Operating Systems

Lesson 10

0368.2162.06	מערכות הפעלה	37	16	21	43.2%
0368.2162.07	מערכות הפעלה	32	6	26	18.8%
0368.2162.08	מערכות הפעלה	23	8	15	34.8%

Windows Security

- Protect from
  - Malicious user
  - A bug
- Account level security
  - Every action is performed under some process/thread
  - Every process/thread run under some account
  - Accounts and group of accounts has different privileges
- Accounts and groups examples
  - User account
  - System account
  - Guest account
  - Administrators groups
- System Privileges examples
  - Install software
  - Create or delete accounts
  - Change system date and time

Access Control Model

- Control the ability of a ***process*** to access ***securable objects***
- When a thread attempts to use a ***securable object***, the system performs an access check before allowing the thread to proceed.
  - Compares the security information in the thread's ***access token*** against the security information in the object's ***security descriptor***

Access Control Components

- **Access Token**
  - Contains information about a logged-on user (user, groups)
  - Generated on user logon
  - Contains a list of privileges
- **Security Descriptor**
  - Contains the security information that protects a securable object

Security Identifiers (SID)

- Identify entities that perform actions in the system
- Each account has a unique SID
- Each time a user logs on, the system retrieves the SID for that user
- The system uses the SID in the access token to identify the user

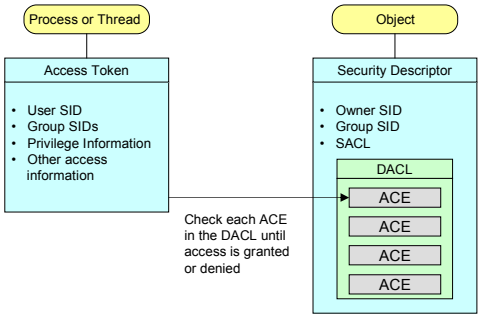
Access Control List (ACL)

- List of Access Control Entries (ACE)
- System Access Control List (SACL)
  - Used by administrators for logging
- Discretionary Access Control List (DACL)
  - identifies the entities that are allowed or denied access to a securable object

Access Control Entry (ACE)

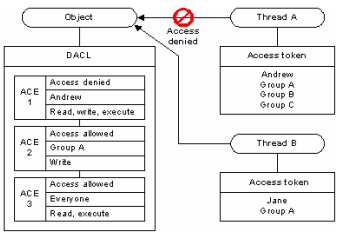
- A (SID) that identifies the entity to which the ACE applies
- Access mask that specifies the access rights
- Type of ACE (allow/deny)

Threads and Securable Objects



Controlling Access

- Allow access to one thread while denying access to another



Example: Token Dump

EX#5

- "Fortune teller" TCP socket server
  - 2 processes server.exe and client.exe
  - Client.exe <IP address> <port>
  - Server.exe <directory> <port>
  - Server will respond with "words of wisdom" to connecting clients
  - Client will connect, receive "words of wisdom", wait for any key from user and exit
  - Server will never exit

EX#5: Server

- Accept folder name (with trailing slash) and port as command line parameters
- In a folder will be 10 files named 0.txt 1.txt and 9.txt
- Server will accept connection from client, randomly select a file and send its content to a client, then disconnect
- Files are ASCII files with no more then 256 characters each

Ex#5: Client

- Gets Server's IP and port as command line parameters
- Connect on start
- Obtain data from socket and disconnect
- Print data to a console
- Wait for user's "any key"
- Exit

EX#5: Tips

- Server: 80 lines of code
- Client: 50 lines of code
- Data is ASCII but file names are in UNICODE
- Don't forget to add ws2_32.lib to a linker input properties in the project
- Use rand_s to generate random numbers

Ex#5: Challenge (optional)

- Server can handle only single client
- Open new thread for each connected client, pass socket handle
- Test using delay(sleep) between characters to see that indeed several clients are connected
- Open files using read sharing
- Traffic reduction: Send only single NULL character (use true string length)