers can legally monitor employees' use of the telephone and internet in the workplace by listening to voicemail messages, listening to and recording phone calls and reading emails, all of which are defined, in the UK, to be interception. The Regulation of Investigatory Powers Act 2000 (RIPA 2000) made the interception of telecommunications (including email, fax, phone and internet services) illegal. According to Gillespie (Gillespie, 2015 Chapter 4, heading *Lawful business practices*) the UK Government realised that businesses may have a legitimate reason for intercepting the communications of employees, and so introduced the Telecommunications (Lawful Business Practice) (Interception of Communications) Regulations 2000, to provide certain exemptions from RIPA 2000. Under the Lawful Business Practice Regulations, interception is legal if it is solely done for monitoring or recording transmissions relevant to the business, the employee is using equipment provided for their work and 'the system controller has made all reasonable efforts to inform every person who may use the telecommunication system in question that communications transmitted [...] may be intercepted'.

Interception may be part of routine monitoring of all workers, or it may be targeted at a certain employee who is suspected of unauthorised activity, such as ignoring confidentiality issues or not following financial protocols. There is some cross-over between interception under RIPA 2000 and the GDPR. Interception may involve the processing of personal data, in which case data protection law, such as that of secure storage of recorded phone calls, keeping data only as long as it is needed for a specific purpose, etc. would apply.

Covert monitoring is rarely legal in UK law, and the ICO's advice is that it should be authorised by senior management only in exceptional circumstances such as that 'there are grounds for suspecting criminal activity or equivalent malpractice and that notifying individuals about the monitoring would prejudice its prevention or detection'. In addition the investigation should be targeted very specifically at finding evidence, should be time-limited and stop immediately once completed. See: https://ico.org.uk/media/for-organisations/documents/1064/the_employment_practices_code.pdf (note that in other respects the code has not been updated to take account of the GDPR).

Different companies will have different policies, and different countries different laws, but, when starting work for a new employer, it is a very good idea to find out what laws and policies – with respect to workplace monitoring of communications – apply to you.

6.14 Unresolved legal issues at the time of writing

This section does not contain any examinable material.

6.14.1 The EU Directive on Copyright in the Digital Single Market and the end of the internet as we know it

As part of its proposals to strengthen the digital single market, the EU proposed the Directive on Copyright in the Digital Single Market 2016/0280(COD), see: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52016PC0593&from=EN>.

Articles 11 and 13 of the proposed Directive have caused some concern. In a story published in June 2018, *The Guardian* noted that:

an open letter signed by 70 of the biggest names of the internet, including the creator of the world wide web, Tim Berners-Lee, and the Wikipedia founder, Jimmy Wales, argued that article 13 would take 'an unprecedented step towards the transformation of the internet from an open platform for sharing and innovation, into a tool for the automated surveillance and control of its users'.

See: www.theguardian.com/technology/2018/jun/20/eu-votes-for-copyright-law-that-would-make-internet-a-tool-for-control

The Guardian explored claim and counterclaim in a September 2018 report, with opponents claiming the result of Articles 11 and 13 will be 'catastrophic' while one supporter is quoted as saying: 'This is a great day for Europe's creators'. See: www.theguardian.com/law/2018/sep/12/eu-copyright-law-may-force-tech-giants-to-pay-billions-to-publishers-facebook-google. The report notes that Google and Facebook would be forced to share more of their revenue with news agencies:

Europe's biggest news agencies have [...] accused Google and Facebook of 'plundering' the news and their ad revenues, resulting in a 'threat to democracy'.

The report also notes that musicians such as Paul McCartney and Adele are in favour of the Directive:

Article 13 would make platforms such as YouTube seek licences for content such as music videos, which artists say will allow them to properly negotiate better royalties.

In another *Guardian* story from September 2018, German Christian Democrat Axel Voss is quoted as follows:

'We have had this with the banks, we have had this with the telecoms [industry] and now with the internet giants: that regulation done by the EU regulator will lead to something catastrophic,' he said. 'We will not end the internet.'

See: www.theguardian.com/technology/2018/sep/09/battle-over-eu-copyright-law-heads-for-showdown

In September 2018, *Wired* published an article with the heading: 'If we don't act now, Article 13 could break the internet by mistake', see: www.wired.co.uk/article/article-13-will-kill-the-internet-by-mistake

The *Wired* piece argues that 'power and profit has shifted from record labels and publishing houses to internet platforms', and that lobbying from the losers in this shift has convinced the EU to 'find ways to redistribute income back to them'. The article states:

Articles 11 and 13 – a so-called 'link tax' for news content, and a rule making internet platforms liable for copyright infringements by their users – were never meant to be carefully considered fixes to issues in copyright law. They are meant to be sticks EU-based publishing and music giants can wield to force US-based internet platform giants to the negotiation table on their terms. To achieve that, some kind of license needs to be required, and the media industry needs to get to set the price.

The article reflects the author's belief that the Directive is about curbing the power of the world's largest tech companies, Google, Apple, Facebook and Amazon (collectively known as GAFA). However, Article 13 reflects how the

internet and the world wide web have made possible file sharing and other acts that have made copyright infringement more likely, monitoring of infringements problematic and have made it more difficult to enforce copyright.

The Directive came into force in June 2019 with EU member states having 2 years to pass laws implementing its provisions.

Article 11 of the EU Directive on Copyright in the Digital Single Market

This has been called the 'link tax' and would force sites such as Google and Facebook to pay publishers for showing links accompanied by snippets of text that preview the story for users. It is claimed that this will not apply to individuals sharing links, but does this include individuals that have a large social media following?

Article 13 of the EU Directive on Copyright in the Digital Single Market

This Article has been variously called an upload filter, and a meme ban. Sites would have to find some way to automatically check new content for copyright. It has been claimed that this will cause memes to be rejected, since the software will not be able to discriminate between parody and copyright theft. Some have expressed the fear that the Article will be a gift to 'copyright trolls', and that there are likely to be many false positives with the recourse individuals have in such cases, if any, undetermined at this point. *The Guardian* has raised the possibility 'for widespread exploitation of the system as a new strain of denial-of-service attacks by online mobs, to get content people dislike taken down with a barrage of fake claims.' See: www.theguardian.com/commentisfree/2018/sep/13/tech-giants-eu-internet-searches-copyright-law

6.14.2 The future of software

In 2010 Oracle – after acquiring Sun Microsystems and with it the Java programming language – took out a case against Google in the US court system. Oracle claimed copyright and patent infringement for Google's use of the Java API in the Android operating system, even though Oracle makes the API available to developers to use with no restrictions. Google claimed fair use, and the courts initially found in their favour in 2012. The case dragged on until 2018 when an appeal from Oracle was upheld. In 2016 *Wired* published an article claiming that the determination of the case would decide the future of software, see: www.wired.com/2016/05/oracle-google-case-will-decide-future-software/ This remains to be seen; however, many in the tech industry have been following the case closely.

Wired wrote another article in March 2018, claiming that a federal appeals court judgment in Oracle's favour (upholding another court decision in 2016) 'could force many software companies to rewrite parts of their products, even if they're not using Java or any other Oracle software. That will not only be expensive, but could make applications and services from different companies less compatible'. *Wired* urged Google to appeal to the Supreme Court 'because the decision will affect not just Google and Oracle, but the entire software industry'. See: www.wired.com/story/the-case-that-never-ends-oracle-wins-latest-round-vs-google/ After another court victory for Oracle in August 2018, Google announced it was taking its case to the Supreme Court.

See: <https://arstechnica.com/information-technology/2016/06/the-googleoracle-decision-was-bad-for-copyright-and-bad-for-software/> for a good discussion of the issues, and arguments for and against the 2016 court judgment in Oracle's favour.

6.14.3 Net neutrality

In the USA, the issue of net neutrality has been contentious for years. Net neutrality means that sites such as Netflix cannot be required to pay a premium so that their customers' access is not slowed down or blocked. Removing net neutrality would contradict a fundamental principle of the internet that a 'best effort' is made to deliver each packet. Within a particular data stream certain packets may be prioritised, but data streams themselves cannot be prioritised – all have the same importance. In 2017, the Trump administration repealed the Federal Communication Commission's (FCC) 2015 Open Internet Order, which affirmed the principle of net neutrality. This has met some resistance, including that 30 states have introduced laws guaranteeing net neutrality for their citizens. Under pressure from citizens, the House of Representatives has voted to block the FCC's decision, but the Senate also needs to vote on this, and so far efforts to force a vote have failed; this is despite 86 per cent of Americans being opposed to the repeal of net neutrality.

6.14.4 Artificial intelligence

In 2018 researchers from MIT published the result of interviews conducted worldwide, regarding the ethics of driverless cars. They were looking at the issue of how a driverless car should be programmed to respond in situations where lives are in danger. Specifically, should the car prioritise the life of passengers and sacrifice pedestrians? Should the car try to minimise the loss of life regardless of who would be sacrificed? Should the lives of the young have priority? See: <http://news.mit.edu/2018/how-autonomous-vehicles-programmed-1024> for more detail. This research has been the subject of much discussion and comment, with the *The Guardian* quoting a Google engineer who believed that the research has limited application because the correct answer in most situations was to brake, quickly: 'your control is much more precise by slamming on the brakes than trying to swerve into anything. So it would need to be a pretty extreme situation before that becomes anything other than the correct answer'. See: www.theguardian.com/technology/2018/oct/24/who-should-ai-kill-in-a-driverless-car-crash-it-depends-who-you-ask

One website summed up the research with: 'Most people would like others to buy cars programmed to save the lives of pedestrians, but would themselves prefer to ride in a driverless car that protected its own passengers at all costs, the researchers found.' See: www.iflscience.com/technology/driverless-cars-should-sacrifice-their-passengers-for-the-greater-good-just-not-when-im-the-passenger/

Ethical issues can also become legal issues; for example, if a driverless car kills someone, is the person who wrote the algorithm responsible for a decision that prioritised the safety of those in the car over pedestrians? Is the manufacturer of the car responsible? Or will countries at some point start legally requiring certain standards to be hard-wired into autonomous vehicles? If so, what will those standards be?

Wired wrote about the ethics of AI, with one contributor arguing that 'national and supranational laws and regulations, such as GDPR, will be crucial to establish boundaries and enforce principles' while another claimed that the law would have to vary by country to take account of cultural differences, see: www.wired.co.uk/article/artificial-intelligence-ethical-framework The report also notes that AI is moving into space, but fails to suggest strategies for when space-based AI turns against humanity.

Through the **AI4People** initiative, the EU is attempting to ‘create a common public space for laying out the founding principles, policies and practices on which to build a “good AI society”’. The initiative aims to agree and recommend an ethical framework for AI, followed by policy recommendations, and in 2020 ‘to undertake analysis to decide how best to utilise law, standards and regulation to accommodate the new capacities, practices and behaviours enabled by AI’. This includes considering the ethical challenges to policymaking of, for example, “predictive policing” and the use of drones in warfare’. The **AI4People** initiative also aims in 2020 to pilot a ‘Global Mark of Compliance’, for certifying ethical AI companies, practices, services and engineers. See: www.eismd.eu/ai4people/

6.15 Overview of the chapter

In this chapter we considered the law relating to hacking, writing malicious code, data protection for individuals, defamation and intellectual property. We considered the law in each area with particular respect to the EU and the UK, but tried to give an international perspective. We noted that IT professionals need to keep up-to-date with new and developing technology, as well as with technologically related social, legal and political issues. Finally, we looked at some currently unresolved legal issues of potential interest to IT professionals.

6.16 Reminder of learning outcomes

Having completed this chapter, and the Essential reading and activities, you should be able to:

- explain the difference between patent and copyright, describe some ways in which internet technology has made it more difficult to enforce copyright or monitor copyright infringements, and discuss arguments for and against extending patent protection to computer software
- list the principles of data protection
- explain what rights you have as a data subject in relation to persons or organisations holding your details on file
- explain what companies must do to keep within the law if they keep records of individuals on manual or electronic files (assuming UK jurisdiction)
- explain the legal implications of computer hacking, virus writing and unauthorised access to computer systems.

6.17 Test your knowledge and understanding

6.17.1 Sample examination questions

- (a) (i) Which one of the following is true? [2 marks]
1. **Defamation** means publishing a statement that could harm a person's reputation.
 2. In the context of defamation, **publishing** means telling one or more people.
 3. A defamatory statement that is untrue falls under the law of either libel or slander.
 4. All of the above.
- (ii) Which one of the following is correct? [2 marks]
1. TRIPS is an international convention about intellectual property law.
 2. CHIPS is an international convention about intellectual property law.
 3. GRIPS is an international convention about intellectual property law.
 4. None of the above.
- (iii) Which one of the following is true? [2 marks]
1. Under the GDPR a data subject's genetic data is not protected.
 2. Under the GDPR a data subject's genetic data is considered personal data and can be processed with other personal data, applying the same safeguards.
 3. Under the GDPR a data subject's genetic data is considered to be special category data; processing of such data is subject to additional safeguards to that given to personal data.
 4. None of the above.
- (b) (i) Give the missing words from the following two sentences, (A) and (B):
- (A) _____ may be granted to applications that include an inventive step, and that can be used in industrial applications. They have to be applied for, they may not necessarily be granted, and, if granted, expire after a limited time.
- (B) _____ is automatically granted to the originator, is time limited and the rights granted can be divided into moral and economic rights [2 marks]
- (ii) What is the EU's **database right**? What intellectual property does it protect? [7 marks]
- (c) At the moment there is not one agreed upon a definition of cybercrime, with different authorities having different opinions. Provide your definition of cybercrime. [10 marks]

Notes

Chapter 7: Network security issues

7.1 Introduction

This chapter provides a technological perspective on issues of hacking and computer security, giving examples of vulnerabilities and discussing malicious software, or malware, including viruses, worms and Trojan horses. Note that Section 7.12 is for information only and is not examinable.

7.1.1 Aims of the chapter

This chapter aims to help you to develop an understanding of network security with respect to hacking and malicious code.

7.1.2 Learning outcomes

By the end of this chapter, and having completed the Essential reading and activities, you should be able to:

- explain the differences between viruses, worms and Trojan horses
- describe some of the software vulnerabilities they exploit and the malicious techniques they use
- advise on measures to avoid exposure to malicious code
- explain in general terms how anti-virus software functions.

7.1.3 Essential reading

This chapter in the subject guide provides a complete account of the topics covered and is Essential reading.

7.1.4 Further reading

- Comer, D. E. *Internetworking with TCP/IP Volume 1*. (Harlow: Pearson Education, 2014) 6th edition Pearson new international edition [ISBN 9781292040813]. Chapter 29, *Internet Security and Firewall Design* (IPSec, SSL).
- Gillespie, A.A. *Cybercrime: Key Issues and Debates*. (Abingdon, Oxon; New York, NY: Routledge, 2015) [ISBN 9780415712217 (hbk); 9780415712200 (pbk); 9781315884201 (ebk)]. Chapter 3 *Targeting the Technology*.
- Casad, J. *Sams Teach Yourself TCP/IP in 24 Hours*. (Indiana: Pearson Education, 2017) 6th edition [ISBN 9780672337895]:
 - **Hour 11:** TCP/IP Security
 - **Hour 19:** Encryption, Tracking and Privacy
- www.theregister.co.uk/2012/12/14/first_virus_elk_cloner_creator_interviewed/ – an interview with the man credited with creating the first computer virus.
- Schneier, B. *Click Here to Kill Everybody: Security and Survival in a Hyper-connected World*. (New York; London: W.W. Norton & Company, 2018) [ISBN 9780393608885 (hbk); 9780393357448 (pbk; 2019)] [ASIN B07BLMQKZK (ebk)].
 - Introduction *Everything Is Becoming a Computer*
 - Chapter 1 *Computers are still hard to secure*
 - Chapter 2 *Patching is failing as a security paradigm*.

7.1.5 References cited

- <https://dictionary.cambridge.org/dictionary/english/phishing>
- www.yourdictionary.com/class-break
- <https://blog.malwarebytes.com/threats/trojan-dropper/>
- <https://blog.malwarebytes.com/threats/remote-access-trojan-rat/>
- <https://theintercept.com/2014/03/15/nsa-facebook-malware-turbine-non-denial-denial/>
- www.theregister.co.uk/2018/04/13/slow_android_security_fixes/
- www.wired.com/story/wannacry-hero-marcus-hutchins-new-legal-woes-white-hat-hackers/
- www.theregister.co.uk/2019/04/19/marcus_hutchins_pleads_guilty/
- www.theregister.co.uk/2017/04/10/shadow_brokers_open_sources_hacker_trove/
- www.theguardian.com/technology/2017/may/12/nhs-ransomware-cyber-attack-what-is-wanacrypt0r-20
- <https://enterprise.verizon.com/resources/reports/dbir/>
- www.kaspersky.co.uk/blog/browser-extensions-security/12750/
- www.wired.co.uk/article/police-iphone-hacking-grayshift-graykey-uk
- www.theguardian.com/commentisfree/2018/apr/06/phone-camera-microphone-spying
- www.theregister.co.uk/2015/08/24/smart_fridge_security_fubar/
- www.kaspersky.co.uk/resource-center/threats/adware
- www.symantec.com/blogs/expert-perspectives/surge-blended-attacks-stirs-new-cyber-worries
- <https://nakedsecurity.sophos.com/2016/06/16/is-angler-exploit-kit-dead/>
- www.avast.com/c-locky
- <https://nakedsecurity.sophos.com/2016/04/26/ransomware-in-your-inbox-the-rise-of-malicious-javascript-attachments/>
- www.kaspersky.co.uk/resource-center/definitions/what-is-a-polymorphic-virus
- www.webroot.com/us/en
- www.kaspersky.co.uk/resource-center/definitions/metamorphic-virus
- www.kaspersky.com/enterprise-security/wiki-section/products/fileless-threats-protection
- www.mcafee.com/enterprise/en-gb/threat-center.html
- <https://securelist.com/usb-threats-from-malware-to-miners/87989/>
- www.theguardian.com/technology/2018/jan/23/cybercrime-130bn-stolen-consumers-2017-report-victims-phishing-ransomware-online-hacking
- https://usa.kaspersky.com/about/press-releases/2018_fifa-2018-and-bitcoin-among-2017-most-luring-topics
- www.kaspersky.com/resource-center/definitions/drive-by-download
- <https://blogs.technet.microsoft.com/msrc/2017/07/26/announcing-the-windows-bounty-program/>
- www.theguardian.com/books/2014/may/12/glenn-greenwald-nsa-tampers-us-internet-routers-snowden
- <https://arstechnica.com/gadgets/2018/01/meltdown-and-spectre-every-modern-processor-has-unfixable-security-flaws/>
- <https://searchsecurity.techtarget.com/news/252435175/Intel-bug-bounty-programs-widened-after-Meltdown-and-Spectre>

- <https://arstechnica.com/information-technology/2018/10/supermicro-boards-were-so-bug-ridden-why-would-hackers-ever-need-implants/>
- www.bbc.co.uk/news/business-48588661

7.2 Glossary of key terms

- **Computer program:** compiled, executable instructions telling a computer how to perform certain tasks.
- **App:** a computer program that depends on the operating system to be able to run.
- **Floppy disk:** now obsolete removable storage medium used in the first personal computers.
- **Attack surface:** this is a concept that cybersecurity specialists use. Bruce Schneier describes it as 'all the possible points that an attacker might target and that must be secured' (Schneier 2018, Chapter 1, heading *The complexity of computerized systems means attack is easier than defence*).
- **Hacker:** someone who attempts to gain unauthorised access to computers and computer networks.
- **Script kiddie:** a hacker of limited ability, who uses off the shelf tools developed by more talented and/or experienced hackers.
- **National Security Agency (NSA):** the NSA has its origins in the First World War, intercepting and decrypting the communications of enemies and friends. It is the US agency responsible for signals intelligence, which includes telecommunications and the internet.
- **Government Security Headquarters (GCHQ):** The UK equivalent of the NSA, works in close cooperation with the NSA.
- **Malicious code:** code that is not under the control of the user of the computer system, and is designed to take actions that the user has not initiated and would not choose, such as causing damage to a system or computer, security breaches or other unwanted action.
- **Class break:** once vulnerability in a computer system or software has been found, it is likely to be exploited many times. 'A security breach that is the first of its kind, and leads to others of the same class', see: www.yourdictionary.com/class-break
- **Malware:** a contraction of **malicious software**. It is software written with the intention of carrying out acts that are unauthorised by the owner of the machine or system in which they have been installed.
- **Dropper:** malware that is designed to install other malware onto the target machine. They are defined at: <https://blog.malwarebytes.com/threats/trojan-dropper/> as: 'Downloaders and droppers are helper programs for various types of malware such as Trojans and rootkits. [...] They don't carry any malicious activities by themselves, but just open a way for attack by downloading/decompressing and installing the core malicious modules. [...] Very often, they auto-delete themselves after the goal has been achieved.'
- **Virus:** viruses existed before the internet, and were spread by users swapping floppy disks. A virus is a self-replicating program that attaches itself to legitimate programs, and when they are invoked, it is too.
- **Worm:** A worm is a self-replicating autonomous program that spreads itself by exploiting security loopholes in a network.

- **Trojan:** malicious software that pretends to be something desirable, such as a helpful utility, or a game.
- **Remote access Trojan:** a type of Trojan that allows a hacker to get access to the target machine by installing a back door. 'This backdoor into the victim machine can allow an attacker unfettered access, including the ability to monitor user behavior, change computer settings, browse and copy files, **utilize the bandwidth** (Internet connection) for possible criminal activity, access connected systems, and more' (emphasis the author's own). See: <https://blog.malwarebytes.com/threats/remote-access-trojan-rat/> Access to bandwidth is often used to recruit an infected machine into a botnet.
- **Backdoor:** These can be legitimately written by developers, to provide access to a system without having to pass through all of its security requirements. However, they have obvious utility to criminals and hackers, who will attempt to find existing backdoors, and to install their own. See: <https://searchsecurity.techtarget.com/definition/back-door>
- **Botnet:** A network of compromised machines controlled by a third party, who is able, through a controlling interface, to send a single command that all machines in the botnet will carry out. The machines in the botnet can include personal computers, mobile and Internet of Things devices.
- **Denial of Service (DoS) and Distributed Denial of Service (DDoS):**
 - **DoS** is a malicious attempt to prevent legitimate users from accessing a system or server, by overloading the server with network traffic.
 - **DDoS** is a malicious attempt to prevent legitimate users from accessing a server by using many machines in an organised attempt to overwhelm the server with network traffic. Such attacks often make use of botnets.
- **Fileless malware:** malware that is resident in memory, rather than saved on the hard drive.
- **Targeted malware:** malware designed to target a particular individual or organisation. This may be for revenge attacks, or for industrial espionage or sabotage.
- **Rootkit:** a set of tools that allows hackers, and legitimate security specialists to take control of machines.
- **Bootkit:** malicious software comprising a rootkit that hides in the master boot record.
- **Stuxnet:** a computer worm used to attack Iran's nuclear program in 2010, and thought to be a joint American/Israeli product, although neither country has admitted any responsibility.
- **Mirai:** malware that can be used to recruit devices running Linux into a botnet. It has been used in large scale attacks, using Internet of Things devices.
- **WannaCry:** a notorious world wide coordinated ransomware attack from 2017.
- **Codec:** software to compress and decompress audio and video. There are thousands of codecs, some are free and some not. Codecs are potential malware vectors.
- **Malvertisements:** the use of online advertisements to spread malware.
- **Drive-by downloads:** By taking advantages of known and unpatched security issues on websites, criminals can infect targets who have only opened and viewed the web page, without any other form of interaction.

- **Packet sniffing:** software that intercepts packets in a network data stream. Packet sniffers have legitimate uses, but are also deployed by hackers to steal sensitive data.
- **Phishing:** 'an attempt to trick someone into giving information over the internet or by email that would allow someone else to take money from them, for example, by taking money out of their bank account'.
<https://dictionary.cambridge.org/dictionary/english/phishing>
- **Spoofing:** disguising a source as something other than it is in order to inspire trust and conceal malicious intent. Emails can easily be spoofed by falsifying the 'from' field. URLs can be spoofed by making small changes such that the URL looks like that of a well-known and trusted site; normally clicking on such a link will take the user to a site whose design mimics that of the trusted site.
- **Sandbox:** software that provides an environment to run and test programs in, isolated from critical systems.
- **Signature of a virus:** The signature of a virus or other malware is a unique sequence of bits used to identify it by anti-virus and anti-malware software.
- **Signature detection:** used by anti-virus and anti-malware software to detect viruses and other malware. A unique sequence of bits is identified in the virus (or other malware) and a string replicating this is added to the software vendor's database as the malware signature for identification purposes.
- **Whitelisting:** anti-malware software can add applications to a trusted list, known as a white list, which allows access to certain services and protocols without question.
- **Firewall:** hardware and/or software (may be a combination) to prevent unauthorised access to networks. Firewalls monitor and control network traffic, both outgoing and incoming.
- **Internet of Things:** the connection of consumer products and devices with embedded systems (for example, thermostats, fridges, cars, fitness trackers, cameras, etc.) to the internet.
- **MS-DOS:** Microsoft disk operating system. Before Windows, Microsoft supported a text based operating system, MS-DOS.
- **Software patches:** updates to software designed to improve it, or fix a problem, delivered over the internet. This includes fixes for newly discovered vulnerabilities. Operating systems for home users, such as Windows, are normally configured to automatically apply updates.
- **Responsible disclosure:** a system where researchers finding security vulnerabilities in software notify the vendor, giving them time to develop a patch before the public are notified. Public notification is delayed in order not to tip off criminals about the newly discovered vulnerability, but it is not delayed indefinitely in order to encourage vendors to develop a patch for the vulnerability.
- **Bug bounty programs:** software and hardware manufacturers pay researchers to notify them of any security vulnerabilities first.
- **Zero day vulnerabilities:** vulnerabilities used to attack hardware or software before the vendor knows about and can patch them.
- **CVE ID numbers:** used to uniquely identify publicly known cybersecurity vulnerabilities.

- **Two-factor authentication:** improves security by picking two of three things: (1) something the user knows (password or phrase, answer to a security question); (2) something they have with them (for example, mobile phone); and (3) something they are (biometrics).
- **Browser plug-in:** A plug-in provides more functionality for a web page, for example playing video. Since users installed plug-ins to their web browser in the form of executable code they became notorious security holes, and are now blocked by many browsers.
- **Browser extensions:** Extensions are used for such things as ad blockers, and allow browsers to be customized with source code. Since the code is not executable they are less of a security concern, but still have security issues.

7.3 What is hacking?

Hacking is gaining unauthorised access to a computer or a computer network.

7.3.1 White hat hackers

Most authorities distinguish between malicious hacking and hacking that is driven by curiosity or some other motive. One simple distinction that has often been made is between **white hat** and **black hat** hacking. A white hat hacker is supposedly a hacker who is curious, or is testing system security. However, different authorities vary in their definitions: some say a white hat hacker is a security specialist testing network security, while others include ethical hacking, done by those who believe in freedom of information, for the purpose of sharing information that they believe should be freely available. Others define white hat hackers to be paid professionals in computer security, who only ever hack with permission, while others say that ethical hackers are those who hack into systems without permission but always notify the system owners of any security flaws they have found. White hat hacking could be summed up as hackers who attempt to circumvent security measures in order to access a computer or a computer network with no malicious intent.

In contrast black hat hackers are those who do have malicious intent, either in terms of profiting from their unauthorised access, or simply in terms of doing damage, cybervandalism. There are also **grey (or gray) hat hackers**, supposedly a mixture of the first two motivations, for example, a grey hat would hack without permission and might attempt to receive payment for disclosing security flaws. Or they might not disclose those security flaws at all, or for some time, in order to maintain their access to the system. Their intent is not malicious, but their motives are also impure. To this list others would add **script kiddies**: those who use hacking tools written by others and have limited coding skills; and **hacktivist**: those who are motivated by a belief in social change. Still others talk about **red hat hackers** (supposedly hackers that aggressively go after black hat hackers) and it all gets very confusing.

7.3.2 Motivations for hacking

Gillespie (2015, Chapter 3, heading *Taxonomy of hacking*) discusses a taxonomy of hacking based on behaviour and skill, starting with novices (aka script kiddies), including disgruntled employees and petty criminals, and ending with professional criminals, 'information warriors' (those who attack a state) and political hacking (hacktivist). Gillespie notes that the taxonomy does not include 'white hat' hacking, since it is delineating a spectrum of illegal behaviour, but does show that people hack for very different reasons.

One thing missing from Gillespie's list is the state. As Edward Snowden has revealed, the NSA, helped by GCHQ and other actors, is using malware to spy on its citizens. In a 2015 report, the *Intercept* claimed that the NSA had used a fake 'Facebook server in order to covertly infect targets with malware "implants" used for surveillance'. See: <https://theintercept.com/2014/03/15/nsa-facebook-malware-turbine-non-denial-denial/>

7.3.3 What hackers do

Gillespie (2015, Chapter 3, heading *What do hackers do?*) lists six things that another authority believes are the most common hacker actions:

1. Theft of resources
2. Theft of information
3. Vandalism (more grandly called 'sabotage')
4. 'Website defacement'
5. Denial of service attacks, including distributed denial of service
6. Installing malware.

7.4 Why are networks and computers so vulnerable?

There are several reasons for the vulnerability of networks to malicious software (malware) and other attacks.

Bruce Schneier (2018, Chapter 1) suggests several reasons why computer security is still a challenge, and an area that provides gainful employment to many.

- **'Computers are still hard to secure'**: Schneier claims that securing computers is hard, and quotes security expert Gene Spafford who said: 'The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards – and even then I have my doubts'.
- **'Most software is poorly written and insecure'**: Schneier claims that as the market does not reward good software that takes a lot of time and money to develop, the focus on meeting deadlines and staying within budget means that software bugs do not get fixed, provided that the software meets a basic standard of functionality. Or, as Schneier says, all software has bugs, some have more than others, and it is hard to know which of these bugs will turn out to be security issues, but some will.
- **'The internet was never designed with security in mind'**: see next section.
- **'The extensibility of computers means everything can be used against us'**: This point is particularly about the Internet of Things. Embedded systems have general purpose chips, cameras do not have camera chips, and ovens do not have oven chips (sorry about that). Hence anything that a computer can do, an embedded system in your smart door lock can do. Schneier says this has three implications:
 - Extensible systems are hard to secure because there are so many potential configurations, it is hard to guard against them all.
 - Extensible systems cannot be externally limited, although you can try. Schneier notes that it is impossible to prevent a music player from playing pirated music, or a 3D printer from printing gun parts. There may be software that prevents the average user, but it will not stop an expert.
 - Extensible systems can have new capabilities added, legitimately, and by hackers.

- **'The complexity of computerised systems means attack is easier than defence':** Schneier states that the internet is the most complex machine ever built, with a huge attack surface. An attacker only has to find one vulnerability. In addition, complexity means that users get security wrong, leaving themselves vulnerable.
- **'There are new vulnerabilities in the interconnections':** The many interconnections leads to emergent properties and unintended outcomes that are hard to predict, even for experts.
- **'Computers fail differently':**
 - Distance does not matter in the same way it does with other crime. A home owner does not have to secure their house against every potential burglar, anywhere in the world, but a sysadmin has to secure their network against every attacker.
 - The ability to attack can be captured in software and sold, or given away. Attackers with limited abilities can utilise tools designed by the smartest and most experienced hackers.
 - 'Computers fail all at once or not at all': Schneier introduces the concept of a 'class break' – where a vulnerability affects all computers with a certain characteristic, such as all those running a certain operating system. This is made worse by a 'software and hardware monoculture' where three operating systems, and two mobile operating systems, dominate the market.
- **'Attacks always get better, easier, and faster':** as technology improves so do attacks. Attackers will use today's tech, often deployed against systems that are out of date.

Schneier notes that class breaks 'lead to worms, viruses and other malware. Think "attack once, impact many."'

Schneier also believes that credential stealing is getting easier and authentication harder. He notes a fundamental problem with authentication, which is that once you are in, you are in. That is many systems are designed such that once you are through the authentication process you can generally do anything at all without further authentication.

7.4.1 Protocol design

Security was not originally 'designed in' to TCP/IP protocols since they were developed at a time when computing power was generally only available to governmental organisations, universities and large companies, and access to networks was strictly limited. Security features such as encryption and user authentication have had to be 'layered' on top of the original design. For example, the original specification for SMTP (Simple Mail Transfer Protocol, see Chapter 5) did not include any requirement that the sender identified in the 'From' header have the same ID as the logged-on user. Once, email clients would open file attachments by default if a user clicked on the icon, which could cause a malicious program to execute. Because email became such a notorious virus vector, current providers such as Gmail automatically virus scan all attachments, and most organisations implement virus scanning for their email servers. Another factor exploited by attackers is that certain protocols require a mandatory response to particular types of messages; this can be exploited in the form of denial of service attacks (see Section 7.7.4). In particular, a server waiting for the third part of TCP's three-way handshake will wait some time before timing out, and this has been exploited in denial of service attacks, with to date no optimum way of defending against it (see later).

7.4.2 The security paradigm

Bruce Schneier considers it problematic that the paradigm most systems operate by is that once the user is through the credential check, they can do anything that it is possible for them to do in that system, without further checks. Some organisations will request further authentication as the user attempts to take certain actions. Banks that provide services to individuals – known as consumer or retail banking – have thought about what actions an unauthorised user could take that could compromise their customers' accounts, and therefore ask for extra authentication before carrying out those actions. For example, adding a payee to a list of the people and organisations that the customer can transfer money to is an obvious first step for an unauthorised user in clearing out the customer's bank account, and could result in an extra security check, such as a message sent to the customer's phone with a security code.

7.4.3 Criminal involvement and opportunity

Computers have escaped from their academic and military ecology and spread worldwide, into banks, business, health care, etc. There are now opportunities for criminals that did not exist in the early internet. Once viruses were annoying but harmless. One of the first viruses was written by a teenager and it left a message on infected machines, but did no deliberate damage. Compare that to the 'WannaCry' attack in 2017, which infected computers worldwide, encrypting data and demanding a ransom for decryption in bitcoin.

7.4.4 Operating systems and network security

Operating systems were not built with security in mind, and are still constantly playing catch up with the writers of malware. It took some time for the most widely-used operating systems to migrate from a 'personal computing' to a 'networked computing' model. Enterprise systems such as Unix were designed to support multiple users concurrently, and consequently have a battery of mechanisms to ensure that users cannot access or corrupt files or memory locations belonging to other users or the operating system itself. Generally, multi-user systems have a class of 'superusers' or 'administrators' who can modify system files – other users do not have this privilege.

When personal computers started becoming popular during the 1980s, they typically ran operating systems that were designed for a single user and lacked some of the built-in security features of multi-user operating systems. For example, MS-DOS (Microsoft Disc Operating System) was implemented on the Intel 8088 chip, which had no mode bit and therefore no way to have different levels of privilege, hence no way to prevent user programs overwriting the operating system or addressing hardware devices directly. It was during the heyday of DOS that the virus-writing culture established itself.

The first Windows operating systems (versions 1.0 released in 1985 to 3.1 released in 1992) were built on top of DOS which had no privilege levels for different users or other security features. Windows NT, a multi-user operating system intended for high-end devices and first released in 1993, did provide security features. A version was released for work stations and one for servers. Windows XP, released in 2001, was based on Windows NT and had different user privilege levels, but most users logged on as the administrator as they found the limited user account too restrictive, in particular much Windows software assumed it had administrator privileges and would not run properly in a user account. This meant that if malicious software found its way onto

the machine, it would have the highest level of privilege; that is, access to modify and delete all files. In addition Windows XP had an autorun feature that made external media such as CDs and flash drives particular security risks as applications on such media would automatically be run once the operating system detected a connection. Another vulnerability for Windows users was that they would search the internet for software, leaving them vulnerable to being tricked into installing malware.

In 2001 Apple switched to OSX, a Unix-based OSX operating system, which had a variety of security features. It took until 2004, prompted by consumer concern about malware, before Microsoft became serious about network security in home computing. The release of Windows XP Service Pack 2 represented a step change in their approach to security, incorporating a security centre that dared to suggest to users that they might like to install anti-virus software.

Windows 10 has the Microsoft Store, a development of the app store first introduced in Windows 8. The Microsoft Store is the primary place to access Windows apps, and other content including music, films, TV and games. Apps in the store are tested and certified by Microsoft; therefore, with a central place to find trusted software, Windows is catching up to Android (the Google Play store) and Linux (software center). Today Windows incorporates a firewall, has different access levels and no longer runs programs with autorun. In particular, Windows – similarly to Unix – has:

- **Hardware protection:** only programs operating in what may be variously called ‘supervisor’, ‘administrator’ or ‘monitor’ mode can access certain areas of memory, address hardware devices directly or execute ‘privileged’ machine code instructions.
- **File protection:** users are prevented from modifying or reading files outside their own directory; in particular system files such as the Windows registry can only be updated by authorised users.
- **Patching:** newly discovered security vulnerabilities are remedied with patches sent to home users and systems administrators. Machines can be configured to automatically apply patches.

7.4.5 Patching, responsible disclosure, bug bounties and exploit Wednesday

The development of a robust system of patching took some time. Bruce Schneier writes (Schneier, 2018, Chapter 2 *Patching is failing as a security paradigm*):

There are undiscovered vulnerabilities in every piece of software. They lie dormant for months and years, and new ones are discovered all the time by everyone from companies to governments to independent researchers to cybercriminals. We maintain security through (1) discoverers disclosing a found vulnerability to the software vendor and the public, (2) vendors quickly issuing a security patch to fix the vulnerability, and (3) users installing that patch.

It took us a long time to get here. In the 1990s, researchers would disclose vulnerabilities to the vendors only. Vendors would respond by basically not doing anything, maybe getting around to fixing the vulnerabilities years later. Researchers then started publicly announcing that they had found a vulnerability, in an effort to get vendors to do something about it – only to

have the vendors belittle them, declare their attacks ‘theoretical’ and not worth worrying about, threaten them with legal action, and continue to not fix anything. The only solution that spurred vendors into action was for researchers to publish details about the vulnerability. Today, researchers give software vendors advance warning when they find a vulnerability, but then they publish the details. Publication has become a stick that motivates vendors to quickly release security patches, as well as the means for researchers to learn from each other and get credit for their work; thus publication further improves security by giving other researchers both knowledge and incentive. If you hear the term ‘responsible disclosure’ it refers to this process.

When first used, patches for operating systems could themselves be buggy, hastily written and not properly tested. However, over the years, vendors have improved their testing regimes, in particular Microsoft, Apple and Linux, the companies that supply the operating systems that most of the world uses. It has been more than a year since this author had to roll back a Windows update that caused more problems than it solved.

Microsoft started patching with Windows 98, released in 1998, which provided access via Internet Explorer to the Windows Update centre – this checked to see if users had the most up-to-date Windows components and hardware drivers. Over time this became part of the Windows operating system, with the facility of automatically applying software updates, including patches for security vulnerabilities. Today Microsoft sends out Windows updates on Tuesdays, known as patch or update Tuesday, with the following day known as exploit Wednesday, as this will be the day that hackers and malware writers try to take advantage of unpatched systems.

Many hardware and software vendors operate bug bounty programs, in the hope that this increases the likelihood that they will be the first to know about recently discovered vulnerabilities, hence avoiding zero day exploits (that is, attacks exploiting vulnerabilities that the vendor does not know about). In 2017, Microsoft announced their latest program, with bounties from \$500 to \$250,000, see: <https://blogs.technet.microsoft.com/msrc/2017/07/26/announcing-the-windows-bounty-program/> There is disagreement about whether bug bounty programs encourage responsible disclosure or reward extortion. Responsible disclosure means that researchers tell the vendor first of any security issue they discover, without notifying the public. This is because notifying the public also notifies criminals. The researcher and the vendor agree a period of time before disclosure to the public is made, giving the vendor time to develop a patch.

In general, supporting operating systems is very expensive; hence companies limit themselves to supporting only the most recent versions, meaning that patches are only available for recent versions of operating systems. In the UK various NHS trusts were vulnerable to the 2017 WannaCry attack not just because they had not applied the security patch that would have prevented the attack, but also because some were using the unpatchable Windows XP, which Microsoft had stopped supporting in 2014. Following the WannaCry attack, Microsoft issued a fix for unsupported versions of Windows including XP, but this was intended to be a one-off exception.

Despite the security risks, there are still vendors who can be slow to issue patches, and users who can be slow to apply them. For example, *The Register* reports that many manufacturers do not patch all Android bug fixes notified by Google in a timely way, and sometimes not at all. See: www.theregister.co.uk/2018/04/13/slow_android_security_fixes/.

7.4.6 Responsible disclosure and the chilling of research

Responsible disclosure can only work effectively if it is supported by the computing industry. Since so much of the industry is based in the USA, law and practice in that country affect all of us. Bruce Schneier claims that due to push back from the industry, many capable and talented people who could be working on finding vulnerabilities, are not, leaving it to criminals who have no such inhibitions:

In order for [responsible disclosure] to work, we need security researchers to find vulnerabilities and improve security, and a law called the Digital Millennium Copyright Act (DCMA) is blocking those efforts. It's an anti-copying law that [...] includes a prohibition against security research. Technically, the prohibition is against circumventing product features intended to deter unauthorized reproduction of copyrighted works. [...] Since software can be copyrighted, manufacturers have repeatedly used this law to harass and muzzle security researchers who might embarrass them.

One of the first examples of this harassment took place in 2001. The FBI arrested Dmitry Skylarov at the DefCon hackers conference for giving a presentation describing how to bypass the encryption code in Adobe Acrobat that was designed to prevent people from copying electronic books. [...] In 2011, Activision used it to shut down the public website of an engineer who had researched the security system in one of its video games. There are many examples like this.

[...]

The chilling effects are substantial. Lots of security researchers don't work on finding vulnerabilities, because they might get sued and their results might remain unpublished. If you're a young academic concerned about tenure, publication and avoiding lawsuits, it's just safer not to risk it.

Schneier, 2018, Chapter 2 *Patching is failing as a security paradigm*

On 19 April 2019 Hutchins pled guilty to two charges, in a plea deal arrangement, and was facing up to 5 years in prison on both charges. The comments below the report in *The Register*, speculated that the charges, relating to actions Hutchins took as a teenager, might be a way to blackmail him into working for the security services, see: https://www.theregister.co.uk/2019/04/19/marcus_hutchins_pleads_guilty/ In July 2019 Hutchins was sentenced to time served and released, see: https://www.theregister.co.uk/2019/07/26/hutchins_sentencing/ While Hutchins admitted to breaking the law, *The Register* notes that 'according to Uncle Sam's prosecutors, virtually all the victims of Hutchins' malware were outside America, making this whole US trial thing pretty odd.'

7.4.7 New developments and commercial imperatives

Schneier believes that the Internet of Things has introduced new vulnerabilities to the internet. Firstly so many more connected devices simply means a bigger attack surface. Then more complexity means more chances of an interaction that can lead to a new vulnerability. Equally as important, the Internet of Things has seen the entry into the computing world of manufacturers who do not understand even the most basic computer security

issues, and do not understand responsible disclosure, seeing such disclosure as an attack to be defended against, rather than an opportunity to improve:

Just like the computer vendors of the 1990s, IoT manufacturers tout the unbreakability of their systems, deny any problems that are exposed, and threaten legal action against those who expose any problems.

(Schneier, 2018, Chapter 2 *Patching is failing as a security paradigm*)

Schneier notes that the development paradigm is either to get the product right and then sell it (traditional manufacturing) or to get it working, market it, discover failures and fix them quickly (software). In the Internet of Things these two paradigms collide. Many IoT devices cannot be patched, or can be patched but only if their owner is a highly skilled computer scientist. Schneier believes that this situation will improve over time, and companies will discover the joy of patching, and even how to automatically update their embedded systems. At the same time he notes that many manufacturers do not have the resources to employ the staff that they need to harden and update their systems. In addition if manufacturers ship new devices regularly, they cannot support operating systems in all devices they have ever sold. Even companies such as Microsoft time-limit the software that they support, on the grounds of cost. Schneier notes that smart consumer products such as cars and fridges, may have an operating lifetime of 25 years or more. There is no current software that is 25 years old and still supported.

Schneier writes that:

the current system of patching is going to be increasingly inadequate as computers become embedded in more and more things. The problem is we have nothing to replace it with.

Schneier suggests the answer is to integrate the paradigm of getting it right the first time with the paradigm of patching, in order to have both long term stability and 'reactive capability'.

7.4.8 Government as the problem and the solution

In 2017 *The Register* reported that 'The self-styled Shadow Brokers group has made a collection of NSA hacking tools and exploits publicly available.'

Further:

Documents leaked by intelligence whistleblower Edward Snowden provide persuasive evidence that hacking tools previously leaked by the Shadow Brokers included malware and exploits that began life at the signals intelligence agency.

See: www.theregister.co.uk/2017/04/10/shadow_brokers_open_sources_hacker_trove/

The Guardian reported:

WannaCry malicious software has hit Britain's National Health Service, some of Spain's largest companies including Telefónica, as well as computers across Russia, the Ukraine and Taiwan, leading to PCs and data being locked up and held for ransom.

The ransomware uses a vulnerability first revealed to the public as part of a leaked stash of NSA-related documents in order to infect Windows PCs and encrypt their contents, before demanding payments of hundreds of dollars for the key to decrypt files.

See: www.theguardian.com/technology/2017/may/12/nhs-ransomware-cyber-attack-what-is-wanacrypt0r-20

Governments, not just in the USA, are employing the brightest people they can find to discover new hacking tools, which they are not routinely disclosing. Since governments have the resources to employ so many people, and give them the best technology to support their efforts, they are very likely to find vulnerabilities before anyone else. These vulnerabilities may later be independently discovered by responsible researchers, or by criminals.

Schneier believes that companies who drive the internet, such as Google and Facebook, want to be able to monitor users in order to profit from their data, and their desire for insecurity fits with a government that wants to be able to break security at will. Hence another reason for network insecurity is that it fits the priorities of those developing products and shaping the regulatory environment, or lack of it. Schneier believes that government is a problem, in concealing weaknesses, targeting those who reveal them and not producing a robust regulatory environment that enforces basic cybersecurity standards. He also believes that government regulation is the solution.

7.5 Is Windows more of a target than other operating systems?

Some users have the impression that Windows is targeted more than other operating systems, even allowing for its market dominance. There may be several reasons for that. One could be that Microsoft's historic lack of concern about security made Windows more of a target because it was easy, certainly much easier than Unix or Linux, which had good security from the beginning. This has led to a build up of expertise in targeting Windows.

Other reasons could include:

- Since OSX, the Mac OS has been based on Unix, making breaching its security challenging.
- Hackers with limited abilities ('script kiddies') may use off the shelf tools, and these are mostly available for Windows.
- High-value targets, such as corporations and governments, tend to run Windows.

Activity 7.1

Which one of the following is **not** a reason why networked computers have security vulnerabilities?

1. The original TCP/IP protocols were not designed with security in mind.
2. Popular operating systems for personal computers were not designed with security in mind.
3. Credential stealing is still a major route for unauthorised access, and is very hard to prevent.
4. Government regulation is needed to enforce best security practices, but most governments have failed to regulate.
5. The Linux operating system dominates the world wide web infrastructure, and is easily hacked.
6. The Internet of Things has led to a larger attack surface.
7. The Internet of Things, has seen the entry into the computing world of manufacturers who do not understand, and hence do not apply, even the most basic computer security.
8. The Internet of Things uses general purpose computer chips, hence it consists of millions of computers with unused capacity and poor security that can, for example, easily be recruited by hackers into botnets.

9. The paradigm of patching security issues leaves networks vulnerable to attack, as hackers will try to exploit known security vulnerabilities in unpatched systems.
 10. The internet is very complex, and gets more complex every day. Complexity means unintended side effects that are hard to predict, even for experts, and can include security vulnerabilities.
-

7.6 How hackers hack

The improved security features of modern operating systems have not caused virus writers and hackers to give up. The challenge now is to find ways to usurp supervisor or administrator privileges, which can lead to unrestrained access to files and memory.

7.6.1 Use the credentials of a legitimate system user

Hackers hack by exploiting vulnerabilities in systems, which can take time and be technically demanding, particularly when the hacker is looking for new vulnerabilities. Much easier is to exploit known vulnerabilities, easier still is using the log-in details of an existing user, and the easiest hack of all is asking a system user for their password.

Bruce Schneier (Schneier 2018, *Authentication is getting harder, and credential stealing is getting easier*) notes that:

In 2016, Rob Joyce, then head of the NSA's since-renamed Tailored Access Operations (TAO) group – basically the country's chief hacker – gave a rare public talk. In a nutshell, he said that zero-day vulnerabilities are overrated, and credential stealing is how he gets into networks.

He's right. As bad as software vulnerabilities are, the most common way hackers break into network is by abusing the authentication process. They steal passwords, set up man-in-the-middle attacks to piggyback on legitimate log-ins, or masquerade as authorized users. Credential stealing doesn't require finding a zero-day or unpatched vulnerability, plus there's less chance of discovery, and it gives the attacker more flexibility in technique.

Or hackers can simply ask for your password, known as 'social engineering'. For example, they call a legitimate user pretending to be from their ISP or employer's IT department, and ask the user to tell them their password. Reformed hackers have testified that a remarkably effective way to get confidential information is simply to ask for it.

Users tend to prefer passwords that are easy to remember, so they may pick their child's name, their partner's name, car licence number, etc. Hackers can therefore exploit personal knowledge.

Hackers can attempt to discover passwords with brute force, by writing a program that makes successive login attempts using every string in an electronic dictionary as a password. This can be countered by disabling logins for a particular user after a fixed number of consecutive failed attempts. While most systems limit login attempts, surprisingly brute force attacks can still work on systems with poor security.

Phishing is a slightly more sophisticated way of asking for passwords, and it works. Verizon estimated in their 2017 'Data Breach Report' that 7.3 per cent of users fell for phishing emails, whether via opening an attachment or clicking

on a link, and this is in line with the findings of other researchers.

See: <https://enterprise.verizon.com/resources/reports/dbir/>

Some installations require that users change their passwords at regular intervals. It is questionable whether this enhances security; there is a danger that users may run out of ideas and choose passwords which are easier to guess. On the other hand it clearly makes good sense to choose a password that is not a word in the dictionary and is too obscure for hackers to be able to guess, though not so obscure that the user finds it hard to remember.

Another danger is that people have so many passwords to remember for work, leisure, shopping, banking, etc. that they resort to using the same password for every login. Hackers know this, and once they have somehow discovered a user's email address and email password, will attempt to login using these details on various sites.

7.6.2 Persuade the user to hack themselves

Where organisations have good security, criminals may try to persuade the user to carry out the actions necessary to hack. For example, criminals will target retail banks' customers, rather than the retail banks, with various schemes, some of them very sophisticated, that attempt to persuade customers that they need to transfer their money to an account of the criminals' choosing. For example, one scheme involves phoning a target, and persuading them to give the criminal the telephone number on the back of their bank or credit card. The criminal then phones again, spoofing that number, so that the target sees this number on their caller identification and thinks they are talking to their bank/credit card company. Naturally the target is told to transfer money to the criminal's account, with the idea that they will comply, believing this to be a legitimate request.

7.6.3 Arbitrary code execution exploit

An arbitrary code execution exploit is a vulnerability in programming languages that can be exploited by skilled hackers to install malware. If the attack is carried out over a network, it is known as remote code execution exploit.

A buffer overflow attack is a sophisticated technique, and an example of an arbitrary code execution exploit. The 'overflow' happens when a program accepts input that is larger than the space reserved to store it (the buffer) and some of the data overwrites the executable part of the program after the buffer. This data will then become part of the program and is likely to execute undesired functions. This technique was used in one of the very first exploits, the Internet Worm, and is still an attack vector today.

Buffer overflow attacks assume a combination of careless programming in the target system, allowing input data to be stored after the end of the buffer, and clever programming by the virus writer who will need very detailed knowledge of the target system's software. This technique can be used for unauthorised access and for installing malware. Many programming languages are vulnerable to buffer overflow attacks, with C and C++ particularly so.

Dedicated hackers are always looking for new arbitrary code execution exploits, or ways into systems.

7.6.4 Opportunism

Hackers may track security updates for operating systems, and then use those vulnerabilities to attack sites and servers that have not patched the issue. For

example, the WannaCry ransomware attack of 2017 was only an issue for those who had not applied an emergency patch issued by Microsoft in April 2017. In the UK this included many NHS local health trusts, despite NHS Digital (the NHS's IT centre) issuing critical alerts.

In 2003, Microsoft began sending out patches together on a Tuesday, rather than rolling them out as and when. 'Patch Tuesday' led to 'Exploit Wednesday', a day of attacks based on the vulnerabilities exposed the previous day. By the next day attackers have had time to consider the security patches, and how to use the vulnerabilities they exposed to attack unpatched machines.

Attacks based on known vulnerabilities may be an issue with Android phones, as some manufacturers do not patch all bug fixes notified by Google in a timely way, and sometimes not at all.

Exploit kits are available for sale to hackers. They bundle together known security vulnerabilities into a point and click interface, allowing even unskilled users to exploit known security issues for personal gain or other criminal purposes.

7.6.5 Session hijacking (AKA cookie hijacking) and packet sniffing

Hackers may intercept packets sent in a data stream, in order to gain unauthorised access. In particular attackers may be looking for a cookie that is being used to authenticate a user on a website they are visiting, allowing them to take control of the session. There are several ways to find and steal cookies, but the simplest, that this attack often exploits, is that data streams: (1) may be sent as unencrypted data; and (2) data streams can easily be intercepted in transmission, particularly over wifi networks.

Packet sniffing software, used to intercepts packets in a network data stream, has many legitimate uses in network security and monitoring, but the software can also be used for session hijacking, and to collect other sensitive information such as log in details, depending on the encryption used (or not used).

7.6.6 Creating a backdoor

A hacker may get access once through a stolen password, but cannot rely on continued access via the same route as passwords change and users move on. Or they may have gained access through a security loophole that will be patched at some point. To guard against this, hackers can install backdoors that let them into the system without the knowledge of the system administrator. The sysadmin may know that the network was hacked, but they may not know about – or check for – a backdoor.

7.6.7 Browser plug-ins and extensions

Browser plug-ins were developed at a time when browser development was limited, frustrating web designers who were looking for greater interactivity. Java applets (now deprecated) were particularly used to provide interactivity, and became a notorious security problem. Plug-ins are much less of a problem than they were. With HTML5, functions once carried out by browser plug-ins are being incorporated into web pages directly, limiting the need in future for plug-ins. In addition many browsers do not allow their use, or only allow the use of a limited number of particular plug-ins, hence hackers are targeting extensions. Kaspersky notes:

First of all, extensions can be downright malicious. That happens mostly with extensions that come from third-party websites, but sometimes — as in cases with Android and Google Play — malware sneaks into official markets as well.

For example, security researchers recently uncovered four extensions in the Google Chrome Web Store that posed as innocuous sticky notes apps but in fact were caught generating profits for their creators by secretly clicking on pay-per-click ads.

[...]

Browser extensions are an interesting target for crooks, because a lot of extensions have massive user bases. And they are updated automatically, which means that if a user had downloaded an innocuous extension, it can be updated to become malicious; that update would be pushed to the user right away — and the user won't notice anything at all.

A good developer won't do such a thing, but their account can be hijacked and a malicious update can be uploaded to the official store on their behalf. That's what happened when crooks used phishing to get the access credentials of the developers of a popular plugin* called Copyfish. In that case, the plugin, which originally performed optical character recognition, was used by crooks to serve additional ads to users.

Sometimes, developers are approached by companies that offer to buy their extensions for a rather tidy sum. Extensions are usually hard to monetize, which is why developers are frequently eager to agree to such deals. After the company purchases the extension, it can update it with malicious features, and that update will be pushed to users. For example, that's exactly what happened to Particle, a popular Chrome extension for customizing YouTube that was abandoned by its developers. A company bought it and immediately turned it into adware.

See: <https://www.kaspersky.co.uk/blog/browser-extensions-security/12750/>

**Note that the author of the web page has mistakenly called Copyfish a plug-in, when it is actually an extension. In fact the author seems to use the terms 'extension' and 'plug-in' interchangeably (see glossary for the difference) but the blog post is all about extensions. You can read more about Copyfish here:*

<https://thehackernews.com/2017/07/chrome-extension-hacking-adware.html>

Extensions may collect data about users, which can lead to privacy issues. One example is an extension that collected the web browsing history of users, which was later sold on, incompetently anonymised, by the owners. Poor anonymisation meant that users could be identified from their browsing histories. See: <https://www.pcworld.com/article/3139814/web-of-trust-browser-extensions-yanked-after-proving-untrustworthy.html>

7.6.8 Hacking mobile phones

Wired reported, in October 2018, on a hacking device that can be plugged into an iPhone to unlock it. Three UK police services had bought the device from a company called *Grayshift*:

The company, which has a former Apple engineer as a senior member of staff, emerged at the end of 2017 and can reportedly unlock iPhones running everything up to Apple's latest operating system, iOS 12.

[...]

[The company] sells a small box, called GrayKey, that plugs into an iPhone and unlocks it. To do this it exploits alleged vulnerabilities in Apple's software. Once an iPhone has been unlocked it is possible to access everything on the device, including messages, photos and activity logs. GrayKey has proved to be a significant headache for Apple, which has improved security on its phones in an attempt to block it.

See: www.wired.co.uk/article/police-iphone-hacking-grayshift-graykey-uk

In addition to needing physical access to a phone to hack it, *The Guardian* reports that it is very easy to turn a phone on remotely, and to use it to spy on you, something that both governments and hackers do. Governments access phones through built in back doors, while:

Hackers can also gain access to your device with extraordinary ease via apps, PDF files, multimedia messages and even emojis.

An application called Metasploit on the ethical hacking platform Kali uses an Adobe Reader 9 (which over 60% of users still use) exploit to open a listener (rootkit) on the user's computer. You alter the PDF with the program, send the user the malicious file, they open it, and hey presto – you have total control over their device remotely.

See: www.theguardian.com/commentisfree/2018/apr/06/phone-camera-microphone-spying

7.6.9 The Internet of Things

Internet of Things devices have low security standards; for example, using default passwords which the user cannot change. Many IoT devices have low to no security standards to begin with, are rarely or never patched, and there are billions of them. In 2015, *The Register* reported that a Samsung smart fridge was leaving the Gmail credentials of owners open to theft, see: www.theregister.co.uk/2015/08/24/smart_fridge_security_fubar/

7.7 Definition of malware

Malware, a contraction of **malicious software**, is developed by hackers to steal data, commit fraud and extortion and cause harm to computers and networks. Over time, as the internet has increased in size and penetration into our daily life, so has malware. In this section we will discuss types of malware, together with hacking tools.

Malware may be delivered via email, downloaded from a web page or distributed via removable media. Since the Microsoft store was launched, Windows users are much less likely to download malware from web pages, so it is likely that the store itself is a target of hackers and cybercriminals, in terms of thwarting Microsoft's security measures in order to place malware in the store.

7.7.1 Not malware: Potentially unwanted programs (PUPs), cookies and spyware

PUPs are **potentially unwanted programs**. Also known as PUAs – **potentially unwanted apps**. These are programs that get installed when the user downloads another program. In the licence agreement of the legitimate program is a statement that the user agrees to the installation of the PUP. Hence the user has technically agreed to the download and installation, although most people will click on the licence agreement without reading beyond the first page, if that. Hence PUPs are not malware, as they are installed with the user's permission. Normally PUPs are adware or spyware.

Cookies are placed on a machine without user intervention, similarly to malware. Third-party cookies have been used to track browsing habits, but now must be notified to users, and agreed by them, at least in the EU. Third-party cookies come close to malware, as does spyware, but Gillespie notes (2015, Chapter 3, *Targeting the technology* section *Spyware*) that spyware (and tracking cookies) invade privacy rather than causing harm or stealing data. Gillespie further notes that the situation is complicated by an overly broad definition, and by legitimate companies such as, 'for example, Google, Microsoft and Apple' all possibly using spyware at one time or another. Some authorities may include in their definition of spyware stealing bank details and credentials, and this would tip it into malware. Gillespie believes it is more useful to define spyware as invading privacy and more a matter for civil law than criminal, and not to label malware that commits obviously criminal acts such as stealing bank details as spyware.

7.7.2 Types of malware

Virus is used as a generic term for malicious programs which are spread by some form of user action (such as sharing an infected file, or opening an infected email attachment) and normally infect executable files, such that when the infected program is run, so is the virus. Viruses can also infect documents and the master boot record. The name reflects that computer viruses have the ability to copy themselves, and once on a machine viruses will attempt to spread from program to program, or from file to file. Viruses have a payload, the action that they are programmed to take, which can range from doing something annoying but harmless, to reformatting your hard drive and hence destroying all your files.

Worms are programs that can transmit themselves through a network by exploiting security loopholes, as opposed to viruses proper which need a 'host', and which require human intervention to spread. Worms, like viruses, self-replicate, but are autonomous and do not attach themselves to programs or files. Worms also have a payload, which can range from just replicating themselves, to infecting a computer with ransomware.

Trojan horses or **trojans** masquerade as legitimate programs such as games or utilities. Users download and install them believing that they are installing legitimate software. Trojans cannot replicate themselves. The name 'Trojan' comes from the ancient myth of the battle of Troy, when the Greek forces were able to invade the city by concealing themselves in a giant wooden horse. The Trojans thought this was a gift from the departing Greeks and dragged it inside the city, and thus inside their defences, whoops.

Logic bombs have payloads that are triggered by a particular date or event. Delaying the payload may be done to give malware time to spread before being discovered and targeted by anti-virus vendors. Logic bombs are unusual malware in that they are often used to compromise systems that the user

has legitimate access to as time or event triggers have obvious applications for those enacting revenge on employers and others. The classic example is deleting all files once a certain person is removed from the employees' database. Viruses and Trojans often incorporate logic bombs. Hence logic bombs may be malware installed by a legitimate user of the system, but they can also be the payload of other malware, such as a virus, worm or Trojan.

Time bombs are logic bombs that trigger their payload on a particular date. A time bomb may be legitimate software that stops working after a set time in order to allow customers to trial the software before purchase, or in order to prevent testers from continued use of beta versions.

Ransomware is a type of malware that extorts money from users, usually as bitcoin or other forms of virtual payment. Common forms of ransomware steal or encrypt data, including sensitive data and photographs, and threaten to publish or destroy the data. Some authorities think that the biggest threat today is ransomware that encrypts the user's files. Some users do get their data returned after paying a ransom, most do not, either because the criminal had no intention of so doing, or because the originator of the attack is using tools they do not really understand, and does not know how to decrypt. Ransomware can be spread in various ways, such as by worms and Trojans; it can also be sent as an email attachment or contracted from an infected web page. It may be used as a cover for other attacks, where the intention is data stealing or destruction. Note that ransomware is not a virus, because it does not have the ability to copy itself, and is an unusual form of malware, as the user of the target system will be notified that they have been infected.

Adware is defined by Kaspersky as 'programs that are designed to display advertisements on your computer, redirect your search requests to advertising websites and collect marketing-type data about you – for example, the types of websites that you visit – so that customised adverts can be displayed.' Adware might be a PUP, that the user has theoretically agreed to, but if it is installed without the user's consent, for instance via a Trojan, or via a visit to an infected website, then it is malware. See: www.kaspersky.co.uk/resource-center/threats/adware

Blended threats are attacks made with a combination of different kinds of malware, including viruses, worms and Trojans. A blended threat often exploits known vulnerabilities, and employs multiple attack vectors with the intention of doing real harm as quickly as possible. Symantec describes blended threats differently to other authorities, describing it as 'a natural, accidental, or purposeful combination of a physical with a cyber incident'; for example, a hurricane hits an area followed by a cyber attack shutting down critical infrastructure. See: www.symantec.com/blogs/expert-perspectives/surge-blended-attacks-stirs-new-cyber-worries

Botnets: A bot (short for 'robot') is an automated network process, often used for legitimate tasks. However, bots can be installed on target machines and used to recruit them into a botnet – a network of bots. Machines in the botnet are often called 'zombies'. The person or organisation controlling the botnet usually takes care that the owner of the infected machine does not know that they have been infected. Since the botnet may only use a small part of each machine's bandwidth, it may be possible to hide the infection for some time. Botnets are typically used for DDoS attacks and for sending spam, but may also be used for other types of fraud including click fraud, or may be used to mine bitcoin.

Botnets can be controlled by a command and control server that treats each infected machine as a client. Command and control servers could once exist for some time, even years, but now they are a target of security specialists

and tend to be discovered quite quickly. Because of this there are now peer-to-peer botnets.

Rootkits: a collection of software tools, developed by hackers to take control of a computer or network with administrator level access. Once a hacker has gained access to a system or machine they may attempt to install a rootkit, or a **dropper may infect a machine in a drive-by attack** (see Section 7.8.3), and open the door to the rootkit. Rootkits can be bought, and Schneier reports that they can 'give even unskilled hackers enormous capabilities' (Schneier 2018, Chapter 1, heading *Computers fail differently*). Rootkits themselves are not malware, as they can and do have legitimate uses for security specialists and others. They are very hard to detect and remove.

Bootkits: A bootkit is a rootkit that overwrites some of the boot record, such that it is loaded every time the system is rebooted. Bootkits are malware, and are even more difficult to detect and remove than rootkits.

Exploit kits are sophisticated software tools, that bundle together known security vulnerabilities into a graphical interface that allows even fairly naive users to point and click in order to utilise the tools available for complex attacks. Sophos reports that:

An exploit kit is usually delivered directly into a potential victim's browser in the form of convoluted and hard-to-follow JavaScript, and automatically tries out a series of attacks, typically in the most likely sequence, until one of them works, or they've all failed.

See: <https://nakedsecurity.sophos.com/2016/06/16/is-angler-exploit-kit-dead/>

7.7.3 Types of viruses

Most viruses attach themselves to executable files so that when the file is run, they are too. There are two distinct types of viruses that do not follow this pattern.

Boot sector virus. During the early years of personal computers, viruses were mostly passed by, now obsolete, floppy disks. The first personal computers did not have a hard disk, but were booted up with floppy disks. After the operating system was loaded, another floppy would be used to load the software the operator intended to use. When a PC is started the CPU loads the bootstrap software to initialise the operating system. If the PC is booted from an infected disk the virus will be loaded and executed too. Even after computers normally had hard drives, the CPU would boot from the first available drive in alphabetical sequence, and this would be the A-drive – the floppy disk drive (the first personal computers had an A-drive for 3.5 inch floppy disks, a B-drive for 5.25 inch floppies, hence when hard drives were added they naturally became the C-drive). Therefore, if a disk was in the drive when the computer was turned on (always a possibility) the PC would attempt to boot from the floppy, and the virus code might be loaded and executed. If the machine had a hard drive, the virus would infect its master boot record, meaning that it would be loaded and executed every time the machine was started up.

Once executed, the virus could infect other floppy disks loaded by the machine, by using MS-DOS commands. Since boot sector viruses used MS-DOS commands to spread to other floppy disks, they declined sharply after Windows 95 (released in 1995) began to dominate the market. Boot sector viruses still exist, and can be spread by removable media or email. Today they can encrypt the boot sector and the hard drive might need reformatting to get rid of them.

The macro virus. Macros are ‘shortcuts’ which can save typing commands or text in applications such as word processing. Some simple uses are adding a standard footer to a Word document, or a macro to add standard data into a spreadsheet. To make a macro the user can simply run through the commands that they want to automate, while recording them. The recorded commands are replayed each time the template document is used to create a new document. Visual Basic is the programming language used to save macros, introduced by Microsoft in 1991, with support discontinued in 2008, although Microsoft applications still allow its use.

At first Microsoft applications enabled macros by default. This meant that when a user opened a file, the macro would execute. Malware writers quickly saw the possibilities, particularly as Microsoft Office files were often shared via email, and in the 1990s and early 2000s macros were one of the most common ways to spread malware. After Windows brought in greater security measures, combined with users developing more awareness of threats via email, malware writers moved on.

Today macros have made a comeback, often used as the first stage of an attack, with the aim of infecting a machine with a Trojan or with ransomware. For example, Locky ransomware, a sophisticated attack discovered in 2016 that locked users out of their files, was spread by macros. Avast, an anti-virus vendor, reports:

The malware spreads through fake emails and infected attachments, including .doc, .xls or .zip files. The opened documents don’t display correctly, and the user is asked to “enable macro if data encoding is incorrect”. This is a social engineering technique used to trick people, because once they enable macros, a binary file starts running and downloading Locky.

See: www.avast.com/c-locky

One reason macro viruses have returned is that once a user runs a macro, they are authorising it to do anything on the system that they themselves are allowed to do, including downloading and installing files.

The script virus. These are viruses written in scripting languages such as JavaScript and Visual Basic for Applications (VBA). There is overlap with macro viruses that may be written in VBA. Malicious JavaScript applications can be sent via email, with Sophos reporting in 2016 that JavaScript viruses were being used to fetch ransomware and were sent via email as attachments. These emails would purport to send such things as documents relating to a court appearance or a delivery of goods, as a zip file. The zip file, once opened in Windows File Explorer, might contain documents such as *OpenMelmHarmless.pdf.js*, but if the user’s system was set up to hide extensions then they would see *OpenMelmHarmless.pdf*, and, in a moment of inattention might click on the document, particularly as the JavaScript icon is a scroll, which looks like a document. See: <https://nakedsecurity.sophos.com/2016/04/26/ransomware-in-your-inbox-the-rise-of-malicious-javascript-attachments/>

Web scripting virus. These are viruses that infect web pages, usually written in JavaScript. Since JavaScript has been incorporated into HTML5 it has become a particular target of malware writers, with anti-virus vendors seeing an increase in JavaScript viruses since 2016. Web scripting viruses are used in **drive-by attacks** (see Section 7.8.3), and also in **malvertisements** (see Section 7.8.5), where malicious code is hidden in online adverts, often on legitimate sites.

Stealth virus: since viruses make changes to the size of files and the time that files were accessed, and since these changes can be searched for and detected

by anti-virus software, a stealth virus is one that can disguise these changes, but only while it is active in memory.

Polymorphic virus. This is a virus that avoids detection by using encryption to frustrate signature-based detection. Kaspersky reports that:

In 2015, it took the combined efforts of the FBI and Europol to bring down a botnet – a network of computers – running advanced polymorphic malware called Beebone. The malware was used by a criminal gang to control at least 12,000 computers around the globe and could change itself up to 19 times a day to avoid detection.

Only a year earlier, the first polymorphic, self-replicating ransomware virus was discovered. Called VirLock, it can infect files, replicate itself and change form in addition to locking the computer screen of a host computer like traditional ransomware.

See: www.kaspersky.co.uk/resource-center/definitions/what-is-a-polymorphic-virus

In 2016, Webroot, a cyber security business, reported that 97 per cent of malware was polymorphic, and in their 2017 threat report this had reduced slightly to 94 per cent. See: www.webroot.com/us/en

A polymorphic virus encrypts itself, but has to decrypt itself in order to release its payload. Such a virus propagates by re-encrypting a copy of itself and attaching it to another application. While other viruses may use encryption, in a polymorphic virus each encryption is unique, making it hard to detect the virus.

Metamorphic viruses are even harder to detect than polymorphic viruses as they can rewrite their own code. Metamorphic viruses take extensive skill and experience to write. Kaspersky describes them as follows:

A metamorphic virus is one that can transform based on the ability to translate, edit and rewrite its own code. It is considered the most infectious computer virus, and it can do serious damage to a system if it isn't detected quickly. Antivirus scanners have a difficult time detecting this type of virus because it can change its internal structure, rewriting and reprogramming itself each time it infects a computing system.

See: www.kaspersky.co.uk/resource-center/definitions/metamorphic-virus

7.7.4 Other malicious exploits

Denial of service (DoS) is not a type of virus but an exploit. DoS attacks seek to put a computer system out of action by overwhelming it with network requests, for example, by using the 'ping' command (ping is a utility used by network administrators to find out if a remote machine is accessible through the internet), sending large numbers of emails or mailing massive files ('mail bombing'). Another technique is the so-called 'two-way handshake': the initiating system sends the first packet of a TCP three-way handshake causing the target system to send an acknowledgement, but fails to send the third packet. The effect of this is that following TCP specifications, the target system has to send repeated acknowledgements until it reaches the limit defined by the network administrator.

Distributed DoS attacks are launched by large numbers of computers simultaneously, controlled in a botnet. Computers in the botnet are often known as zombies. It can be difficult for enterprises to protect themselves

against DoS attacks; for instance TCP/IP protocols require network-layer software to respond to every 'ping' request, and so a system receiving large numbers of 'pings' in a short time will have its CPU time tied up in sending responses. In 2016 the Mirai botnet was formed of Internet of Thing devices, and was used in a massive DDoS attack that restricted internet activity on a large scale in Europe and North America. Internet of Things devices are an increasing target for botnets, since such devices are often always on (your fridge for example), have low security standards (default passwords), are rarely or never patched, and there are billions of them.

Malicious emails: Criminals practise spoofing, sending emails that appear to be from a trusted source. Such emails may:

- be phishing emails. Phishing messages purport to come from trusted sources such as banks, or social media sites, but are really attempts to steal log-in credentials, bank and credit card account numbers. Phishing emails will include a link that looks genuine, and if clicked on will take the user to a fake website that looks like the real thing, where they will be asked to log-in. Users may not realise that the web page is fake, and that their credentials are now compromised.
- include a link to download malware, usually with a spoofed URL that looks genuine to the user at first glance.
- have attachments that are infected with malware, but are disguised as something benign.
- be an attempt to get the user to click on a link in order to find the address of machines that are in a private network.

Brute force attacks: Trying as many different passwords as possible, using an automated process. Most systems will guard themselves against such attacks by cutting off users after a certain number of failed attempts to log on.

Credential stuffing: Criminals use stolen credentials to attack sites other than the ones that the credentials were stolen from. This works because of the habit many users have of reusing passwords, and is similar to a brute force attack in that it is done in bulk and automated.

Credential (password) spraying: attackers attempt to circumvent protection from brute force attacks by using a restricted range of common passwords. Attackers attempt to gain access by harvesting as many log in IDs as possible for a particular web page or organisation, and then attempting to log in by using each ID in turn with the same password. This can be done for each password on their list, and by changing the log in ID with each attempt, rather than the password, attackers evade protections from brute force attacks.

7.7.5 Malware developments and current threats

Fileless malware is a relatively recent threat. Kaspersky reports:

Fileless malware is malware that does not store its body directly onto a disk. This type of malware became more popular in 2017 because of the increasing complexity of its detection and remediation. Although such techniques were limited to targeted attacks in recent years, today they proliferate more and more in the current threat landscape, and Kaspersky Lab registers new families of trojan-clickers or even adware with fileless components.

See: www.kaspersky.com/enterprise-security/wiki-section/products/fileless-threats-protection

Fileless malware is malware that does not install itself on the hard drive but is resident in memory. Kaspersky has identified fileless malware that uses the Windows PowerShell, an administrative tool and part of the operating system, to be resident in memory. Since Windows tools such as the PowerShell are trusted applications, they are whitelisted by anti-malware vendors.

McAfee maintain a threat centre at: www.mcafee.com/enterprise/en-gb/threat-center.html/ In June 2019 four of its top 10 threats were types of ransomware. The others were four recently exposed software vulnerabilities and two spear-phishing campaigns – **spear-phishing** means targeting particular individuals for phishing.

7.8 Gaining access

Before a virus can do any damage it has to get onto your system. Virus authors use various techniques for penetrating systems, some requiring sophisticated systems knowledge and programming techniques and others not. The most popular way is to persuade users themselves to do the job, by making the virus look like something harmless or even desirable. While malware can be spread via removable media, the most popular way for malware to gain access to a machine is via emails and via the world wide web.

7.8.1 Removable media

Kaspersky reports that:

In 2016, researchers from the University of Illinois left 297 unlabelled USB flash drives around the university campus to see what would happen. 98% of the dropped drives were picked up by staff and students, and at least half were plugged into a computer in order to view the content. For a hacker trying to infect a computer network, those are pretty irresistible odds.

See: <https://securelist.com/usb-threats-from-malware-to-miners/87989/>

Kaspersky reports that removable media are still a source of infection, particularly for cryptocurrency mining software, with some users infected for years without knowing. Other threats spread by removable media include Trojans and banking malware, with emerging markets the most vulnerable to infection via this route. Kaspersky reports that Stuxnet (CVE-2010-2568), the worm that famously did serious damage to Iran's nuclear program in 2010, was spread by removable media. The report goes on to compare the threat from removable media to web-based threats:

These numbers pale in comparison to web-borne threats: in 2017, Kaspersky Lab's file antivirus detected 113.8 million likely removable media threats, while its web antivirus repelled just under 1.2 billion attacks launched from online resources. In light of this, it can be easy to overlook the enduring risks presented by removable media, even though around four million users worldwide will be infected in this way in 2018.

7.8.2 Email

Email has become more secure over the years, but is still the most popular malware vector worldwide. Email may contain a link that will take the user to a site that will infect their machine with malware, or attempt to steal personal financial information (phishing emails) or it may contain attachments that are infected, or both. In any case the attacker will try to make the email look benign, and/or from a trusted source by spoofing the email header records.

At one point users of Microsoft Outlook could be infected just by opening an email, as the client would automatically run any JavaScript in the message, and this could be compromised and infect the machine. In fact the preview pane would preview the first new message in the list, and this could run any script embedded in the HTML code, potentially infecting the machine, without the user even opening a message. Hackers also took advantage of the Windows hidden file extension, to attach files such as *ATotallySafeDocument.doc.exe* – if file extensions were hidden, the user would think they were opening *ATotallySafeDocument.doc*; however, Microsoft quickly changed Outlook settings to always show extensions.

Today's email clients practice image blocking, because spammers can use images (which are downloaded once the email is opened) to know whether your email address is live, or not. Image downloads may also reveal your IP address, a privacy issue, and it may also be that it is not an image that is being downloaded, but malicious code. Images that are inside an email message are also blocked as a precaution since such images are sent as attachments, and automatically opening attachments is a clear vulnerability.

Despite greater email security, meaning that today the user has to do something to get an infection via email, it is still the most popular vector. Attackers can easily send millions of email worldwide, and only a small percentage of users need to have a moment of inattention or thoughtlessness for criminals to make large profits. *The Guardian* reported that £130 billion was stolen worldwide by cybercriminals in 2017, including phishing, ransomware and other online attacks, see: www.theguardian.com/technology/2018/jan/23/cybercrime-130bn-stolen-consumers-2017-report-victims-phishing-ransomware-online-hacking

Criminals are also very good at picking up on trends, as this 2018 press report from Kaspersky notes:

[C]riminals have been following a global agenda by using hot topics such as FIFA 2018 and Bitcoin to fool users and steal their money or personal information in the last 12 months.

Spammers have shown themselves to be thoughtful actors, instantly monitoring global issues and major events worldwide with one main purpose – to capture and capitalize on their victim's attention. [...].

While the world was intensively preparing for FIFA 2018 last year, spammers have been actively spreading related emails. Thus, they've been sending victims fraudulent messages with official logos of the event, including organizers and sponsor brand information, and notifying users about lottery wins and even promising free tickets.

Another hot spam and phishing topic in 2017 was cryptocurrency –as Bitcoin's price drastically increased. [...]

According to Kaspersky Lab's discoveries, criminals have been using tricks such as websites disguised as cryptocurrency exchanges or fake services offering cloud mining (i.e. the use of specialized data centers for rent). [...]

Moreover, criminals have distributed different types of malware in spam emails, under the guise of utilities for earning Bitcoins, or instructions for cryptocurrency trading.

See: https://usa.kaspersky.com/about/press-releases/2018_fifa-2018-and-bitcoin-among-2017-most-luring-topics

The report also notes that phishing attackers are starting to use HTTPS; namely, secure sites, so that the advice to users to check that any link asking for personal data took them to a secure site, is no longer helpful.

7.8.3 The world wide web: drive by downloads

Viruses and other malware can be downloaded from the web, and not just by users clicking on links in malicious emails. Users can search for software, and be fooled into downloading malware. Or the user dismisses a pop up on an infected web page, but the click starts malware downloading. Malicious websites can ask the user to download malware, often by suggesting that the user needs to install a codec in order to view a video. Alternatively the user may download legitimate software, but malware comes attached. There are 'drive-by' downloads that take advantages of known and unpatched security issues on sites, hence just by opening and viewing the web pages the viewer becomes infected. Often the drive-by download is only some small piece of code (the dropper) that is quickly downloaded, but it makes the infected machine contact another machine and this sends the malware. One serious issue with drive-by attacks, is that often the web page is a legitimate page, whose owners do not know it has been infected.

The infected code is probably hidden in JavaScript, which loads and runs when the user opens the web page. In fact as HTML5 is integrated with JavaScript, and has HTML5 has come to dominate the web, this has made JavaScript a particular target for hackers and malware writers. In another way HTML5 and JavaScript may have improved web page security, by making many plug-ins redundant. Plug-ins are notoriously insecure, plus the more software a system has, the bigger its attack surface.

Sites used for drive-by downloads are often infected by use of exploit kits, described by Kaspersky as follows:

These kits contain software designed to run on web servers and identify software vulnerabilities on machines and web browsers to determine which systems are ripe for the plucking.

See: www.kaspersky.com/resource-center/definitions/drive-by-download

Often the infected sites are those popular on social media. Links get shared with friends by friends, they are trustingly clicked on and the friend gets unknowingly infected while viewing the amusing cat video.

7.8.4 The Windows Store

Now that the Windows Store is operating, providing a one-stop shop for apps that have been tested and approved by Windows, users are much less likely to download malware, as they are less likely to search the web for the software that they need. However, criminals and hackers are likely to be targeting the Windows Store, and in fact the Windows Store has been accused of containing an app (now removed) that downloaded adware, and in February 2019 Symantec reported that they had found 8 apps in the store that 'surreptitiously use the victim's CPU power to mine cryptocurrency' see: <https://www.symantec.com/blogs/threat-intelligence/cryptojacking-apps-microsoft-store-for-details>. The Google Play store is definitely a target, with malware regularly being discovered masquerading as legitimate apps.

7.8.5 The world wide web: malvertising

Criminals have realised that it is possible to infect an advertisement with malware and send it out into the world wide web via networks that accept ads for placing on other websites. Many owners of web pages provide their own content, but sell space to companies that place adverts on the web. Third-party resellers do not advertise their own products and services, but instead recruit others wishing to advertise, and maintain a network of sites that will accept their advertisements. Hence they act as the middleperson between those wishing to advertise on the world wide web; and web page owners and authors wishing to earn money from their sites. Criminals write advertisements for fake or imaginary products, services or campaigns, and place them with legitimate companies who then place them on legitimate web pages. Malverts can be very hard to detect.

7.8.6 Self-propagation

Worms do not need user intervention to spread themselves. They can transmit themselves through a network by exploiting security loopholes.

7.8.7 Hacking to install malware

Attackers may hack into systems in ways explained above in order to install malware. This attack would be highly targeted, as it is labour intensive and can only attack one system at a time. It is most likely to be used against high-net-worth targets, such as corporations.

7.8.8 How viruses install themselves

Most viruses will install themselves by attaching to the end of legitimate programs, then change the start of the program to point to the virus before the program code. When the program is invoked, the virus code will execute before passing control to the host program. This can be detected by anti-virus software by searching for program file size changes, but now viruses may hide the file size increase – making detection harder. They may also disguise the last modified date to make detection harder.

Spacefiller or cavity viruses attempt to insert themselves into blocks of unused memory inside a program file. This does not increase the file size; however, it takes skill to write such a virus, and not all programs are potential hosts, hence these viruses are rare.

Boot sector viruses. These overwrite the master boot record (MBR) meaning that they are loaded every time the machine is booted up.

Microsoft Office documents and spreadsheets can include macros; that is, scripts in languages such as JavaScript or Visual Basic, which execute when the document is opened, and are intended to save time by automating certain commands. Since Office files are routinely sent as email attachments, macros became a popular way to spread malware in the early internet. Various security measures reduced the risk, but now macros seem to be making a comeback as a popular virus vector. In Word a macro will infect the normal.dot or normal.dotm template, which is loaded every time Word is run.

7.9 The payload

Possible malware payloads include:

- installing keyloggers to read sensitive data
- searching through information for sensitive data, such as bank details and passwords. This includes searching browser programs for stored passwords and other configuration data

- sending sensitive data, files and photographs to the hacker
- ransomware that extorts money by encrypting data, or by other means such as stealing sensitive data or photographs and threatening to make them public
- recruiting the device into a botnet
- installing a dropper, malware that can download more malware
- installing software that can hijack secure sessions (man-in-the-middle malware)
- creating a backdoor for a hacker.

7.9.1 The infection

Once a virus has managed to execute on your computer, it will typically carry out some or all of the following actions:

1. Install a permanent copy of itself on your hard disk by infecting another program with its own code.
2. Propagate itself to other systems by emailing itself to contacts in your email account.
3. Deliver its payload.

A **virus** cannot spread without human interaction. It must be downloaded, installed by running infected removable media, or by opening an email attachment. Even though a virus may attempt to spread via email, a user has to open the email and click a link or download an attachment before the virus has succeeded.

A **Trojan** once installed will take the actions it was designed for, but does not propagate itself.

A **worm** can propagate itself through a network without any human intervention, by exploiting security loopholes.

7.9.2 Possible virus payloads

- Delete files or reformat the hard drive.
- Install a keylogger in order to intercept passwords or other sensitive information such as credit card details, which can then be transmitted to the virus author's machine (keyloggers).
- Install a packet sniffer to read sensitive data transmitted over a network or over the internet.
- Use email accounts to send spam.
- Do nothing, or almost nothing, although nearly all viruses modify or corrupt at least one file.
- Annoy the user but do little harm.

7.9.3 Payload of a worm

A worm may be written just to spread itself, so the damage that it does is limited to overwriting files, consuming bandwidth and increasing network traffic, which may cause congestion. However, worms may also steal data and send it over the internet, including stealing passwords and confidential documents. The WannaCry attack of 2017, that encrypted files and demanded a ransom for decryption, was done by a worm. A worm can also be used for installing a backdoor, allowing the target machine to be included in a network of machines hijacked for a particular purpose, known as a botnet.

7.9.4 The Trojan payload

Trojans may avoid disrupting the host system in order to remain undetected while, for example, stealing corporate information, or credential stealing. Trojans can also be used in ransomware attacks, when it is quite obvious to the user that they have been infected. There are Trojans that open a back door allowing remote control of the machine. These are known as RATs – **Remote Access Trojans**. Once installed on a PC, RATs give hackers complete control: they can record keystrokes and websites visited, copy/delete files, etc.

A RAT normally has bot software incorporated; this allows a machine to be conscripted into a botnet, a network of many machines, potentially millions, which a hacker can control with one command. Trojans may also do such things as collect email addresses from contact lists for spam – if on an infected mobile phone they may send SMS messages and cost the user money – or they can download and install other programs including a rootkit.

Activity 7.2

Answer the following questions:

- (i) Which one of the following is **not** a type of malware?
 1. Logic bomb
 2. Bootkit
 3. Ransomware
 4. PUPs
 - (ii) The difference between a worm and a virus is:
 1. Worms are autonomous and do not need to attach themselves to a host
 2. Viruses can make copies of themselves, and worms cannot
 3. A virus pretends to be a legitimate program, while a worm hides itself from the user
 - (iii) Which of the following is a type of virus?
 1. Igneous
 2. Metamorphic
 3. Sedimentary
 4. All of the above
 - (iv) Which of the following is a vector for malware?
 1. Email
 2. Removable media
 3. Online advertisements
 4. All of the above
 - (v) Answer true or false to the following.
 1. Internet of Things devices are targeted by hackers for botnets because they are often always on and have low security standards.
 2. Viruses can use their target's address book to send spam.
 3. Malvertising is only found on compromised websites.
 4. Your fridge may be planning to rob a bank.
-

7.10 Computer security and defence against malware

Computer security is a flourishing industry, with a wide choice of anti-virus software available. In this section we describe some of the ways anti-virus and anti-malware software operates, and other ways to protect against infection by malware.

7.10.1 Symptoms of malware infection

Computer behaviour and performance

- Your computer takes longer than normal to start up. This can be because it is applying updates, but it can also be a sign of infection.
- When your computer is booting up you see odd messages or windows, especially those relating to lost access to drives.
- Your machine has performance issues, it is slow to load files and programs, and slow to display web pages. Slow performance can be because your hard drive is nearly full, or is badly fragmented, or it may be that you have a lot of apps open using a lot of processing power and memory. Windows users can press together CTRL, ALT and DELETE to open the task manager and see what is using processing power and memory. If none of these things apply it may be that malware is using your processor and memory.
- Your computer crashes. All machines crash eventually, but this can be caused by malware. It is a good idea after a crash and a successful reboot to check that anti-virus software is up-to-date and to run a scan.
- The computer restarts itself. Restarting may be a sign that malware has been installed.
- The computer often restarts without notice.
- Odd behaviour. Unusual or unexpected sounds or problems with the display such as flickering, can be a sign of malware infection. If your computer opens files you did not ask it to, does not respond to clicks, scrolls or otherwise takes actions you have not asked it to, it could be infected.

Files

- Some viruses will increase the size of infected files. This was once a certain sign of infection, but viruses are sophisticated enough now to hide such changes. Malware may also change the access date and time of infected files, but this can also be hidden.
- If the amount of used space on your hard drive has increased/decreased unexpectedly, this could be a sign that malware is using your storage space. Files may be moved, deleted or encrypted by malware. In addition, malware may add files and existing files can be overwritten by new files. It is a good idea to keep track of the size of your hard drive, although good anti-virus software will do this.

Issues with software

- Certain applications will not start. Some malware targets certain applications.
- Apps opening or closing without your intervention.
- Apps go missing or become corrupted and no longer work as they should.
- Icons appear on your desktop for apps that you did not install.

Internet and email

- If your internet connection is in use even when you are not using a browser or search engine, it can be that malware is sending information over the internet. If you have a metered network connection, running out of data unexpectedly may be a sign of infection, or of hacking. If you do not have a metered connection, Windows 8 and Windows 10 allow you to set your connection as metered in **network and internet settings**, which will then track your usage.
- Contacts and friends receive email and social media messages from you that you did not send. You might also notice messages in your sent mail that you did not send. Alternatively, contacts do not receive email messages you did send. This could mean a virus is attempting to spread itself using your accounts, but it could also mean that your accounts have been hacked. As well as a virus scan, it would be a good idea to log out of all social media accounts from all of your devices, log out of your email, and change your password. If you have not already, moving to two-factor authentication improves security.
- Your web browser's home page has changed, and you did not change it.
- Your web browser takes you to websites that you did not ask it to, or your web searches bring up unexpected sites.
- You receive a message that your IP address has been blacklisted. This probably means that your machine has been identified as being part of a botnet.

Hardware issues

- If your hard disk spins excessively it can be a sign of a virus, although it can also be a sign of incipient hard disk failure.
- If your printer starts printing documents that you have not sent it is possible that your printer is infected, although more likely it is your computer. Your printer is more likely to be hacked than infected. Bruce Schneier (Schneier 2018, *Introduction: Everything Is Becoming a Computer*) reports that in 2017 someone hacked 150,000 insecure printers 'and had them repeatedly print ASCII art and taunting messages.' Printing unasked for documents can also be your printer's way of telling you that its firmware has been updated.
- Documents do not print correctly.
- Speakers play music or sounds unexpectedly.
- The keyboard is remapped.

Pop-ups and dialogue boxes

- You see pop-ups with advertisements when you are not using a web browser. This probably means your machine has been infected with spyware or malicious adware. Do not click on such ads, even to close them. Adware will also attempt to use your web searches to direct you to advertising sites.
- You see unexpected pop-ups on your machine when you are not web browsing. These may warn you of virus infections. If the pop-up is not from your own anti-virus software do not click on it, even to close it.
- Your computer opens dialogue boxes on your screen that have text that makes no sense.

Warnings from the OS

- Your computer warns you that something is wrong. You can search for error messages you receive from your operating system with your web browser to see if they can be warning signs of an infection. These messages could include:
 - The OS warns you about missing system or application files.
 - The OS warns you that you are running out of disk space.
 - The OS warns you that you do not have access to a drive or a partition.
 - The OS tells you that certain files or programs will not open.
 - A program is attempting to access the internet without your consent.

Anti-virus issues

- Your anti-virus software tells you that it is disabled, or that the automatic update feature is not working. You may get a warning that your firewall is off. Some malware tries to disable your protection, so these can be signs of infection.
- You are unable to run your anti-virus software, or unable to install anti-virus software.
- Your anti-virus software reports a virus.
- MS-Word macro protection warns that a file contains macros.

Note: Many of these can have legitimate causes or result from software errors (bugs). False positives can also occur; for instance, if you are installing a package that modifies system files, your anti-virus program may report this as an intrusion.

7.10.2 Anti-virus and malware detecting software

Anti-virus scanners work by scanning hard drives for executable files to find the signature of malware. The signature of a virus or other malware is a unique sequence of bits used to identify it. Signatures are stored locally in a database. As new infections are discovered they are added to the vendor's database, hence anti-virus software regularly connects to the vendor via the internet and updates the database it is relying on. Signature detection is often used by anti-virus scanners in combination with heuristic detection, looking for code that is similar to known viruses, in order to detect variant versions of known infections.

Malware can also be detected by **behaviour-based detection** – comparing the actions of software to a list of known suspicious activities. Behaviour-based security software operates by classifying potential actions into acceptable and unacceptable, and in this way can detect and give protection from zero day threats, targeted, fileless and polymorphic malware and ransomware. Such anti-malware software can also include rollback of malware behaviour and **Automatic Exploit Prevention** (AEP), to detect unknown malware that takes advantage of known software vulnerabilities.

Behaviour based detection can also be applied proactively, by running software in a **sandbox**. In this technique software is run in a protected environment, and its actions monitored, recorded and analysed. This technique has been in use for some time, and malware authors try to find ways to circumvent it, including detecting when the software is being run in a sandbox and only allowing certain benign behaviours then.

Data mining together with machine learning is being utilised to analyse large data sets to find patterns, and use these patterns to detect viruses and other

malware, such as polymorphic malware. This is quite a recent trend, and the subject of ongoing academic research.

7.10.3 How to protect against unauthorised access, credential stealing and infection

Individuals and companies can take a number of common-sense measures to reduce their exposure to malicious code, whether or not they decide to install AV software.

Low tech

It is very easy to hack into microphones and web cams. One solution is to put tape over your laptop's web cam when you are not using it. *The Guardian* reports that former FBI Director James Comey does this. Hackers can and do access web cams and take photographs without your knowledge; these photos may even be posted online. The same report notes that Edward Snowden has reported on NSA programs to capture images from web cams, see: www.theguardian.com/commentisfree/2018/apr/06/phone-camera-microphone-spying Note that tape over the web cam's lens will not disable the audio, which might still be turned on without your knowledge, unless you find out how to disable it.

Credential protection

Perhaps the most important protection is to be very careful with log-in details, both in terms of keeping them secure, and choosing passwords and security questions that are hard to guess. For anything critical, change passwords regularly and use sites such as: <https://haveibeenpwned.com/> that allow you to check if your passwords have been compromised. It goes without saying to never give your password to another person, because hackers regularly report that asking people for their password works. Bruce Schneier (and Edward Snowden) recommends moving to two-factor authentication (2FA), as it improves security, although is not foolproof. One acknowledged problem with 2FA is that when it is implemented it is often with the use of a text message to a mobile phone, and mobile phones can be vulnerable to SIM-swapping attacks. In a SIM-swapping attack a criminal fraudulently takes control of a phone number, and then uses it to access the victim's accounts. In fact cybersecurity investigator Brian Krebs is very much against the spreading use of mobile phones in 2FA, see: <https://krebsonsecurity.com/2019/03/why-phone-numbers-stink-as-identity-proof/>

One very simple protection is to search a list of the most common passwords, and change any of your passwords that are on the list. Lists of common passwords are used in credential spraying attacks (see Section 7.7.4). You can find lists of the 10,000 most popular English passwords on the web, including at Wikipedia: https://en.wikipedia.org/wiki/Wikipedia:10,000_most_common_passwords You should also be careful about security questions, and not reuse them so that if one of your accounts is compromised, the same information cannot be used to hack into your other accounts.

You might also consider using hardware-based security such as Yubikey, which supports logging in with one-time passwords. See: www.yubico.com/

Some authorities suggest using a password manager, one that stores unique and strong passwords for all of your services. This is because, with so many passwords to remember, there is a temptation to reuse passwords, or to write them down. Never reuse the same password as once a hacker has one set of log-in details, he or she will try that password at multiple sites (when this process is automated it is known as credential stuffing).

Email

Malware tries to spread itself through email, so do not click on links in emails including from your contacts, unless you are certain that the message is genuine. Remember that the **from** field can be spoofed. Never click a link in an email, or download an attachment, if the sender is not someone you know or expect a contact from.

If you get a link in an email that you think might be genuine, it can be a good idea to type the address into your browser manually, as this can help you to identify spoofed URLs, links that look genuine at first sight, but differ in some way from the URL of the genuine site that you think it is.

Also exercise caution when opening email attachments, although most providers will automatically scan attachments, this is no guarantee against all malware. Use an email client that sends suspected spam into a separate folder, as most do. Never respond to spam, as spammers will know your address is live, and you will receive more.

Patch your operating system

Keep your operating system up-to-date with all patches. Unpatched OS's leave users vulnerable to opportunistic attacks as hackers and criminals will try to find unpatched systems in order to exploit known weaknesses.

Do not use an out of date operating system

Many of the victims of the Wannacry attack in May 2017 were using older versions of the Windows operating system that Microsoft no longer supported.

Keep your attack surface small

Do not keep apps and programs that you no longer use on your devices, uninstall them. The more apps that you have, the greater your attack surface. You could also consider turning off unwanted functions in the apps that you do use.

Safe browsing

- Keep browsers and plug-ins/extensions up-to-date, as browsers and plug-ins/extensions are also vectors for attack.
- The more plug-ins/extensions you have, the more likely you are to get infected, or hacked, so only keep the ones you really need and use.
- Check the security settings in your web browser.
- Do not click on pop-up ads.
- Use an ad blocker as this can protect you from drive-by attacks, which often use online ads.
- Be very careful when asked to download codecs, as codecs are a notorious malware vector.

Anti-virus software

- Install anti-virus software and keep it up to date. Look for software that also protects against malware including spyware and ransomware. Anti-virus programs now can do things such as protect your web cam from spying (only allows trusted apps to use it), wall off folders from ransomware, warn you about visiting suspect websites or against downloading suspect apps. Choose software that has the capabilities that you need.

- Always have your firewall on.
- Activate macro virus protection in Word and Excel.

Protect your data

- Consider encryption for data, web browsing and mobile devices.
 - Hard-drive encryption can slow your operating system, as files have to be decrypted to be read and used, then encrypted again. One solution is to partition the hard drive, and only encrypt files in the partition. You should place in the partition any data (including photographs) that you would not want to share with the world. Encrypting any sensitive data can help to protect you against ransomware attacks that steal and threaten to publish sensitive data. Remember that with encryption there is always the risk of losing the access details and being permanently locked out of your own files.
 - The Electronic Frontier Foundation (EFF) www.eff.org/, is a US non-profit advocacy group for online privacy and free speech. The Tor Project (www.torproject.org/) is another US non-profit with close links with the EFF. The Tor Project was started by computer scientists to protect online privacy and freedom by developing and promoting technical tools to promote privacy and thwart surveillance online. In a joint project EFF and the Tor Project have developed a web-browser extension that encrypts web traffic and can protect against your account details being intercepted and session hijacking (see Section 7.6.4). The extension, *HTTPS Everywhere*, forces web sessions to use the more secure HTTPS protocol and is available for many popular web browsers (for example, Google Chrome, Firefox for Android) although not any produced by Microsoft. You may choose to use a web browser that markets itself as secure, but note that it is important to distinguish between privacy and security. The Tor web browser, for example, is great for privacy, but does not protect the user from malware.
 - Use an app that encrypts cell/mobile phone calls and texts in order to protect your information. Signal (<https://signal.org/>) is an app that encrypts voice and text messages, although its focus is on privacy rather than protection from malicious attacks.
- Keeping back-ups and keeping them up to date, and not plugged into your network, is one way to protect against ransomware attacks. One option is to use software that will regularly back up data, perhaps to a (removable) external hard drive. Backing up data can be slow and tedious, hence using software that automates the process, and speeds it up by only saving files that have changed since the last backup, encourages good habits.

Malicious websites

- **Avoid** Warez sites; that is, sites that provide pirate software because:
 - Warez sites cannot be trusted, and are likely to attempt to infect you
 - the software you download may be compromised
 - pirate software cannot be patched, leaving you vulnerable.
- **Avoid** downloading files if not from trusted sources.

Removable media

- Make sure that any removable media are automatically scanned by your anti-virus software. Never auto-play removable media as malware can then be automatically installed.

- Be suspicious of programs and files on removable media if not from trusted sources.
- You can install and use a sandbox program in order to run files safely. Be aware that some malware can detect sandboxes and alter their behaviour accordingly.

Activity 7.3

1. What is a software patch?
 2. Give one reason why patching may not be a good approach to software development in terms of computer security.
 3. Give an example of a notorious malware attack where a software patch played an important part.
-

7.11 Hardware vulnerabilities

We have not been used to thinking of hardware as a source of vulnerability, at least until Edward Snowden revealed that the NSA routinely intercepts and inserts backdoors into US routers for export, see: www.theguardian.com/books/2014/may/12/glenn-greenwald-nsa-tampers-us-internet-routers-snowden Nevertheless, hardware vulnerabilities are an increasing subject of both online and traditional news media reportage.

In 2018, *ArsTechnica* reported on issues with chips that arose because of the speculative execution used in pipelining. One was an attack on Intel chips, known as **Meltdown**, and another on AMD and ARM chips known as **Spectre**, see: <https://arstechnica.com/gadgets/2018/01/meltdown-and-spectre-every-modern-processor-has-unfixable-security-flaws/> After discovery of **Meltdown**, Intel funded research, and included Meltdown attacks in their 'bug bounty' program, where researchers are paid for discovering and notifying Intel of vulnerabilities. See: <https://searchsecurity.techtarget.com/news/252435175/Intel-bug-bounty-programs-widened-after-Meltdown-and-Spectre>

In addition, in 2018 there were reports that the **Supermicro** motherboard was compromised. Bloomberg, the originator of the stories, claimed that a tiny chip was being added to the motherboards, before they were shipped to the USA from China. The chip was allegedly later used to steal data. *ArsTechnica* reported on these claims and found them dubious, while noting that issues with the motherboard's firmware had been discovered in 2013 and 2014, and could have been exploited to do exactly those things that Bloomberg claimed were being done. See: <https://arstechnica.com/information-technology/2018/10/supermicro-boards-were-so-bug-ridden-why-would-hackers-ever-need-implants/>

One of the biggest tech news stories in 2019 has been a trade war between the Chinese telecommunications company Huawei, and the US government, leading to Google refusing to trade with Huawei from May 2019. This has meant Google rescinding Huawei's Android licence and no longer providing other proprietary services. The US government claims to have national security concerns about Huawei's equipment because of its links to the Chinese government, while the company itself claims to be a private company, entirely independent of the Chinese government. On 10 June 2019 the BBC reported that:

The US has encouraged allies to block Huawei – the world's largest maker of telecoms equipment – from their 5G networks, saying the Chinese government could use its products for surveillance.

See: www.bbc.co.uk/news/business-48588661

7.11.1 Hardware with software vulnerabilities

There has been online speculation about malware on SIM cards for mobile phones, with suggestions that this is something that state actors are either attempting, or actually doing. In September 2019 *The Hacker News* reported on an attack vector in software that is embedded in some SIM cards, called *Simjacker*, and claimed:

A specific private company that works with governments is actively exploiting the SimJacker vulnerability from at least the last two years to conduct targeted surveillance on mobile phone users across several countries.

See: <https://thehackernews.com/2019/09/simjacker-mobile-hacking.html> for more information

7.12 Resources for further and up-to-date information

The following websites provide useful information:

- **Bruce Schneier's** blog *Schneier on Security*. Schneier is an internationally renowned cyber security expert, and a very clear thinker about what kind of future the current technological, social and political landscape is leading us to.
www.schneier.com/
- **Brian Krebs'** blog – an investigative journalist in computer security who has come under cyber attack because of the effect his stories have had on the activities of cyber criminals. For example, one of his reports led to a huge drop in the volume of spam sent worldwide.
<https://krebsonsecurity.com/>
- **The Hacker News** is a good place to find in-depth coverage of the latest cybersecurity threats.
<https://thehackernews.com/>
- The blog of **Marcus Hutchins**, the man who stopped the 2017 WannaCry attack.
www.malwaretech.com/
- **Symantec** anti-virus centre
<http://www.symantec.com/avcenter>
- **McAfee's** anti-virus blog
<https://securingtomorrow.mcafee.com/>
- **Sophos** run a threat newsroom
<https://nakedsecurity.sophos.com/>
- **Kaspersky's** threat news
<https://securelist.com/>
- Interesting site with **up-to-the minute malware news**
www.bleepingcomputer.com/
- A good site if you come under **ransomware attack**
www.nomoreransom.org/crypto-sheriff.php?lang=en
- A Google project – a map of ongoing **DDoS attacks world wide**
<http://www.digitalattackmap.com/>

- **CSSR:** Centre for Computing and Social Responsibility (CSSR). Focusses on the ethical and social implications of ICT
<http://www.dmu.ac.uk/research/research-faculties-and-institutes/technology/centre-for-computing-and-social-responsibility/ccsr-home.aspx>
- **CERT** centre at Carnegie-Mellon University: highly informative on network security
www.sei.cmu.edu/about/divisions/cert/index.cfm
- **Risks Digest:** online magazine published by the Association for Computing Machinery (ACM)
<http://catless.ncl.ac.uk/Risks/>
- **Microsoft's** UK security centre
www.microsoft.com/en-gb/security/default.aspx
- The UK's **National Cyber Security Centre**, with the aim of making the UK the safest place in the world for cyber business. The NCSC is part of GCHQ
www.ncsc.gov.uk/
- **The European Institute for Computer Anti-Virus Research (EICAR)**, representing organisations such as universities, experts, governmental bodies and vendors
<http://www.eicar.org/>
- **The European Union Agency for Network and Information Security (ENISA)** describes itself as 'a centre of expertise for cyber security in Europe'
www.enisa.europa.eu/
- **Common Vulnerabilities and Exposures (CVE).** A list of publicly known cybersecurity vulnerabilities. CVE IDs (or CVEs, or CVE numbers) provide an agreed reference point for known problems, so that cybersecurity specialists around the world can be sure that they are discussing the same problem. Participation in CVE numbering is worldwide, voluntary and growing. CVE feeds into the U.S. National Vulnerability Database (NVD), and is sponsored by the office of Cybersecurity and Communications at the U.S. Department of Homeland Security
<https://cve.mitre.org/>
- Part of the USA's Department of Homeland Security, **the National Cybersecurity and Communications Integration Center's (NCCIC)** is 'the Nation's flagship cyber defense, incident response, and operational integration center.' The NCCIC operates a '24/7 situational awareness, analysis, and incident response center.'
www.dhs.gov/national-cybersecurity-and-communications-integration-center
- The **National Security Agency** of the USA
www.nsa.gov/

7.13 Overview of the chapter

In this chapter we considered what hacking is, why people hack and how. We looked at why the internet has so many security issues and what is being done, and what can be done, to mitigate them. We identified types of malware, how malware infects computers and how to defend against this. We looked in some detail at symptoms of a malware infection, and considered the operation and continued development of both malware and anti-malware software.

7.14 Reminder of learning outcomes

Having completed this chapter, and the Essential reading and activities, you should be able to:

- explain the differences between viruses, worms and Trojan horses
- describe some of the software vulnerabilities they exploit and the malicious techniques they use
- advise on measures to avoid exposure to malicious code
- explain in general terms how anti-virus software functions.

7.15 Test your knowledge and understanding

7.15.1 Sample examination questions

- (a) (i) Under what circumstances should you give your password to another person? [2 marks]
1. When they ask for it
 2. When your ISP or network administrator asks for it
 3. You should never give your password to another person
 4. None of the above
- (ii) Which one of the following may be an indication that your computer has been infected with malware? [2 marks]
1. Your computer restarts itself
 2. You find an icon on your desktop for an app you did not install
 3. Your hard disk is spinning excessively
 4. All of the above
- (iii) A **bug bounty** is: [2 marks]
1. An eradication programme for invasive insect species
 2. A hacker's term for the gains to be made from software bugs
 3. A reward offered by hardware and software vendors to members of the public who notify them of security vulnerabilities in their products
 4. None of the above
- (b) (i) What is a signature in terms of virus protection? [3 marks]
- (ii) Explain why metamorphic and polymorphic viruses present a particular challenge to anti-virus software. [6 marks]
- (c) (i) Give three reasons why the Internet of Things (IoT) could make computer and network security more difficult. [6 marks]
- (ii) What solution would you propose to the security challenges presented by the IoT? [4 marks]

Notes

Appendix 1: The binary number system

Powers of 2

Power of 2	Result	Smallest number of bits in unsigned binary number	Biggest unsigned binary number with that number of bits	Decimal equivalent of biggest unsigned binary number
2^0	1	1	1	$2^1 - 1 = 1$
2^1	2	2	11	$2^2 - 1 = 3$
2^2	4	3	111	$2^3 - 1 = 7$
2^3	8	4	1111	$2^4 - 1 = 15$
2^4	16	5	11111	$2^5 - 1 = 31$
2^5	32	6	111111	$2^6 - 1 = 63$
2^6	64	7	1111111	$2^7 - 1 = 127$
2^7	128	8	11111111	$2^8 - 1 = 255$
2^8	256	9	111111111	$2^9 - 1 = 511$
2^9	512	10	1111111111	$2^{10} - 1 = 1,023$
2^{10}	1,024	11	11111111111	$2^{11} - 1 = 2,047$
2^{11}	2,048	12	111111111111	$2^{12} - 1 = 4,095$
2^{12}	4,096	13	1111111111111	$2^{13} - 1 = 8,191$
2^{13}	8,192	14	11111111111111	$2^{14} - 1 = 16,383$
2^{14}	16,384	15	111111111111111	$2^{15} - 1 = 32,767$
2^{15}	32,768	16	1111111111111111	$2^{16} - 1 = 65,535$
2^{16}	65,536	17	11111111111111111	$2^{17} - 1 = 131,071$
2^{17}	131,072	18	111111111111111111	$2^{18} - 1 = 262,143$
2^{18}	262,144	19	1111111111111111111	$2^{19} - 1 = 524,287$
2^{19}	524,288	20	11111111111111111111	$2^{20} - 1 = 1,048,575$
2^{20}	1,048,576	21	111111111111111111111	$2^{21} - 1 = 2,097,151$
2^{21}	2,097,152	22	1111111111111111111111	$2^{22} - 1 = 4,194,303$
2^{22}	4,194,304	23	11111111111111111111111	$2^{23} - 1 = 8,388,607$
2^{23}	8,388,608	24	111111111111111111111111	$2^{24} - 1 = 16,777,215$
2^{24}	16,777,216	25	1111111111111111111111111	$2^{25} - 1 = 33,554,431$
2^{25}	33,554,432	26	11111111111111111111111111	$2^{26} - 1 = 67,108,863$
2^{26}	67,108,864	27	111111111111111111111111111	$2^{27} - 1 = 134,217,727$
2^{27}	134,217,728	28	1111111111111111111111111111	$2^{28} - 1 = 268,435,455$
2^{28}	268,435,456	29	11111111111111111111111111111	$2^{29} - 1 = 536,870,911$
2^{29}	536,870,912	30	111111111111111111111111111111	$2^{30} - 1 = 1,073,741,823$
2^{30}	1,073,741,824	31	1111111111111111111111111111111	$2^{31} - 1 = 2,147,483,647$

Recall the format of a binary number = $b\ b\ b\ \dots\ b\ b\ b$, where b can take the values 0 and 1 only. This means that the maximum value for a 7-bit field is 111 1111 which is $1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$. Note that adding one to this number would give the 8-bit number 1000 0000, or 2^7 exactly. Hence the maximum size for a 7-bit unsigned binary number is $2^7 - 1$.

Note that this means that there are $2^7 - 1$ unique 7-bit binary numbers, counting from 1. If we include the number zero, then we would have 2^7 unique 7-bit numbers exactly. Hence in the general case, the biggest unsigned binary number with n bits is $2^n - 1$. If we include zero as a potential number then this gives 2^n unique n -bit binary numbers.

For example, with 4 bits we have 15 ($2^4 - 1$) unique numbers starting from 1, but 16 (2^4) if we include zero.

Unsigned binary 4-bit number	Decimal equivalent
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Appendix 2: Answers to Sample examination questions

Chapter 2

(a)

- (i) 1. Data is transmitted in chunks with dynamic routing; once a router has forwarded the data packet its memory is cleared and it is ready for other tasks.
- (ii) 4. All of the above.
- (iii) 2. Internet Corporation for Assigned Names and Numbers

(b)

- (i) Client and server processes operate on machines that can communicate through a network.

The client:

- requests a connection to the server when it needs the service that the server provides.
- may disconnect when they have what they need.

The server:

- listens for and accepts connection requests.
- listens for requests from connected clients.
- fulfils requests when it hears them and sends a response to the client.

- (ii) Port numbers were developed as a way of connecting to particular services on remote machines. Port numbers allow clients to signal the service they wish to access on a remote machine by appending a port number to the IP address, and they allow the receiving machine to identify the requested service.

(c)

- (i) Advantages of layering in network models:
 - Layering provides a structure and language to help designers, developers and administrators define and discuss networking protocols, tasks and functions.
 - Layering simplifies protocol design and development by strictly defining and limiting the interfaces between layers, which delineates and restricts the information that protocols need.
 - Restricting interfaces and keeping design and implementation separate, means that a technological change in one layer does not mean changes must be implemented in other layers, simplifying design and development of hardware.
 - Since the networking model and protocols are independent of implementation, hardware and software from different suppliers and vendors can work together, promoting competition and preventing any one company from gaining a monopoly.

(ii) The levels are:

- Application layer
- Transport layer
- Internet layer
- Network access layer.

In the TCP/ IP model, layering is implemented by dividing packets into **header** and **data sections**. When an application wishes to transmit data to an application on another system, the data is divided into chunks and passed from the Application layer to the Transport, Internet and Network Access layers before being sent out into the internet. At each layer, the sending stack attaches a header containing information for software at the same stack layer in the receiving system. Hence data can be viewed as travelling down through the layers of the model, with a new header attached at each layer, such that the packet passed from the previous layer becomes the payload of the packet in the next layer. At the bottom layer the data packet is sent across the internet to the receiver, where the headers are removed in reverse order to the order they were applied. The Network Access layer removes the header applied by that layer in the sending machine, before passing the packet up to the Internet layer. This continues until the packet reaches the Application layer.

Chapter 3

(a)

- (i) 2. A segment
- (ii) 3. A machine connected to the internet that forwards data packets
- (iii) 1. The Internet Control and Messaging Protocol

(b)

- (i) 11111111 11111111 11111111 11111000 in 32-bit binary.
This corresponds to 255.255.255.248 in dotted decimal notation.
- (ii) The default address would be the last entry in the routing table.

(c)

- (i) Each packet in a data stream is treated as an independent unit, with no connection to previous or subsequent packets in the same stream. Routers accept packets for forwarding/delivery with no memory of the packets they have recently dealt with. Hence packets in the same stream may follow different paths to the same destination; they may be delivered out of order; some may not be delivered at all.
- (ii) Possible answers are:
 - Routers recalculate the **checksum**, and compare it to the value in the IP datagram header. If the values do not match, the data packet is assumed to have been corrupted, and is discarded.
 - Each datagram header has a **time-to-live** number that is decremented by one by each router it passes through. When the time-to-live reaches zero the data packet is discarded.

- If the **version** number indicates a version of the Internet Protocol that the router does not support, the router will discard the datagram.
- If the source address is designated as private under the IPv4 address scheme.

Chapter 4

(a)

- (i) 1. Network Access → Internet → Transport → Application
- (ii) 2. TCP
- (iii) 1. The well-known ports are numbered 0–1023, and are also known as the 'System Ports'.

(b)

- (i) The **checksum** is optional in the current version of UDP to allow for low transmission overheads in reliable networks.
- (ii) UDP may be used to stream audio or video, or for other purposes for which no reply is needed. Since the purpose of the source port is to allow the receiving system to reply, its use is optional, as a reply may not be necessary; it may even be unwanted.

The purpose of TCP is to establish a reliable communication link between the sender and receiver; this requires two-way communication. For example, the link is established with a 'three-way handshake' between sender and receiver. Hence it would not be possible to establish a link if the receiver does not know the sender's source port, the three-way handshake would fail and the Transmission Control Protocol could not work.

(c)

- (i) Data packets are expected to be received in order, and only data received in order will be acknowledged. An acknowledgement confirms receipt of all unacknowledged data received with a smaller sequence number. Since an ACK implicitly acknowledges all segments with a smaller sequence number, an ACK does not have to be sent each time a correct segment is received. Under cumulative ACKs a segment can only be acknowledged if all previous segments have been received.
- (ii) Assume a situation where the receiver has received packet $x - 1$, followed by packets $x + 1$, $x + 2$, followed by many other in order packets. The receiver cannot signal to the sender that while it has not received packet x , it has received packet $x + 1$, followed by a great many other in order packets. If the sender follows the accepted standard and retransmits only the first unacknowledged segment, it must wait for the acknowledgement before it can decide what and how much to resend. Thus, retransmission reverts to a send-and-wait paradigm.

Alternatively, the sender can send again all packets starting with packet x , even though many of them have already been successfully received. This is a potentially large overhead.

Chapter 5

(a)

- (i) 4. All of the above
- (ii) 2. PHP
- (iii) 1. A protocol to map domain names to numerical addresses

(b)

- (i) That HTTP is a **stateless protocol** means there is no need for the client or server to keep track of information about each other; each HTTP request is intended to be independent of earlier requests from the same client.
- (ii) **Fault tolerance** means that a browser will display as much information as it can, by interpreting as much of the HTML code as it can, and ignoring the rest. The advantage is that it makes HTML forward compatible, meaning that newer HTML code can be displayed by browsers using an older version of HTML.

(c)

- (i) Some browsers can recognise these sections and treat them differently; for example, by allowing the viewer to scroll through the body section of large tables, while keeping the header and footer sections constant at the top and bottom of the page. Alternatively, when a table is viewed in successive pages due to its large number of entries, the browser can insert the header and footer sections at the top and bottom of each page.

(ii)



Year	2016	2017	2018
widgerts	11	72	917
wosnames	5	7,000	964
TOTAL	16	7,072	1,881

Chapter 6

(a)

- (i) 4. All of the above.
- (ii) 1. TRIPS is an international convention about intellectual property law.
- (iii) 3. Under the GDPR a data subject's genetic data is considered to be special category data; processing of such data is subject to additional safeguards to that given to personal data.

(b)

- (i) (A) Patents
(B) Copyright
- (ii) Individuals and companies within the EU have a 'database right', a right applying exclusively to databases, that introduces a new form of intellectual property: the work of compiling the database.

In introducing the database right, the EU was concerned that the contents of a database may be copied and rearranged in such a way that the content is identical, but technically no copyright or other laws have been broken. However, the originator of the database may have lost the financial and/or professional investment made in setting up the database and sourcing its contents. Hence database originators were given a *sui generis* (without precedent) right to prevent, or give remedy against, electronic copying. Provided that the database owner can show that their investment has been substantial, they have the database right. The investment can be financial or in terms of the work undertaken. This right only applies to databases created inside the EU, and is independent of the copyright status of the material in the database.

(c) Answers will vary; a good answer might cover the following points:

Cybercrime could be defined as falling into two broad categories:

1. Existing crimes committed with new technology; this includes issues of copyright, defamation, obscenity, fraud, blackmail and theft.
2. Crimes that did not exist before networked computing, such as hacking.

An alternative distinction is between crimes committed using a computer (the computer as tool) and crimes committed on a computer (the computer as target).

More specifically, one authority gives the following four categories of cybercrime:

1. Crimes where the computer is the target of the crime.
2. Crimes that relate to illegal or illicit computer/web content.
3. Crimes against a person committed using a computer.
4. Crimes against property committed using a computer.

You can think of cybercrime as relating only to those crimes that would not be possible without networked computers. For example, defrauding people is a crime, and it does not matter whether the person is defrauded via electronic communication, or face-to-face, or a mixture of both. Defrauding people via online contact could be defined as computer-aided crime, but you may not think of it as cybercrime. This author believes that the definition should only apply to crimes such as unauthorised access to a computer, deliberately damaging computers and the data they contain, DDoS attacks that seek to restrict the access of legitimate users, and so on. Hence, in the above four-point definition, this author would include only the first point as cybercrime.

Chapter 7

(a)

- (i) 3. You should never give your password to another person
- (ii) 4. All of the above
- (iii) 3. A reward offered by hardware and software vendors to members of the public who notify them of security vulnerabilities in their products

(b)

- (i) A signature of a virus or other malware is a unique sequence of bits found in the virus and used to identify it.
- (ii) Polymorphic viruses release their payload and then encrypt themselves before attaching themselves to another application. Each encryption is unique in order to prevent anti-virus software from using signature-based detection. Metamorphic viruses are even harder to detect than polymorphic viruses as they can rewrite their own code as a way of frustrating signature-based detection.

(c)

- (i) Below are six reasons why the Internet of Things may be problematic in terms of computer security (only three are needed); you may be able to think of others.
 1. The addition of billions of extra devices to the internet increases its attack surface.
 2. The addition of billions of extra devices to the internet increases its complexity, and this can mean unexpected interconnections and outcomes that an attacker can exploit.
 3. Embedded systems use general purpose computer chips, meaning that they have the capacity to do anything that any general purpose computer can do, which means that hackers can use them for anything they might use another hacked system for; such as installing a bot for use in a botnet.
 4. Devices in the Internet of Things are made by manufacturers of cars and fridges, for example, who may not have a good appreciation of computer security issues. Hence these devices can be very easy to hack; for example, they may have default passwords that the user cannot change.
 5. Because IoT devices are made by manufacturers who lack experience in computer security, their systems may be difficult or impossible to patch when software vulnerabilities are discovered.
 6. Internet of Things devices may continue to be used for many years. Hence devices may use operating systems or other software that is no longer supported. Unsupported software is no longer patched (if it ever was) leaving it vulnerable to newly discovered security issues.
- (ii) Issues raised by computer security experts regarding the IoT need to be addressed by manufacturers, who need to understand and fall into line with computer industry best practice. Manufacturers may be reluctant to take on additional development and production expenses that their rivals are ignoring, hence government regulation may be needed to protect the consumer by mandating a minimum level of security for all devices.

Appendix 3: Answers to activities

Chapter 2

Activity 2.1

1. Ethernet technology is used for LANs (local area networks).
2. The Network Access layer.
3. The technology has been very successful, because it works well, is relatively cheap and is easy to install.
4. Collision rates, when data packets are lost due to collisions, are high when the network is heavily used. At light or moderate use collision rates are manageable, but collision rates rise in a non-linear way as network traffic increases. On modern Ethernet networks additional technology is used to manage and reduce collisions.
5. Ethernet technology was developed by the Xerox corporation, at its Palo Alto Research Centre (Xerox PARC), in the 1970s.

Activity 2.2

1. The RFC 793 defines the Transmission Control Protocol.
2. The protocol was developed under the aegis of the United States Department of Defence through the Defence Advanced Research Projects Agency.
3. RFC 2468 is not a technical specification, but an obituary for Jon Postel, the editor of RFCs, an important contributor to the design of TCP/IP protocols, and the person responsible for assigning IP addresses in the early internet. Comer notes: 'From the time the Internet began until the fall of 1998, a single individual, the late Jon Postel, ran the IANA and assigned addresses. In late 1998, after Jon's untimely death, a new organisation was created to handle address assignment. (Comer 2014, Section 5.23 *Internet Address Assignment and Delegation Of Authority*).
4. RFC 349 proposed that particular port numbers be assigned to standard protocols.
5. RFC 349 has been updated many times, as the suggestion was taken up, and eventually became the well-known port numbers. RFC 349, published in 1972 was made obsolete by RFC 433, which in turn was made obsolete by RFC 503, and this process continued through more than 20 successive RFCs, culminating in RFC 1700 in 1994. RFC 3232, published in 2002, made RFC 1700 obsolete, and noted that 'RFC 1700 is Replaced by an On-line Database', see: <https://tools.ietf.org/html/rfc3232>
6. RFC 1812 is concerned with how the TCP/IP protocol suite is implemented by routers. The RFC 'enumerates standard protocols that a router connected to the Internet must use'. The document references other RFCs describing protocols it is concerned with and 'corrects errors in the referenced documents and adds additional discussion and guidance for an implementor.'
7. The DISCUSSION and IMPLEMENTATION sections are not part of the standard because they justify, clarify or explain requirements and give advice that those implementing the protocols may want to follow.

Since implementation is intended to be separate from requirements throughout the TCP/IP protocol suite, implementation choices are entirely for implementers, so any suggestions can only be such and cannot be incorporated into the protocols. In other words RFCs abstract the details to describe the processes and tasks that each protocol must carry out in order to provide the service it is promising. Implementation is left to vendors, designers and network administrators.

Chapter 3

Activity 3.1

The first network entry in the table: 162.168.0.0/27

Performing a bitwise AND with the address mask of the network entry and the destination IP address of 221.61.199.133

Bitwise AND

```
11111111 11111111 11111111 11100000 - address mask of 27
11011101 00111101 11000111 10000101 - destination IP
-----
11011101 00111101 11000111 10000000 - result of bitwise AND
```

Comparing the result of the bitwise AND with the network address:

```
10100010 10101000 00000000 00000000 - network entry in table
11011101 00111101 11000111 10000000 - result of AND
```

There is no match.

/*****/

The second network entry in the table

221.61.192.0/19

Bitwise AND

```
11111111 11111111 11100000 00000000 - address mask of 19
11011101 00111101 11000111 10000101 - destination IP
-----
11011101 00111101 11000000 00000000 - result of AND
```

Comparing the result of the bitwise AND, and the network address, it can easily be seen that there is a match:

```
Network address:    11011101 00111101 11000000 00000000
Bitwise AND result: 11011101 00111101 11000000 00000000
```

Hence the second network entry is a match with the destination address.

Activity 3.2

- i. Two fields of the IP datagram header are changed by routers: the **time-to-live** and the **checksum** fields.
- ii. Each router must decrement the **time-to-live**. Since the decremented time-to-live number will have changed the **checksum**, a new checksum must be calculated and added to the header to replace the current value. These two changes are the only changes that the router makes to the IP datagram header before it is encapsulated in a frame, fragmented into more than one frame, or discarded.

Activity 3.3

1. This is a class B network address, so the first 16 bits are used for network ID and the remaining 16 bits are for host id. We want to have 18 subnets with $2^4 < 18 \leq 2^5$, so we need to borrow 5 bits from the host ID. Hence, the subnet mask is:
11111111 11111111 11111000 00000000 or 255.255.248.0
2. The number of host bits are $32 - 21 = 11$ and the number of usable host per subnets is $2^{11} - 2 = 2046$
3. The first subnet address is
10011100 10011101 00000000 00000000 = 156.157.0.0
The second subnet address is:
10011100 10011101 00001000 00000000 = 156.157.8.0
The penultimate subnet is:
10011100 10011101 11110000 00000000 = 156.157.240.0
The last subnet address is
10011100 10011101 11111000 00000000 = 156.157.248.0
4. The range of the host addresses in the second subnet is:
10011100 10011101 00001000 0000 00001 = 156.157.8.1
to
10011100 10011101 0000 1111 1111 1110 = 156.157.15.254

Chapter 4**Activity 4.1**

6. TCP **does not** compress and decompress packets so they can be transferred efficiently.

Activity 4.2

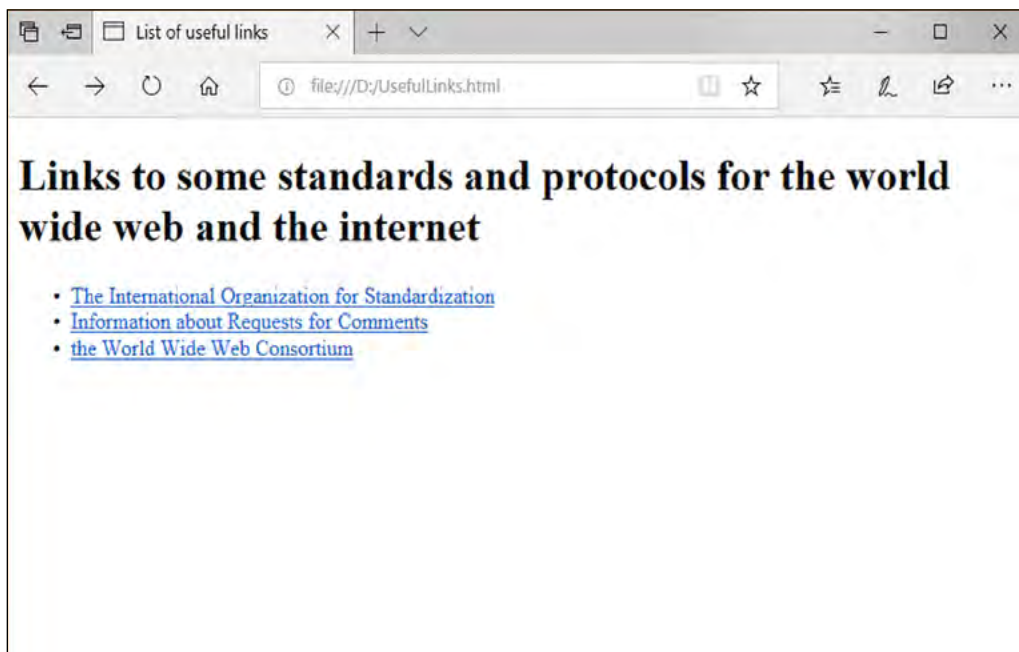
1. FALSE, both are Transport layer protocols.
2. FALSE, both headers have 32 bit words.
3. TRUE
4. FALSE, the well-known port number for HTTP is 80.
5. TRUE
6. FALSE, TCP implements flow control. UDP is simple and fast, but unreliable and does not guarantee that data packets sent will arrive in order, or that they will arrive at all, and makes no attempt at flow control.
7. FALSE, the receiver does not send requests for missing packets. The TCP protocol uses positive acknowledgement with retransmission which means that only those packets that have been correctly received will be acknowledged. Hence the cumulative acknowledgement scheme means that receivers will only confirm to the sender those data packets correctly received. The receiver will not ask to be sent packets that represent gaps in the data stream.
8. FALSE, fragmentation is done by the Internet Protocol.
9. TRUE, note that segments do not have to be received in order, they can come in any order, but once a receiver has ordered packets received using sequence numbers, only those that form the data stream from its beginning to a point x that has no gaps in the data will be acknowledged. Any segments that have been received after point x will be acknowledged when the missing segment or segments are received such that they are now part of an unbroken data stream.

10. FALSE, an acknowledgement with sequence number $x + 1$ implicitly acknowledges all segments with sequence numbers up to and including x , meaning that data packets can be implicitly acknowledged.
11. FALSE, the sender should assume that the data packet following on from the multiply acknowledged packet has been **lost or corrupted** and send it again
12. TRUE
13. FALSE, the SACK scheme is voluntary
14. FALSE, congestion collapse means that the network is so crowded with data packets that delivery slows, and packet loss increases, so much so that the network can no longer function effectively.
15. FALSE, the designers of the early internet assumed there would be few networks connected, hence they did not consider congestion control.

Chapter 5

Activity 5.1

1. `<!DOCTYPE html>`
2. It starts after the `<body>` tag, and ends when the tag is closed with `</body>`.
3. The HTML code above the `<body>` tag is used to give information to web browsers and search engines.
4. No answer is given, but a successfully written page should look something like the page below, when viewed in a web browser:



Activity 5.2

A correct rule-set would be:

```
body{
    background-color: #15FF0E;
}
```


Activity 5.3

There are two reasons: (1) because it is recommended that all formatting in HTML5 is done with CSS; and (2) because, assuming that the developer really wants the image to have the set dimensions, using the `style` attribute will prevent any document level or external style sheet from changing them.

Activity 5.4

Answers are in the files *activity4.css* and *tables4.html*, which can be downloaded from the VLE.

Chapter 6**Activity 6.1**

1. TRUE
2. TRUE, see section 6.5.2
3. FALSE, TRIPS extended copyright protection to computer programs and databases.
4. TRUE
5. FALSE, patent law is covered by more than one international convention, including TRIPS.
6. FALSE, under TRIPS copyright holders must be given a minimum level of rights and protections; individual countries may increase these rights and protections.
7. FALSE, the moral right is the right to be identified as the author/originator. It cannot be sold. Economic rights can be.

Activity 6.2

Trade secrets have legally enforceable protection in all countries ratifying the TRIPS agreement. Trade secrets are information that is commercially sensitive, and can be protected provided that the information is secret, has commercial value because it is secret and the owner of the information has taken reasonable steps to maintain the secrecy.

Source code may contain information that is commercially useful to rival organisations. Not all of this information will be protected by copyright as ideas and algorithms are not protected, and could be implemented by a commercial rival in code with enough differences for copyright not to apply. For example, the source code could be rewritten in a different language. This can be prevented by keeping the source code confidential, and protecting it as a trade secret, with the advantage that its lifetime is not limited as patent and copyright are, but lasts as long as the information is commercially sensitive. Like copyright, trade secret status does not have to be applied for. Trade secret status has advantages over patents, as it does not have to be applied for, and there are no legal fees in establishing a trade secret, and no delays due to the bureaucratic nature of the application process. In addition, trade secret status, unlike a patent, does not have to be applied for in every country that the business operates in.

Activity 6.3

- (i) 1. Hosting; caching; mere conduit
- (ii) 2. An official request to remove illegal web content, or to block access to it
- (iii) 3. Slander and libel are both forms of defamation

- (iv) 1. The Computer Misuse Act 1990 as amended, has three categories of offence. *Note the CMA as amended has five categories of offence.*

Activity 6.4

- (i) 4. None of the above
- (ii) 1. Private individuals
- (iii) 4. All of the above
- (iv) 3. Within 72 hours of the breach
- (v) 2. The **right to be forgotten** and the **right to data portability**

Chapter 7

Activity 7.1

Point 5 is untrue, Linux has good inbuilt security features.

Activity 7.2

- (i) 4. PUPs
- (ii) 1. Worms are autonomous and do not need to attach themselves to a host
- (iii) 2. Metamorphic
- (iv) 4. All of the above
- (v) 1. TRUE
- 2. TRUE
- 3. FALSE
- 4. FALSE. If you have a smart fridge it may be hacked into, and for example recruited into a botnet which is used in an attack on a bank, but it can't plan anything (yet); it's a fridge.

Activity 7.3

1. Patching means that as software vendors find or are notified of software vulnerabilities (bugs) they issue software 'patches'. Patches are quick fixes for the bug applied to the current version of the software; developers will most likely want to incorporate a better solution into the next version of the software.
2. The following are two reasons why it may be problematic in terms of computer security; you may be able to think of others:
 - Patches may not be applied in a timely way, or at all. This allows hackers to find and target unpatched systems.
 - Patching feeds into a paradigm of writing and releasing software quickly, and fixing mistakes at a later date. This means writing software quickly without the intention to test and get it right the first time. This may encourage poor practices on the part of some vendors.
3. The WannaCry attack was a worldwide ransomware attack in May 2017. The attack targeted Microsoft's Windows operating system, using a known vulnerability that Microsoft had issued a patch for in April 2017. Affected systems either had not used the patch, or were using older versions of the Windows operating system that were no longer supported by Microsoft.

Appendix 4: Sample examination paper: Part B

Important note: This Sample examination paper reflects the examination and assessment arrangements for this course in the academic year 2018–2019. The format and structure of the examination may have changed since the publication of this subject guide. You can find the most recent examination papers on the VLE where all changes to the format of the examination are posted.

Note that the examination covers both volumes of the guide, and is divided into two parts. You are expected to answer **two questions from Part A** covering material in Volume 1 of the guide; and **two questions from Part B**, which covers topics in Volume 2 (this subject guide). From each section you are required to choose two questions from a choice of three.

The following three questions are an example of Part B of the examination.

Question 4

(a)

(i) Which one of the following is true? [2 marks]

1. In February 2011, all IPv4 addresses had been issued by ICANN, meaning that the internet has run out of new IPv4 addresses.
2. IPv6 has a 256-bit address format, and was designed to allow for one trillion networks.
3. IPv4 addresses were initially designed to allow for an exponential increase in networks as the internet developed.
4. None of the above

(ii) Which one of the following is true about flow control in the Transmission Control Protocol? [2 marks]

1. The window field of the TCP header is used for flow control. It specifies the amount of data that the host is able to receive.
2. If a host is unable to accept data the Window field will be set to 0
3. The receiver sends a new, non-zero packet when they are able to accept data again.
4. All of the above

(iii) Which one of the following is a reason why packet switching makes the internet more fault tolerant? [2 marks]

1. Because packet switching allows for route tracing in order to identify choke points
2. Because packet switching includes tracking data that can be used to proactively address faults as they arise
3. Because packet switching allows for data to be switched to another path if a router becomes unavailable, which supports dynamic routing
4. None of the above

(b)

- (i) Say which TCP/IP protocols are described by the following definitions. You may give the initials of the protocols if you wish: [6 marks]
- (A) _____: is a standard network protocol used to exchange files.
- (B) _____: is a protocol that uses a simple transmission model with no methods for error checking, flow control or correct ordering of data packets. The protocol assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing in the transport layer.
- (C) _____: The purpose of these control messages is to provide feedback about problems in the communication environment, not to make IP reliable.
- (ii) Fill in the missing words in the following description of the routing algorithm. [3 marks]
1. Does the datagram's destination _____ match an address on this network? If so deliver.
 2. Is there an address in the _____ that matches the _____ portion of the IP address? If so reduce the _____ by one and forward the data packet to that address.
 3. Does the table contain a _____ routing address? If so reduce the _____ by one and forward the data packet to that address, else discard and send back to the source IP address an error message.
- (c) Describe the three-way handshake of the Transmission Control Protocol. Use a diagram to aid your explanation. [10 marks]

Question 5**(a)**

- (i) What was the original purpose of the world wide web? [2 marks]
1. To enable commerce to be done using the internet
 2. For the military to communicate more effectively
 3. For researchers to share documents
 4. None of the above
- (ii) HTML is an application of a markup language called: [2 marks]
1. XHTML
 2. XML
 3. SGML
 4. None of the above
- (iii) Which one of the following is true? [2 marks]
1. CSS stands for Cascading Style Sheets
 2. There are four distinct ways to format a web page with CSS
 3. The highest level of precedence for CSS formatting is given to the external style sheet
 4. None of the above

(b)

- (i) Explain the difference between the internet and the world wide web. [6 marks]
- (ii) What is email spoofing? Describe the mechanism in SMTP that allows email spoofing. [3 marks]

(c) Consider the following HTML5 code:

```
<!DOCTYPE html>
<html lang = "en">
  <head>
    <meta charset="UTF-8" />
    <title>A simple HTML5 web page</title>
    <link rel="stylesheet" href="examq.css">
  </head>

  <body>
    <h1>A top level heading</h1>
    <p>A paragraph. Note a paragraph has at most 8 sentences. This paragraph has three.</p>
    <h2>A second level heading</h2>
  </body>

</html>
```

The HTML5 file given above links to a CSS file, that contains the following rule-sets:

```
body{
  font-family: Courier, "Lucida Console", monospace;
  text-align: center;
}

p{
  font-weight: bold;
  font-style: italic;
}
```

- (i) Identify which part of the HTML code specifies that the document is in HTML5. [2 marks]
- (ii) Which one of the two following boxes, (A) and (B), contains text that is the most likely output of the HTML file? [4 marks]
- (iii) Add a rule-set to the CSS file such that all top-level headings will appear in orange text. [4 marks]

(A)

A top level heading

A paragraph. Note a paragraph has at most 8 sentences. This paragraph has three.

A second level heading

(B)

A top level heading

A paragraph. Note a paragraph has at most 8 sentences. This paragraph has three.

A second level heading

Question 6

(a) Say which of the following statements are true, and which are false.

[6 marks]

- (1) Proprietary software can be freely copied.
- (2) Copyright protects the authors and originators' rights to restrict (or allow) copies to be made of their software.
- (3) A patent is intellectual property protection for an invention.
- (4) A patent given in one country is automatically respected in another country, provided that both belong to the World Trade Organisation.
- (5) If the information is posted on the Internet, you can assume there is no copyright.
- (6) Patents must be applied for while copyright is automatically granted to an author or originator.

(b)

- (i) Briefly describe responsible disclosure in the context of computer security. [3 marks]
- (ii) What is a zero day vulnerability? [3 marks]
- (ii) What is a bug bounty in the context of computer security? [3 marks]

(c) In most jurisdictions computer software is granted author copyright, but, in general, cannot be patented, although there can be exceptions to this. Do you think that patenting software is a good idea in terms of encouraging innovation, providing choice for consumers and stimulating business? [10 marks]

Appendix 5: Sample examination answers: Part B

Question 4

(a)

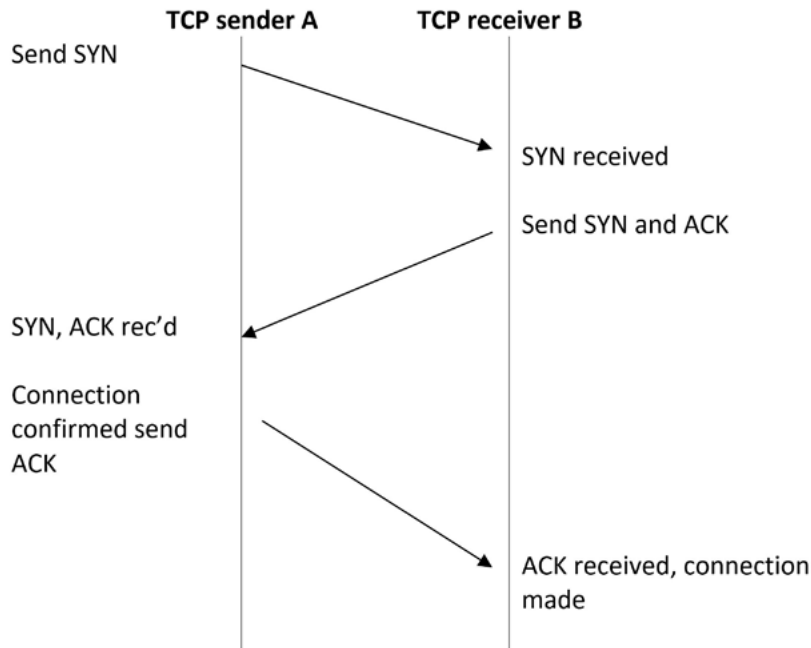
- (i) 1. In February 2011, all IPv4 addresses had been issued by ICANN, meaning that the internet has run out of new IPv4 addresses.
- (ii) 4. All of the above.
- (iii) 3. Because packet switching allows for data to be switched to another path if a router becomes unavailable, which supports dynamic routing.

(b)

- (i) (A) FTP: File Transport Protocol, is a standard network protocol used to exchange files.
- (B) UDP: is a protocol that uses a simple transmission model with no methods for error checking, flow control or correct ordering of data packets. The protocol assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing in the transport layer.
- (C) ICMP: The purpose of these control messages is to provide feedback about problems in the communication environment, not to make IP reliable.
- (ii)
 - 1 Does the datagram's destination IP address match an address on this network? If so deliver.
 - 2 Is there an address in the routing table that matches the network portion of the IP address? If so reduce the time-to-live by one and forward the data packet to that address.
 - 3 Does the table contain a default routing address? If so reduce the time-to-live by one and forward the data packet to that address, else discard and send back to the source IP address an error message.

(c) The three steps of the three-way handshake are as follows.

- 1. The TCP sender, A, sends a packet to the receiver, B, with the SYN bit on and a random value x for the Sequence number. Using a new, random Sequence number for each connection means that packets from previous connections will not be accepted.
- 2. If B receives the packet and accepts the connection, B sends a packet with SYN and ACK bits on, a random number y in its own Sequence field, and x + 1 in the Acknowledgement number field.
- 3. If A receives B's packet it has confirmation that routes are available in both directions and that the sequence numbers have been synchronised. B still needs confirmation of these factors so A sends one more packet to complete the 'handshake', without any flags switched on but Sequence = x + 1 and Acknowledgement = y + 1.

**Question 5****(a)**

- (i) 3. For researchers to share documents.
- (ii) 3. SGML.
- (iii) 1. CSS stands for Cascading Style Sheets.

(b)

- (i) The internet is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the internet. It forms the infrastructure for the world wide web.

The world wide web, or simply the web, is a way of accessing information over the medium of the internet. It is an information-sharing model that is built on top of the internet. The web uses the HTTP protocol to transmit data.

- (ii) Email spoofing is when an email is sent with a false address in the 'From' field of the header. Email spoofing is possible because SMTP commands are not linked to header fields. Thus it is possible to give one, correct, address to the MAIL FROM command, and put a completely different address in the 'From' field of the header that will be seen by the recipient. The protocol will not check and flag up the contradiction.

(c)

- (i) `<!DOCTYPE html>`

- (ii) (A)

- (iii)

```
h1 {
    color: orange;
}
```


Question 6**(a)**

- (1) FALSE
- (2) TRUE
- (3) TRUE
- (4) FALSE
- (5) FALSE
- (6) TRUE

(b)

- (i) **Responsible disclosure** is a system where researchers finding security vulnerabilities in software notify the vendor, giving them time to develop a patch before the public are notified. Public notification is delayed in order not to tip off criminals about the newly discovered vulnerability, but it is not delayed indefinitely in order to encourage vendors to develop a patch for the vulnerability.
- (ii) **Zero day vulnerability**: vulnerabilities used to attack hardware or software before the vendor even knows about the security issue.
- (iii) **Bug bounties** are operated by some software and hardware manufacturers, who pay researchers and others to notify them of any security vulnerabilities that they find, first.

(c) This question does not have a right or wrong answer. Marks will be awarded for sensible, relevant points, coherent argumentation, clarity and depth of knowledge demonstrated. A good answer might include the following points.

- If a vendor had a monopoly on a certain type of software via a patent, there may not be an incentive for them to develop better software for their customers.
- Patenting software could cause problems for the open source community. Open source software could be patented by others, and this would be a barrier to innovation.
- Patenting software would encourage vendors to become more concentrated, since patenting is very expensive, and needs deep pockets. Small business usually cannot afford the expense of paying for patents in every country they need them, and employing lawyers to help. Hence larger organisations would come to dominate the market completely and consumer choice would be decreased.
- Patents could help to ensure that innovators are rewarded for their work.
- Patents can help businesses to profit from their work, and these profits can be re-invested to produce further innovations.

Notes

Comment form

We welcome any comments you may have on the materials which are sent to you as part of your study pack. Such feedback from students helps us in our effort to improve the materials produced for the University of London.

If you have any comments about this guide, either general or specific (including corrections, non-availability of Essential readings, etc.), please take the time to complete and return this form.

Title of this subject guide:

Name

Address

Email

Student number

For which qualification are you studying?

Comments

[illegible]

Please continue on additional sheets if necessary.

Date:

Please send your completed form (or a photocopy of it) to:

Publishing Manager, Publications Office, University of London, Stewart House, 32 Russell Square, London WC1B 5DN, UK.