Computer Security

02r. Assignment 1 & Access Control Review

Paul Krzyzanowski • David Domingo • Ananya Jana
Rutgers University
Spring 2019
Question 1

What three Internet-enabled vulnerability categories does Paul Rosenzweig identify in his essays on cyberwarfare?

1. Anonymity
   - It is easy to attack anonymously – changing your cyber persona and attacking at a distance. Retaliation becomes difficult (or practically impossible).

2. Difficulty of distinction
   - Identifying specific activity on the network is difficult.
   - An attack requires access to a vulnerability. At the network level, authorized and unauthorized communications all look the same. It is difficult to tell if someone is attacking your system until the damage is done.

3. Asymmetry of power
   - In the physical world, the country of Monaco (land area = 2 km²) will never attack the U.S.
   - In the computer world, small states and non-state actors can challenge large nation states.
Why are scams with a minuscule chance of success deployed?

- Economics: a favorable cost-benefit ratio
- The cost of scamming (e.g., sending vast amounts of email) is tiny
- Even a tiny success makes the scam worthwhile
Starting in May 2018, companies doing business in Europe had to comply with the GDPR, the General Data Protection Regulation. Briefly explain the difference between anonymization and pseudonymization.

- **Anonymized data**
  - Data modified “in such a way that the data subject is not or no longer identifiable”
  - Any identifiable information must be discarded
  - Even the party that does the anonymization should not be able to identify a specific individual

- **Pseudonymized data**
  - Data modified “in such a way that the data can no longer be attributed to a specific data subject without the use of additional information.”
  - Data is personally identifiable only when both elements are combined.
Pseudonymization example

Personal data = any information that can identify you
- Name, home address, photo, banking info, email address, IP address

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Nationality</th>
<th>Job</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casey Webb</td>
<td>25</td>
<td>USA</td>
<td>UX Designer</td>
<td>72,000</td>
</tr>
<tr>
<td>Egil Wolff</td>
<td>51</td>
<td>Norway</td>
<td>Manager</td>
<td>105,000</td>
</tr>
<tr>
<td>Nana Nakajima</td>
<td>36</td>
<td>Japan</td>
<td>Programmer</td>
<td>211,000</td>
</tr>
<tr>
<td>Xiuying Li</td>
<td>31</td>
<td>China</td>
<td>UX Designer</td>
<td>84,000</td>
</tr>
<tr>
<td>Katharine Hoffmann</td>
<td>28</td>
<td>Germany</td>
<td>Tester</td>
<td>56,000</td>
</tr>
</tbody>
</table>

Non-identifiable data

<table>
<thead>
<tr>
<th>ID</th>
<th>Nationality</th>
<th>Job</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>USA</td>
<td>UX Designer</td>
<td>72,000</td>
</tr>
<tr>
<td>001</td>
<td>Norway</td>
<td>Manager</td>
<td>105,000</td>
</tr>
<tr>
<td>002</td>
<td>Japan</td>
<td>Programmer</td>
<td>211,000</td>
</tr>
<tr>
<td>003</td>
<td>China</td>
<td>UX Designer</td>
<td>84,000</td>
</tr>
<tr>
<td>004</td>
<td>Germany</td>
<td>Tester</td>
<td>56,000</td>
</tr>
</tbody>
</table>

Restricted: personally identifiable data

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Casey Webb</td>
<td>25</td>
</tr>
<tr>
<td>001</td>
<td>Egil Wolff</td>
<td>51</td>
</tr>
<tr>
<td>002</td>
<td>Nana Nakajima</td>
<td>36</td>
</tr>
<tr>
<td>003</td>
<td>Xiuying Li</td>
<td>31</td>
</tr>
<tr>
<td>004</td>
<td>Katharine Hoffmann</td>
<td>28</td>
</tr>
</tbody>
</table>

+  ⇒  Full data
Question 4

What four components constitute security engineering?

1. Policy
   Definition of what you are supposed to achieve

2. Mechanism
   Ciphers, access controls, tamper-resistant hardware, etc.

3. Assurance
   Amount of reliance (trust) you have in each mechanism

4. Incentive
   – The motivation that the people guarding & maintaining the system will do their job properly
   – The motivation that attackers will have to defeat your system
Access Control Discussion
MAC vs DAC

• **DAC = Discretionary Access Control**
  – The user is in charge of setting file permissions
  – If you own a file, you can set any access permissions you want on it … and even give it away
  – The root user (user ID 0) has the power to change any permissions

• **MAC = Mandatory Access Control**
  – System owner (administrator) defines security policies
  – Users cannot override them, regardless of their privilege level

• MAC takes priority over DAC
Subjects and objects

- **Subjects access objects**
  - They perform actions on objects

- **Subjects are users and processes**
  - Processes run with an ID, and hence privileges, of a user

- **Objects are resources**
  - Typically files and devices
  - They do not perform operations
SELinux (Security Enhanced Linux)

- Originally a kernel patch created by the NSA to add MAC to Linux

- Supports three MAC models:
  1. Type Enforcement (TE)
  2. Role-Based Access Controls (RBAC)
  3. Multi-Level Security (MLS) – the Bell-LaPadula Model
     - Multi-Category Security (MCS)
       - Extension of MLS to define categories within a security level

There other security models and implementations available in other distributions!
Type Enforcement (TE) on SELinux

Every subject (e.g., user) and object (e.g., file) on a system is assigned a label

- Processes are subjects – they run with the privileges of a user
- A label assigned to a process is called its domain
- A label assigned to an object (file) is called its type

Access control rules

The security administrator defines what access a domain (subject) can perform on a type (object)

```
allow userdomain bin_t:file: execute;
allow user2domain bin_t:file: read;
```

Allows users with the label "userdomain" execute rights for files with the label "bin_t" and allows users with the label "user2domain" read rights for those files
RBAC in SELinux

• Role-Based Access Control (RBAC) is integrated with the TE model

• Role-based access is specified in terms of TE
  – Management interface
  – Manage privileges based on roles users may assume
  – Control operations that a role can perform

• Essentially the same as TE but goal is to simplify labeling
  – A "role" just groups users and file operations
  – Easier conceptually than setting permissions between arbitrary domains and types

• Note: this doesn’t allow fine-grained roles, such as “access employee names” or “transfer funds”
MAC can reduce the need for root

• Traditionally the *root* user has supreme power
  – You need supreme power to do *any* administrative task

• Models such as TE and RSBAC allow you to define classes of users that can perform certain operations and access certain files
  – E.g., you can define a network administrators who can modify network configuration files and run network commands ... But not create user accounts or reboot the system
The end