COMP283-Lecture 4
Applied Database Management

Introduction	
DB Security	Authentication
	Authorisation
	Accounts and groups
	SQL Injection

COMP283-Lecture 4

DB Security: Authentication

- Authentication
  - DB Accounts
  - Server Accounts
  - Domain Accounts

- Authentication and Authorisation - the means for controlling who has access and to what.
- The security of your database system should be considered and planned for

COMP283-Lecture 4
DB Security: Authentication

- Authentication
  - The process of verifying who some principal is.
  - Usually achieved by declaring who the principal *is* and then providing a secret token known only by the principal and authenticator.
  - e.g. Withdrawing money from a cash machine.
  - In a DBMS, authenticator can be itself, or DBMS can be setup to trust some other system.
    - MS-SQL on Windows can itself authenticate, or can use local host's account manager (SAM).
    - Oracle on Linux can use host server's authentication system (e.g. password file or password system).

COMP283-Lecture 4

DB Security: Authentication

```
##
# User Database
#
# Note that this file is consulted directly only when the system is running
# in single-user mode.  At other times this information is provided by
# Open Directory.
#
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uucp:/usr/sbin/uucico
_taskgated:*:13:13:Task Gate Daemon:/var/empty:/usr/bin/false
_networkd:*:24:24:Network Services:/var/networkd:/usr/bin/false
_installassistant:*:25:25:Install Assistant:/var/empty:/usr/bin/false
_lp:*:26:26:Printing Services:/var/spool/cups:/usr/bin/false
_postfix:*:27:27:Postfix Mail Server:/var/spool/postfix:/usr/bin/false
_scsd:*:31:31:Service Configuration Service:/var/empty:/usr/bin/false
_ces:*:32:32:Certificate Enrollment Service:/var/empty:/usr/bin/false
_mcxalr:*:54:54:MCX AppLaunch:/var/empty:/usr/bin/false
_appleevents:*:55:55:AppleEvents Daemon:/var/empty:/usr/bin/false
_geod:*:56:56:Geo Services Daemon:/var/db/geod:/usr/bin/false
_serialnumberd:*:58:58:Serial Number Daemon:/var/empty:/usr/bin/false
_devdocs:*:59:59:Developer Documentation:/var/empty:/usr/bin/false
_sandbox:*:60:60:Seatbelt:/var/empty:/usr/bin/false
_mdnsresponder:*:65:65:mDNSResponder:/var/empty:/usr/bin/false
_ard:*:67:67:Apple Remote Desktop:/var/empty:/usr/bin/false
_www:*:70:70:World Wide Web Server:/Library/WebServer:/usr/bin/false
```

COMP283-Lecture 4

DB Security: Authentication

- Domain Authentication - Directory Services e.g.:
  - Novell Directory Services
  - Apple Open Directory (OD)
  - Microsoft's Active Directory Services (AD)
  - DBMS trusts the directory service system to verify the user
  - With AD, account credentials are passed to the Domain Controller in encrypted form. Details are checked and if OK an access token is returned.

DB Security: Authentication

- Domain Authentication
  - Organisations already have accounts for users, separate from the DBMS.
  - With Directory Services (DS) users can be members of groups, with different permissions / access etc.
  - New staff will be given a user account in DS as part of standard enrolment.

COMP283-Lecture 4

DB Security: Authentication

- Domain Authentication
  - +ve Database Administrator doesn't necessarily need to change anything to setup access to the db for new users.
  - -ve Unlikely that someone outside of organisation will have a domain account.
    - Might not be company policy to set up extra users
  - e.g. Web store may not be able to use Domain Authentication

COMP283-Lecture 4

DB Security: Authorisation - Roles

- Authorisation
  - Relates to the control of what you are allowed access to.
  - Even though authentication has verified who you are, doesn't mean you can access everything you want.
  - Access allowed within db is defined via *Roles* and *Permissions*.
  - Job of the DBA to manage roles and permissions for creating databases, for database objects (e.g. tables) and access to data.
  - Roles describe a type of task that may need to be performed.
  - A set of permissions required to perform those tasks will be assigned to the role.

COMP283-Lecture 4

DB Security: Authorisation - Roles

- Authorisation
  - Predefined Roles in MS-SQL
    - Bulkadmin - To perform bulk insert operations
    - Dbcreator - Create, alter, drop, and restore databases
    - Diskadmin - Create, alter, and drop disk files
    - Processadmin - Kill running SQL server processes
    - Securityadmin - Manage logins for a server
    - Serveradmin - Configure server wide settings, configure full-text searches, start or stop the server
    - Setupadmin - Configure linked servers, extended stored procedures, and startup stored procedure
    - Sysadmin - Has full access to any task regardless of any specified permissions, it's a super-user
  - User-Defined Roles

COMP283-Lecture 4

DB Security: Authorisation – Permissions

- Data Manipulation Language (DML) permissions
  - Select
  - Insert
  - Update
  - Delete
  - DRI (References) Declarative Referential Integrity
  - Execute

- Permissions related to database DATA (usually assigned for a table, but can be assigned to individual columns)

COMP283-Lecture 4

DB Security: Authorisation – Permissions

- Data Definition Language (DDL) permissions
  - CREATE
  - ALTER
  - DROP

- Permissions related to database OBJECTS, e.g. tables, views, stored procedures, etc.

COMP283-Lecture 4

DB Security: Authorisation – Permissions

- Data Control Language (DCL) permissions
  - GRANT
  - DENY
  - REVOKE

- Used to assign or remove permissions.
- e.g. When you create a new table, DBMS will assign some default roles and permissions to it. You must modify the permissions to suit your security plan.
- A common task would be to assign the SELECT permission to a user on a particular table, when you do so you can also use the "With Grant" permission. This extra permission allows the specified user to also be able to grant the same level of access to another user.

COMP283-Lecture 4

DB Security: Authorisation – Permissions

- Using Domain Authentication may well help when assigning permissions - members of the organisation with similar responsibilities will probably already be assigned to domain groups.
- You can assign roles and permissions to these domain groups rather than having to assign them to each individual.

- GRANT can include a special permission "WITH GRANT"
  - An entity that is granted a permission WITH GRANT gains the ability to assign the same level (or lower) permission to other entities.
- DENY explicitly disallows a permission.
- REVOKE removes a permission setting from the entity
  e.g. If user FRED has SELECT, UPDATE on table "My_Table",
  Then the statement:

```
REVOKE UPDATE ON My_Table FROM FRED;
```

  Will leave FRED with only SELECT on table "My_Table".

COMP283-Lecture 4

DB Security: Authorisation – Permissions

```
-- check the current user
select user();

create table t1 (myId int, mydata varchar(200), myName varchar(200));

insert t1 select 1, 'my data yes', user();
insert t1 select 2, 'my data yes2', user();
insert t1 select 3, 'my data no', 'joe';

select * from t1;

create or replace view v1 AS
select * from t1 where myName = user();

select * from v1

--
--

GRANT SELECT ON <database name>.<view name>
TO <user>@<host> IDENTIFIED BY '<password>'
```

COMP283-Lecture 4

DB Security: Security Hierarchy

- DENY > GRANT
- DELETE > UPDATE > SELECT

- e.g. security on table MyTable
- Imagine a user Fred who logs in to the database using a Windows Domain account and is a member of several Domain groups:
  - USERS, Students, and Computer Science
- Fred's account is also a member of a database server local group: DB_Group1.

DB Security: Security Hierarchy

- Fred is directly assigned a GRANT SELECT permission on the MyTable object by the DBA.
- Fred is also assigned to the user-defined role "DB_Admin", which has been given the GRANT UPDATE permission.
- The Domain USERS group has been assigned to the user-defined role "DB_Users", which has been given the GRANT SELECT permission.

Entity	Role	Permission
Fred		GRANT SELECT
Fred	DB_Admin	GRANT UPDATE
USERS	DB_Users	GRANT SELECT

DB Security: Security Hierarchy

- The Domain STUDENTS group has been assigned to the user-defined role "DB_Student", which has been given the DENY SELECT permission.
- The Domain Computer_Science group has been directly given the GRANT SELECT permission.
- The Local DB_Group1 group has been assigned to the user-defined role "DB_Group1", which has also been given the GRANT SELECT permission.

Entity	Role	Permission
Fred		GRANT SELECT
Fred	DB_Admin	GRANT UPDATE
USERS	DB_Users	GRANT SELECT
STUDENTS	DB_Student	DENY SELECT
Computer_Science		GRANT SELECT
DB_Group1	DB_Group1	GRANT SELECT

DB Security: Security Hierarchy

Entity	Role	Permission
Fred		GRANT SELECT
Fred	DB_Admin	GRANT UPDATE
USERS	DB_Users	GRANT SELECT
STUDENTS	DB_Student	DENY SELECT
Computer_Science		GRANT SELECT
DB_Group1	DB_Group1	GRANT SELECT

- What access does Fred have to the table MyTable?

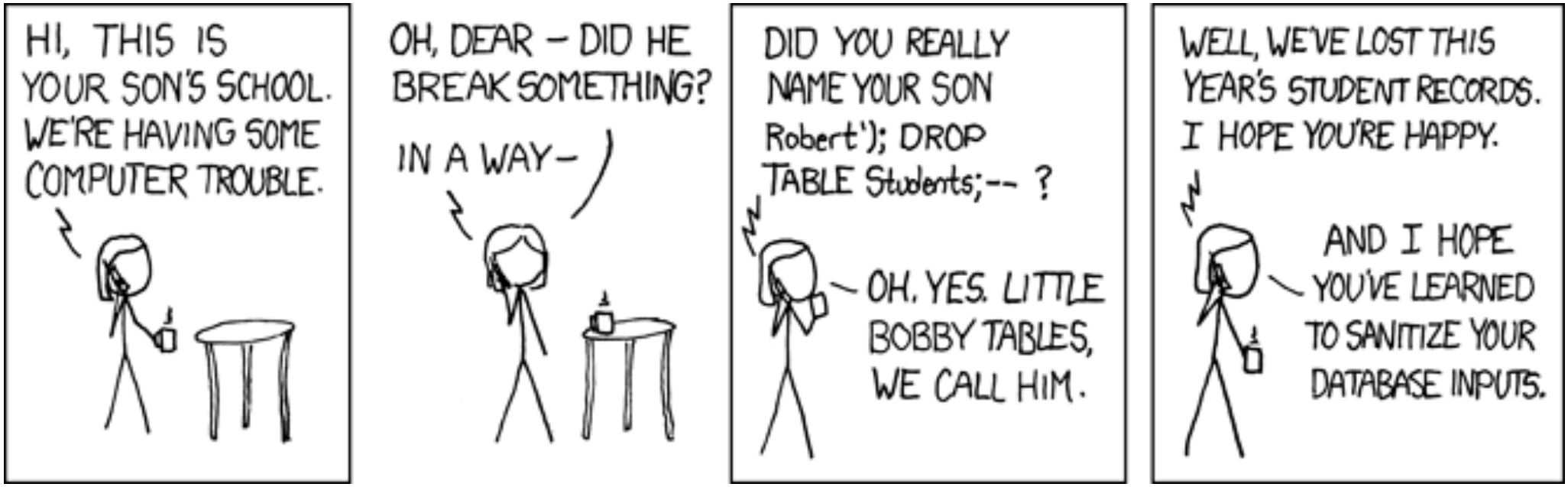
COMP283-Lecture 4

DB Security: Other Considerations

- Physical security.
  - Restrict physical access to the server to authorised persons
- Database integrity.
  - Ensure that other apps running on the server don't interfere with the DBMS, or consume too many resources.
- SQL Injection.
  - Abusing a client application by inserting executable code or a system command into an SQL query that is run on the server.

COMP283-Lecture 4

DB Security: Other Considerations



https://www.xkcd.com/327/

DB Security: Other Considerations

- To prevent these attacks the client application should be written to check and verify any user input.
- Some programming languages simplify this for you by providing functions or modules to deal with these checks.
  - e.g. in php use parameterised execution aka *prepared statements*
  - Any user input gets passed as parameters to the query code, and the parameters are tested before being passed to the database server for execution.
- Alternatively you could process the entered parameter (user input) and test for unwanted escape characters and key words.
- Could run the database using a restricted user account – the account will be granted enough database permissions and roles to support the client application, but no privileges on the server.

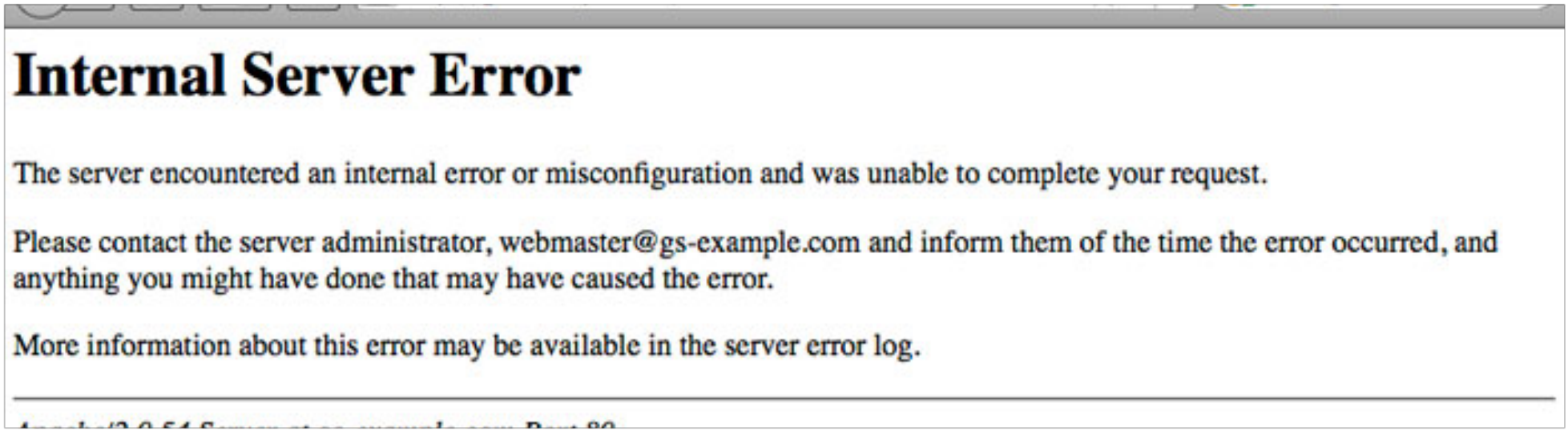
COMP283-Lecture 4

DB Security: Other Considerations

```
SELECT data
      FROM table
         WHERE Emailinput = '$email_input';
```

hacker@programmerinterview.com'

```
SELECT data
      FROM table
         WHERE Emailinput = 'hacker@programmerinterview.com'';
```



Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator, webmaster@gs-example.com and inform them of the time the error occurred, and anything you might have done that may have caused the error.

More information about this error may be available in the server error log.

COMP283-Lecture 4

DB Security: Other Considerations

- After a bit more experimentation by the hacker they are able to figure out or guess some of the table names and then…

```
Y';
    UPDATE table
     SET email = 'hacker@ymail.com'
     WHERE email = 'joe@ymail.com';
```

```
SELECT data
          FROM table
              WHERE Emailinput = 'Y';
     UPDATE table
      SET email = 'hacker@ymail.com'
      WHERE email = 'joe@ymail.com';
```

COMP283-Lecture 4

DB Security: Other Considerations

- Using prepared statements, the parameter is inserted in an SQL template. It cannot become an SQL command.

```
$preparedStatement = $db->prepare('INSERT INTO table (column) VALUES (:column)');

$preparedStatement->execute(array('column' => $unsafeValue));
```

COMP283-Lecture 4

Conclusion

- Introduced database security roles and permissions.
- Identified other security considerations.
- Introduced authentication methods/systems.
- Talked a little about SQL Injection