Computer Security

02r. Assignment 1 & Access Control Review

Paul Krzyzanowski

TAs: Fan Zhang, Shuo Zhang

Rutgers University

Fall 2019

Question 1

What is meant by “security theater”?

Bruce Schneier writes:
"Security theater refers to security measures that make people feel more secure without doing anything to actually improve their security.”

Often applies to NOT totally useless measures but a focus on low-probability threats.

- Classic examples
  - – Airport security:
    - ineffective body scanners, no more than 200 ml of liquids, taking shoes off, …
  - – Random searches on subway stations: just go to a different station
  - – Disallowing photos of certain federal buildings, bridges, tunnels – the bad guys can do so in a covert manner anyway
  - – Security alert levels – *what are you supposed to do?*
- And the time I had to send a document to a bank, and they told me, “no you can’t email a scan – you have to mail or fax the original.”

Question 1: Security Theater Discussion

Examples of security theater in IT

- Forcing frequent password changes (people will pick even worse passwords)
- Website security certificates that make people think the servers are secure
  - All they do is allow you to validate who you're connecting to (but not always)
- Backing up files to a connected backup drive
  - Malware can wipe that data just as easily!
  - A fire or other site disaster cause the backup to be lost with the original

Question 1: Is Security Theater Useless?

- It's often a waste of money
  - Body scanners, extra staff that don't do anything useful
  - Threat detection software that doesn't work
  - Large scale espionage programs that ingest so much data that they can't find the real threats
- It can frustrate people
  - Example: don't allow a mother to bring baby formula on an airplane … but if you do allow it, a bad actor can smuggle other liquids disguised as baby formula
  - Why do I have to show my license to a guard when visiting a certain building even if they don't look me up as a registered visitor?
- It can give people a false sense of security
  - Nobody in this stadium has a knife or gun because everyone had to go through a metal detector.
- But it can make people feel better
  - Parents may feel more relaxed when visiting their baby in the hospital knowing it has an RFID tracker attached to its ankle
- And it can save your job
  - If a disaster occurs, a politician (or IT admin) will get fired for saying "it was a low-probability event so we didn't waste the money" versus saying "we spent billions but the system didn't catch the threat."

Question 2

What is the distinction between a “**subject**” and a “**principal**” when discussing security?

- A subject refers to a physical person in any role
  - This can be the operator or victim
- "A principal is an entity that participates in a security system"
  - This could be a subject or a program, computer communication channel, or a group of people.

Question 3

What is the distinction between a “**trusted system**” and a “**trustworthy system**”?

From page 13 of Security Engineering:

"A trusted system or component is one whose failure can break the security policy, while a trustworthy system or component is one that won’t fail."

A corrupt employee may be *trusted* (working in a position of trust) but not *trustworthy* (worthy of trust, corrupt).

Question 4

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 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 km^2) will never attack the U.S.
   - In the computer world, small states and non-state actors can challenge large nation states

Question 5

Why are scams with a minuscule chance of success deployed?

- Economics: a favorable cost-benefit ratio
- The cost of scamming is tiny
  - Example: it costs little or nothing to send huge amounts of email
- Even a tiny success makes the scam worthwhile

Question 6

What four components constitute security engineering?

From Ross Anderson, *Security Engineering*: Chapter 1, page 4

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`"
- Allows users with the label "`user2domain`" read rights for those files

RBAC in SELinux

Role-Based Access Control (RBAC) is integrated with the TE (Type Enforcement) 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 does not 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
  - Example: a network administrator can read – and modify – any files on the system
- Models such as TE and RBAC allow you to define classes of users that can perform certain operations and access certain files
  - E.g., you can define a network administrator who can modify network configuration files and run network commands ... but not create user accounts or reboot the system

The end