CS-457
Recitation No.2

Eva Papadogiannaki (papadogian)
Michalis Pachilakis (mipach)
Database & Cloud Security: Review Questions
Database & Cloud Security: Review Question No.1

List and briefly describe some administrative policies that can be used with a RDBMS
Database & Cloud Security: Review Question No.1

List and briefly describe some administrative policies that can be used with a RDBMS

- Centralized administration: A small number of privileged users may grant and revoke access rights.
- Ownership-based administration: The owner (creator) of a table may grant and revoke access rights to the table.
- Decentralized administration: In addition to granting and revoking access rights to a table, the owner of the table may grant and revoke authorization rights to other users, allowing them to grant and revoke access rights to the table.
Database & Cloud Security: Review Question No.2

*Explain the concept of cascading authorizations.*
Database & Cloud Security: Review Question No.2

*Explain the concept of cascading authorizations.*

The grant option enables an access right to cascade through a number of users.

If a user has an access right with grant option, the user may pass the right to another user.

When user A revokes an access right, any cascaded access right is also revoked, unless that access right would exist even if the original grant from A had never occurred.
Database & Cloud Security: Review Question No.3

Describe some of the main cloud-specific security threats.
Database & Cloud Security: Review Question No.3

Describe some of the main cloud-specific security threats.

- Abuse and nefarious use of cloud computing
- Insecure interfaces and APIs
- Malicious insiders
- Shared technology issues
- Data loss or leakage
- Account or service hijacking
- Unknown risk profile
Database & Cloud Security: Problems
Database & Cloud Security: Problem No.1

Consider an SQL statement:

```
SELECT id, forename, surname FROM authors WHERE forename = 'john' AND surname = 'smith'
```

a. What is this statement intended to do?

b. Assume that the forename and surname fields are being gathered from user-supplied input, and suppose the user responds with:

Forename: jo’hn
Surname: smith

What will be the effect?
Database & Cloud Security: Problem No.1

Consider an SQL statement:

```
SELECT id, forename, surname FROM authors WHERE forename = 'john' AND surname = 'smith'
```

a. What is this statement intended to do?

b. Assume that the forename and surname fields are being gathered from user-supplied input, and suppose the user responds with:

Forename: jo’hn
Surname: smith

What will be the effect?
Database & Cloud Security: Problem No.1

Consider an SQL statement:

```
SELECT id, forename, surname FROM authors WHERE forename = 'john' AND surname = 'smith'
```

c. Now suppose the user responds with:

```
Forename: jo'; DROP table authors--'
Surname: smith
```

What will be the effect?
Database & Cloud Security: Problem No.1

Consider an SQL statement:

```
SELECT id, forename, surname FROM authors WHERE forename = 'john' AND surname = 'smith'
```

(c. Now suppose the user responds with:

```
Forename: jo'; DROP table authors--'
```

```
Surname: smith
```

What will be the effect?

```
=> Executes the DROP request (deletes table authors)
```
1. String login, password, pin, query
2. login = getParameter(“login”);
3. password = getParameter(“pass”);
3. pin = getParameter(“pin”);
4. Connection
   conn.createConnection(“MyDataBase”);
5. query = “SELECT accounts FROM users WHERE login=’” +
   login + “’AND pass=’” + password + “’AND pin=” + pin;
8. ResultSet result =
   conn.executeQuery(query);
9. if (result!=NULL)
   displayAccounts(result);
10 else
   displayAuthFailed();
Database & Cloud Security: Problem No.2

Figure shows a fragment of code that implements the login functionality for a database application. The code dynamically builds an SQL query and submits it to a database.

a. Suppose a user submits login, password, and pin as **doe**, **secret**, and **123**. Show the SQL query that is generated.

```
SELECT accounts FROM users WHERE login='doe' AND pass='secret' AND pin=123
```

1. String login, password, pin, query
2. login = getParameter("login");
3. password = getParameter("pass");
4. pin = getParameter("pin");
5. Connection
6. conn.createConnection("MyDataBase");
7. query = "SELECT accounts FROM users WHERE login='" +
8. login + "' AND pass='" + password +
9. "' AND pin=" + pin;
10. ResultSet result =
11. conn.executeQuery(query);
12. if (result!=NULL)
13. displayAccounts(result);
14. else
15. displayAuthFailed();
Database & Cloud Security: Problem No.2

Figure shows a fragment of code that implements the login functionality for a database application. The code dynamically builds an SQL query and submits it to a database.

b. Instead, the user submits for the login field the following: ‘or 1 = 1 –’

What is the effect?

1. String login, password, pin, query
2. login = getParameter("login");
3. password = getParameter("pass");
4. pin = getParameter("pin");
5. Connection
   conn.createConnection("MyDataBase");
6. query = "SELECT accounts FROM users WHERE login=’” +
   “‘AND pass=’” + password +
   “‘AND pin=” + pin;
7. ResultSet result =
   conn.executeQuery(query);
8. if (result!=NULL)
   displayAccounts(result);
9. else
   displayAuthFailed();
Figure shows a fragment of code that implements the login functionality for a database application. The code dynamically builds an SQL query and submits it to a database.

b. Instead, the user submits for the login field the following: `’ or 1 = 1 --`

What is the effect?

**SELECT accounts FROM users WHERE login=’’ OR 1 = 1 -- AND pass=’’ AND pin=**
Database & Cloud Security: Problem No.2

Figure shows a fragment of code that implements the login functionality for a database application. The code dynamically builds an SQL query and submits it to a database.

b. Instead, the user submits for the login field the following: ‘ or 1 = 1 – –

What is the effect?

SELECT accounts FROM users WHERE login=’ OR 1 = 1 -- AND pass=’ AND pin=

Returns every account in table users

1. String login, password, pin, query
2. login = getParameter(“login”);
3. password = getParameter(“pass”);
3. pin = getParameter(“pin”);
4. Connection
conn.createConnection(“MyDataBase”);
5. query = “SELECT accounts FROM users WHERE login=’” +
6. login + “ ‘AND pass=’” + password +
7. “‘AND pin=” + pin;
8. ResultSet result =
conn.executeQuery(query);
9. if (result!=NULL)
10 displayAccounts(result);
11 else
12 displayAuthFailed();
The SQL command word **UNION** is used to combine the result sets of 2 or more SQL SELECT statements. For the login code of the figure, suppose a user enters the following into the login field:

‘UNION SELECT cardNo from CreditCards where acctNo = 10032 --

What is the effect?

1. String login, password, pin, query
2. login = getParameter("login");
3. password = getParameter("pass");
4. pin = getParameter("pin");
5. Connection
   conn.createConnection("MyDataBase");
6. query = "SELECT accounts FROM users WHERE login='" +
   login + "'AND pass='" + password +
   "'AND pin='" + pin;
7. ResultSet result =
   conn.executeQuery(query);
8. if (result!=NULL)
   displayAccounts(result);
9. else
   displayAuthFailed();
Database & Cloud Security: Problem No.3

The SQL command word UNION is used to combine the result sets of 2 or more SQL SELECT statements. For the login code of the figure, suppose a user enters the following into the login field:

\[
UNION SELECT cardNo from CreditCards where acctNo = 10032
\]

What is the effect?

SELECT accounts FROM users WHERE login=’’ UNION SELECT cardNo from CreditCards WHERE acctNo = 10032 -- AND pass=’’ AND pin=

1. String login, password, pin, query
2. login = getParameter(“login”);
3. password = getParameter(“pass”);
4. pin = getParameter(“pin”);
5. Connection
   conn.createConnection(“MyDataBase”);
6. query = “SELECT accounts FROM users WHERE login=’’ + login + “‘AND pass=’’ + password + “‘AND pin=’” + pin;
7. ResultSet result = conn.executeQuery(query);
8. if (result!=NULL)
   10 displayAccounts(result);
9. else
   11 displayAuthFailed();
The SQL command word **UNION** is used to combine the result sets of 2 or more SQL **SELECT** statements. For the login code of the figure, suppose a user enters the following into the login field:

```
'UNION SELECT cardNo from CreditCards where acctNo = 10032 --
```

**What is the effect?**

SELECT accounts FROM users WHERE login=’’ UNION SELECT cardNo from CreditCards WHERE acctNo = 10032 -- AND pass=’’ AND pin=

Returns the account info in table **users**
Malicious Software: Review Questions
Malicious Software: Review Question No.1

*What mechanisms can a virus use to conceal itself?*
Malicious Software: Review Question No.1

What mechanisms can a virus use to conceal itself?

A virus classification by concealment strategy includes the following categories:

- Encrypted virus: uses encryption to obscure its content
- Stealth virus: designed to hide itself from detection (using code mutation, compression, or rootkit techniques)
- Polymorphic virus: creates copies during replication that are functionally equivalent but have distinctly different bit patterns
- Metamorphic virus: mutates with every infection, rewriting itself completely at each iteration, using multiple transformation techniques
Malicious Software: Review Question No.2

*What is the difference between machine executable and macro viruses?*
Malicious Software: Review Question No.2

What is the difference between machine executable and macro viruses?

A **machine executable code** can be prepended or postpended to some executable program, or it can be embedded into it in some other fashion. The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.

**Macro viruses** infect scripting code used to support active content in a variety of user document types. Macro viruses take advantage of support for active content using a scripting or macro language, embedded in a word processing document or other type of file.
Malicious Software: Review Question No.3

*What means can a worm use to access remote systems to propagate?*
Malicious Software: Review Question No.3

What means can a worm use to access remote systems to propagate?

To replicate itself, a worm uses some means to access remote systems. These include the following, most of which are still seen in active use:

- Electronic mail or instant messenger facility
- File sharing
- Remote execution capability
- Remote file access or transfer capability
- Remote login capability
Malicious Software: Review Question No.4

What is a “logic bomb”? 
Malicious Software: Review Question No.4

What is a “logic bomb”?

A key component of data-corrupting malware is the logic bomb. The logic bomb is code embedded in the malware that is set to “explode” when certain conditions are met. Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage.
Malicious Software: Review Question No.5

What is the difference between a “phishing” attack and a “spear-phishing” attack, particularly in terms of who the target may be?
Malicious Software: Review Question No.5

What is the difference between a “phishing” attack and a “spear-phishing” attack, particularly in terms of who the target may be?

A phishing attack is an approach used to capture a user’s login and password credentials is to include a URL in a spam e-mail that links to a fake Web site controlled by an attacker, but which mimics the login page of some banking, gaming, or similar site. Given sufficient details, the attacker can then “assume” the user’s identity for the purpose of obtaining credit, or sensitive access to other resources.

A more dangerous variant of this is the spear-phishing attack. However, the recipients are carefully researched by the attacker, and each e-mail is carefully crafted to suit its recipient, specifically, often quoting a range of information to convince them of its authenticity. This greatly increases the likelihood of the recipient responding as desired by the attacker.
Malicious Software: Problems
Malicious Software: Problem No.1

There is a flaw in the virus program of Figure. What is it?

```
program V
1234567;

procedure attach-to-program;
begin
  repeat
    file := get-random-program;
    until first-program-line := 1234567;
  prepend V to file;
end;

procedure execute-payload;
begin
  (* perform payload actions *)
  b;
  // ...... ⇒

procedure trigger-condition;
begin
  (* return true if trigger condition is true *)
end;

begin (* main action block *)
  attach-to-program;
  if trigger-condition then execute-payload;
  goto original program code;
end;
```
Malicious Software: Problem No.1

There is a flaw in the virus program of Figure. What is it?

The virus is easily detected because the infected version of the program is longer than the corresponding uninfected one (in this example, the virus will execute until all files are infected).
Malicious Software: Problem No.2

Consider the following fragment in an authentication program:

```java
username = read_username();
password = read_password();
if username is "133t h4ck0r"
    return ALLOW_LOGIN;
if username and password are valid
    return ALLOW_LOGIN
else return DENY_LOGIN
```

What type of malicious software is this?
Malicious Software: Problem No.2

Consider the following fragment in an authentication program:

```python
username = read_username();
password = read_password();

if username is "133t h4ck0r"
    return ALLOW_LOGIN;
if username and password are valid
    return ALLOW_LOGIN
else return DENY_LOGIN
```

What type of malicious software is this?  => Backdoor
Malicious Software: Problem No.3

Assume you have found a USB memory stick in your work parking area. What threats might this pose to your work computer should you just plug the memory stick in and examine its contents? In particular, consider whether each of the malware propagation mechanisms we discuss could use such a memory stick for transport. What steps could you take to mitigate these threats, and safely determine the contents of the memory stick?
Malicious Software: Problem No.3

The USB memory stick may pose a range of threats to the confidentiality, integrity and availability of a system. Some of them are:

● An executable virus or document infected with a macro virus, which if run or opened can allow the virus to run and spread
● A worm that can run automatically using the autorun capability or by exploiting some vulnerability when the USB stick is viewed
● A trojan horse program or file that would threaten the system if installed or allowed to run

Countermeasures: Examine the memory stick in a controlled environment, (e.g. live-boot linux) which cannot be changed even if the malware manages to run.
Questions?