

# Inhoud

## Voorwoord 1

*Doel van dit boek* 1

## Hoofdstuk 1

### Netwerkarchitectuur en security 3

- 1.1 Leerdoelen 3
- 1.2 Netwerkarchitectuur 3
  - 1.2.1 *Netwerkmodellen* 4
  - 1.2.2 *Routers en switches* 5
- 1.3 Netwerkprotocollen 6
  - 1.3.1 *Transmission Control Protocol (TCP)* 6
  - 1.3.2 *File Transfer Protocol (FTP)* 11

## Hoofdstuk 2

### Internetarchitectuur 13

*Leerdoelen* 13

- 2.1 Internet Protocol (IP) 13
  - 2.1.1 *IP-adressen* 15
  - 2.1.2 *IPv4* 15
  - 2.1.3 *Subnetmasker* 17
  - 2.1.4 *Subnetten* 19
  - 2.1.5 *Standaard-gateway* 20
  - 2.1.6 *IPv6* 21
  - 2.1.7 *Hexadecimale notatie* 21
  - 2.1.8 *Netwerk- en host-ID's* 26
- 2.2 Routercomponenten 27
  - 2.2.1 *Dynamic Host Control Protocol (DHCP)* 27
  - 2.2.2 *Firewalls* 28
  - 2.2.3 *Network Address Translation (NAT)* 28
  - 2.2.4 *Gereserveerde publieke IP-adressen* 29
- 2.3 Domain Name System (DNS) 29
- 2.4 HTTP 31
  - 2.4.1 *HTTP-methodes* 33
- 2.5 OSI- en TCP/IP-modellen 37

## Hoofdstuk 3 Cryptografie 39

- Leerdoelen* 39
- 3.1 HTTPS en SSL 40
  - 3.1.1 SSL 40
  - 3.1.2 SSL-handshake 41
  - 3.1.3 OpenSSL 43
  - 3.1.4 HSTS 50
- 3.2 Encryptie-algoritmes (ciphers) 51
  - 3.2.1 Stream ciphers 51
  - 3.2.2 Block ciphers 55
  - 3.2.3 Authenticated Encryption (AE) 56
  - 3.2.4 Advanced Encryption Standard (AES) 58
  - 3.2.5 Andere encryptietools 63

## Hoofdstuk 4 Software-architectuur 65

- Leerdoelen* 66
- 4.1 UML-diagrammen 66
  - 4.1.1 UML-componentendiagram 66
  - 4.1.2 UML-deploymentdiagram 71
  - 4.1.3 UML Data Flow Diagram (DFD) 72
- 4.2 Software-architectuur-patterns 74
  - 4.2.1 Object Oriented Architectuur (OOA) 75
  - 4.2.2 Resource Oriented Architectuur (ROA) 77
  - 4.2.3 Service Oriented Architecture (SOA) 79
- 4.3 Proxy Server Architectuur 81
  - 4.3.1 Firewalls en filtering 82
  - 4.3.2 Scalability (schaalbaarheid) van architectuur 85
  - 4.3.3 Datacaching 86
  - 4.3.4 Web proxy servers 87

## Hoofdstuk 5 Application Programming Interface (API) 89

## Hoofdstuk 6 Pentest-omgeving inrichten 99

- Leerdoelen* 99
- 6.1 Pentestconfiguratie 99
- 6.2 VirtualBox installeren 101
- 6.3 Kali Linux virtuele machine installeren 101
  - 6.3.1 Harde schijf configureren 108
- 6.4 Advanced Packaging Tool (APT) 117
  - 6.4.1 apt-cache-commando's 118

- 6.4.2 *Synaptic* 119
- 6.5 Installeer LAMP 120
- 6.6 Visual Studio Code 123
- 6.7 Kali-instellingen klonen 124
- 6.8 OWASP Broken Web Applications installeren 126

## Hoofdstuk 7

### Pentesten 135

- Leerdoelen* 135
- 7.1 Reconnaissance 135
  - 7.1.1 *Scanning services met nmap* 136
  - 7.1.2 *Identificeren van applicatie-firewalls* 136
  - 7.1.3 *Spinnen en creëren van sitemaps van de applicatie met Burp en HTTP Track spiders* 139
  - 7.1.4 *Spiders en crawlers* 145
- 7.2 OWASP Foundation 148
- 7.3 Injection (SQL, OS, XXE en LDAP) 149
  - 7.3.1 *Automated scanners* 149
- 7.4 Broken authentication and session 153
- 7.5 Cross Site Scripting (XSS) 156
  - 7.5.1 *Automated scanner* 156
- 7.6 Broken access control 157
  - 7.6.1 *Privilege escalation attack* 158
- 7.7 Security misconfiguration 162
- 7.8 Sensitive data exposure 163
- 7.9 Insufficient attack protection 165
- 7.10 Cross-Site Request Forgery (CSRF) 167
- 7.11 Using components with known vulnerabilities 170
- 7.12 Underprotected APIs 170

## Hoofdstuk 8

### Veilig programmeren 171

- Leerdoelen* 171
- 8.1 Wat is beveiligde informatie? 171
  - 8.1.1 *Assets* 171
  - 8.1.2 *CIA-driehoek* 172
- 8.2 Secure Software Lifecycle (SSLC) 173
  - 8.2.1 *Het project VideoBox* 174
  - 8.2.2 *Doel van de app* 174
- 8.3 SSLC: Analyseren 175
  - 8.3.1 *Beveiligen tegen kwetsbaarheden* 176
- 8.4 SSLC: Ontwerpen 177

8.4.1	<i>Threat modeling</i>	178
8.5	SSLC: Testen van de plannen	183
8.5.1	<i>Handmatig testen</i>	183
8.5.2	<i>Geautomatiseerd testen</i>	183
8.6	SSLC: Coderen	184
8.6.1	<i>Principes van code design</i>	184
8.6.2	<i>Best practices en checklists</i>	185
8.6.3	<i>Bestrijding van de kwetsbaarheden</i>	187
8.6.4	<i>Foutafhandeling</i>	193
8.6.5	<i>Code review</i>	195
8.7	SSLC: Pentesten	198
8.8	SSLC: Implementeren	199
8.8.1	<i>Het DevOps-model</i>	200

## Register 201