Table of contents

Preface 11
Who are hackers? 12
Hacking School Mission 12

Introduction 15
Who is this book addressed to? 16
What will I learn from this handbook? 16
How to use the Hacking School Handbook? 17
Training Operating System and Live Training Videos 17
Summary 18

1. **Recovering lost passwords 25**
Using Advanced RAR Password Recovery 26
Recovery of DOC files with Advanced Office Password Recovery 31

2. **Interception of information over LANs 35**
What will we need, then? 37
A short presentation of the programs used 37
What is the ARP protocol? 39
How does it work? 40
Where do we find the target? 44

3. **Interception of encrypted data, attacks on SSL sessions 49**

4. **Entering the system by the backdoor 55**
Modification of /etc/passwd 56
Adding a new service 56
ICMP backdoor 60

5. **Hiding files using kernel modules 63**
Structure and compilation of modules 65

Compilation of kernel modules....66
Servicing modules through the kernel....68
System calls....69
More about registers....71
Registers....72
Service functions, sys_call_table....73
Access to sys_call_table in the new 2.6.x kernels....73
Substitution of functions in sys_call_table....74
Hiding files using the kernel module....76
Developing our module....81

6. **Buffer overflow attacks....85**
Memory....85
The stack....86
What is a buffer?....88
A simple example of the use of a buffer overflow....92
Advanced example of buffer overflow....94
Use of shellcodes....98
How not to make mistakes....108

7. **Practical examples of remote attacks....111**
Collecting information....111
Examination of web site....113
Choice of programming language....115
The Python language....116
Python modules....119
Writing an exploit....121
Practical uses of exploits....126

8. **Heap overflow attacks....131**
Memory segments....131
Heap....135
Buffer overflow....136
An example of heap overflow....138
An example of bss overflow....140

9. **Format string attacks 145**
What is a format string? 145
Incorrect use of the printf() function 147
Use of the %n tag 149
Practical use of the format string error 156
Using shellcodes 159
Overwriting the EIP copy 160
Overwriting the GOT section 164
Overwriting the DTORS section 166
How do we avoid errors? 168

10. **Practical examples of format string attacks 171**
Choosing software to attack 171
Obtaining access to the transferred shellcode address 174
Determining the best location for the shellcode in memory 178
Finding a location suitable for overwriting 179
Overwriting a specific location with the shellcode address 181
Problems with the query length 186

11. **File stream pointer overwrite attacks 189**
Exploiting the file stream pointers 189
File stream pointer exploitation 193
Attacking FreeBSD 194
Attacking the Linux system 200

12. **Errors on the system kernel level 211**
Kernel errors 211
Buffer overflows – a short reminder 212
Susceptible kernel modules 214
Creating a shellcode 221
Exploit 224
Real-life example - Bluetooth 227
Creating an exploit 229
Lack of address of bt_proto table 232

13. **Exploiting the ICMP protocol 237**
The ping tool 241

Determining the packet route using the traceroute program 243
Exploiting ICMP in DoS attacks 244
Ping flooding 245
Creating own ping flooder 246
Backdoor using ICMP 252
Sending data using ICMP 255
Scanning ICMP 264

14. **Remote identification of the operating system 267**
The stone age 267
OS fingerprinting 269
Active fingerprinting 269
How it works 271
Passive fingerprinting 278
Exploiting p0fa against nmap 282

15. **Netfilter and system security services 287**
Fingerprinting: a recap 287
Kernel modules 289
Netfilter 291
Filtration of packets 295
Port filtering 296
Instant modification of packets 299
Impersonate FreeBSD 305

16. **Securing the system step by step 309**
Preparation of the hard disk 309
Choice of installation 311
Administrator's password 311
Firewalls 312
The xinetd superserver 314
SSH configuration 315
Hiding information and the SUID bit 316
Software to improve system security 319
The kernel – the system's weak point 320
What next? 320

17. **Security scanners** **321**
What are scanners? 322
Nmap 322
Scanning techniques 323
Scanning using connect() 323
TCP SYN scanning 324
TCP FIN scanning 324
TCP ident scanning 325
UDP scanning with ICMP 325
UDP scanning with write() and recvfrom() 325
ICMP echo scanning 326
Fragmentation scanning 326
Scanning with version detection 326
Scanning with the IP protocol 327
ACK scanning 327
Scanning with the TCP window size 327
RPC scanning 328
List scanning 328
Scanning with ping 328
Idle scan 328
Starting up nmap 329
Introduction to Nessus 331
Configuration 332
Starting up Nessus 332
Creating rules for users 333
Using Nessus 334
Nikto 337
Installation 338
Use and options 338
Nikto configuration 341
Practical application of Nikto 343

18. **Improving security with patches** **347**
Grsecurity 349
Patching and configuring a new kernel 351
Compiling a kernel with grsecurity patch 356

Stack-Smashing Protector 359
LibSafe 371

19. **Intrusion detection systems 377**
What is an IDS? 377
Snort.....378
Installation of Snort 379
Configuration of Snort.....379
Use.....380
The rules of Snort.....383
Portsentry 389
Installation of Portsentry.....390
Using Portsentry.....391

20. **Attacking a web server 393**
Targets.....393
The Apache server and its possibilities.....394
PHP 394
Dangerous startup functions 395
Listing of directory contents and reading files.....396
PHP modules 398
Compilation of modules and related problems 406
Everything is blocked. What now? 408
It's impossible to do anything – Have we reached an impasse? 409
CGI.....409
The mod_python, mod_perl, and mod_* modules.....413

21. **Creating shellcodes in the Win32 environment..... 415**
What is a shellcode?.....415
Types of shellcodes 416
Finding the kernel address 416
Exploitation of hard-coded addresses.....416
Finding API addresses using the kernel's export section.....426
API functions 426
What the shellcode needs the API functions for.....427
The export section.....427

Finding API function addresses using the import address table 433
Shellcode to download and start up a Trojan horse using Win32-IF438
Win32 Internet Functions 438

Postscript 449