CSE 3421

Software Security

SUMMER 2021

MD. RAFI-UR-RASHID

LECTURER, DEPT. OF CSE, UIU

Why This Lesson?

An unsecured system is not much attractive for the users, but it is for the hackers

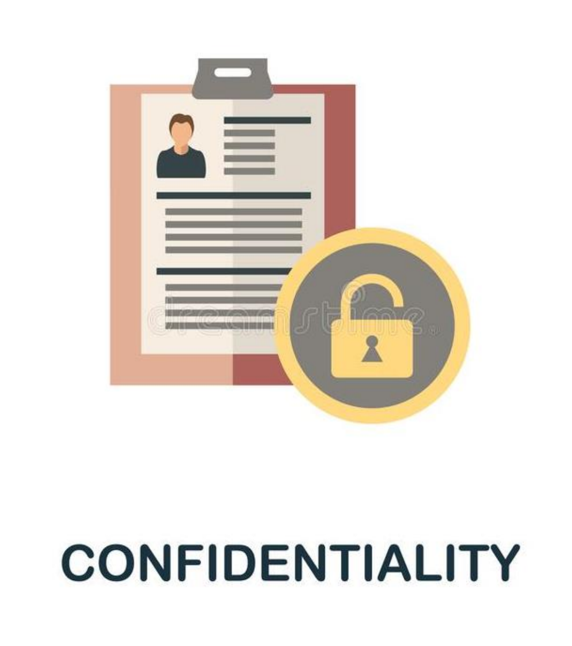
Properties of secure software

- Confidentiality
- Integrity
- Authentication
- Non-repudiation
- Availability

Confidentiality

- The ability of a system to ensure that an asset is viewed only by authorized parties
- Sensitive information is not leaked to unauthorized parties
- Privacy for individuals, confidentiality for data

- **Examples**
  - Exam assignments should not be published (at least until the exam starts)
  - GPA should only be visible to the particular student involved



Integrity

- The ability of a system to ensure that an asset is modified only by authorized parties
- Sensitive information is not damaged by unauthorized parties
- **Examples**
  - Submissions are not edited by anyone other than student
  - Grades are determined only by instructor or auto-grader

Authentication

- The ability to verify the identity of an individual or entity.

- Verify the identity of a person (or other external agent) making a request of a computer system

- **Example**
  - Should be able to determine if the user is indeed a student or an instructor

Nonrepudiation (accountability)

- The ability of a system to confirm that a doer cannot convincingly deny having done something

- Note: opposite of privacy/anonymity; requires a balance

- **Examples**
  - Student cannot deny to have edited submission after the deadline

Availability

- The ability of a system to ensure that an asset can be used by any authorized parties
- A system is responsive to requests
- Sometimes unauthorized control over the system makes it unavailable.
- Another scenario is DOS (Denial of Service)

Core principles

- **Identity:** is how a user tells a system who he or she is (for example, by using a username or User ID);

- **Authentication:** is the process of verifying a user's claimed identity (for example, by comparing an entered password to the password stored on a system for a given username).;

- **Authorization:** defines a user's rights and permissions on a system. After a user (or process) is authenticated, authorization determines what that user can do on the system.;

- **Auditing:** Keep track of users' interactions in a system.

Popular Cyber Attacks

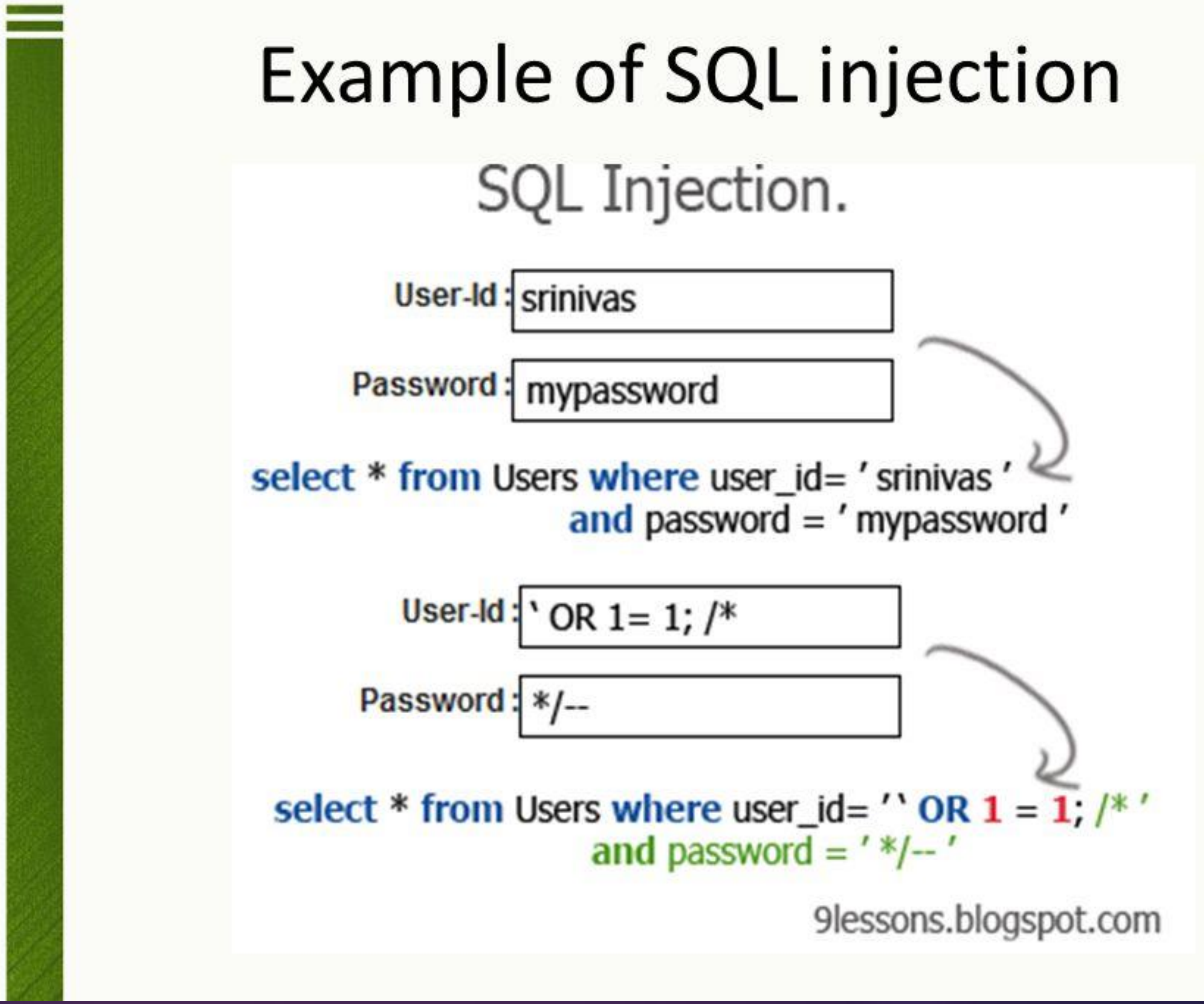
Malware

- Unwanted software that is installed on your system without your consent.
- **Ransomware**: this is malicious software that holds your data hostage until a ransom is paid.
- **Spyware:** These are programs installed to collect confidential information about users
- **Trojan horse:** A seemingly legitimate program, but with a malicious intent.
- **Logic Bomb:** This type of virus is capable of being triggered at a precise moment.

Password Attack

- **Brute Force (Dictionary Attack):** Try by using common passwords.
- **Phishing:** Sending fraudulent but attractive emails
- **Man-in-the-middle:** Sniffing the communication packets
- **Credential Stuffing**: Signed in from another device
- **Keyloggers**: Sniffing while typing the password
- **Failed Attempt**
- **Social Engineering**

SQL injection



Example of SQL injection

SQL Injection.

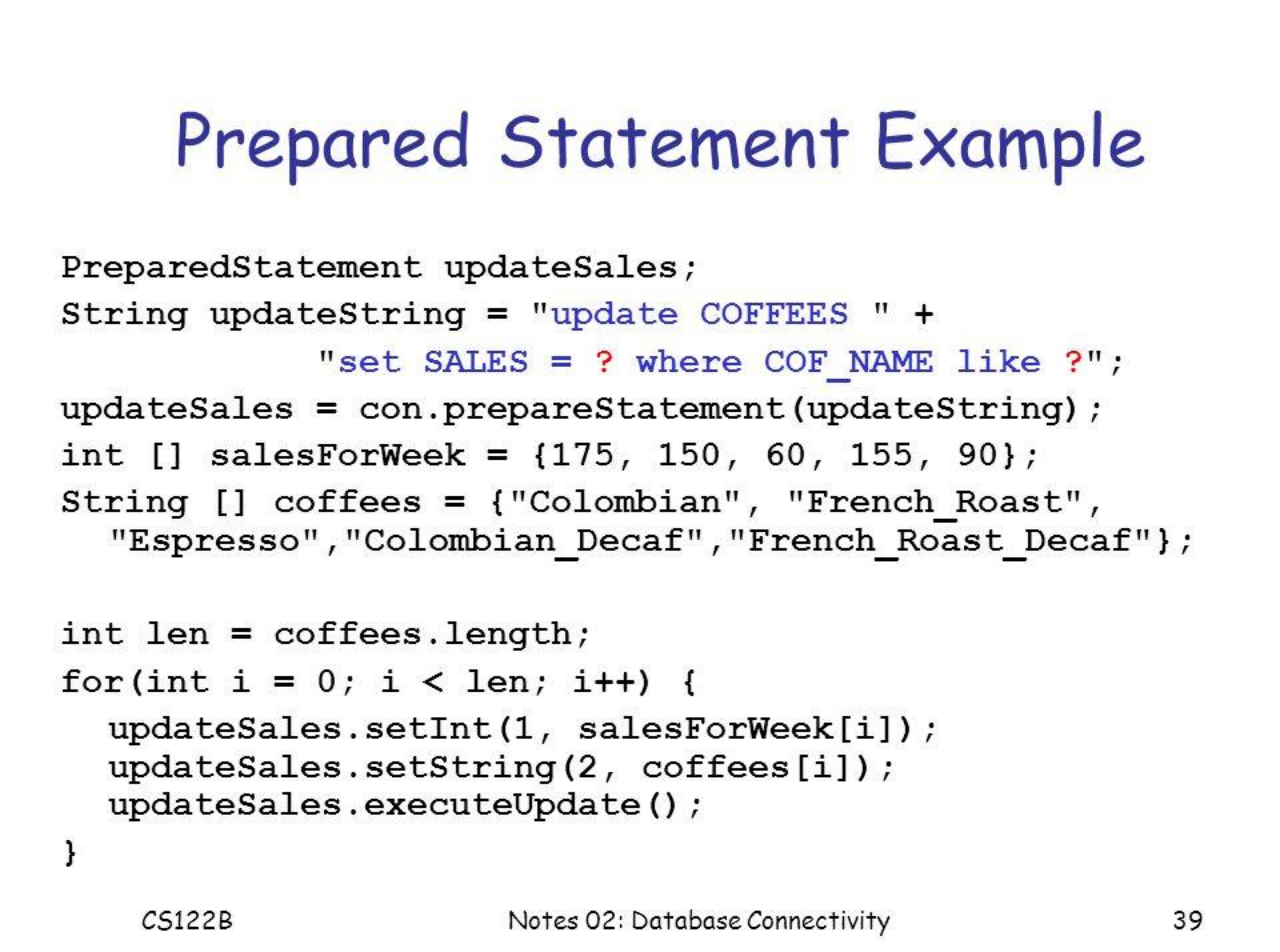
User-Id :	srinivas
Password :	mypassword

```
select * from Users where user_id= ' srinivas '
                    and password = ' mypassword '
```

User-Id :	' OR 1= 1; /*
Password :	*/--

```
select * from Users where user_id= '' OR 1 = 1; /* '
                    and password = ' */-- '
```

9lessons.blogspot.com



Prepared Statement Example

```
PreparedStatement updateSales;
String updateString = "update COFFEES " +
            "set SALES = ? where COF_NAME like ?";
updateSales = con.prepareStatement(updateString);
int [] salesForWeek = {175, 150, 60, 155, 90};
String [] coffees = {"Colombian", "French_Roast",
  "Espresso","Colombian_Decaf","French_Roast_Decaf"};

int len = coffees.length;
for(int i = 0; i < len; i++) {
  updateSales.setInt(1, salesForWeek[i]);
  updateSales.setString(2, coffees[i]);
  updateSales.executeUpdate();
}
```

Zero day Attack

- **Zero day** means very recent.
- Software developers are always looking out for vulnerabilities to "patch" – that is, develop a solution that they release in a new update.
- Sometimes hackers spot the vulnerability before the software developers do.
- Then they can write and implement a code to take advantage of it.
- Also sometimes users ignore the recent software updates. All these cause zero day attack.

https://www.cvedetails.com/vulnerability-list/vendor_id-26/product_id-32238/Microsoft-Windows-10.html

Other Attacks

- Dos and Ddos
- Cross-site Scripting
- Birthday Attack
- Eavesdropping

Good & Bad Practices

Excuses from practitioners

- “It's not exploitable.”
- *“No one will do that!”*
- *“Why would anyone do that?”*
- *“We've never been attacked.”*
- *“We're secure, we use cryptography.”*
- *“We're secure, we use a firewall.”*
- “We've reviewed the code, and there are no security bugs.”

Good practices:

- **Fields length** checking
- **Validate** input not only on client-side but on **server-side** environment too;
- Use "**preparedStatement**()" in **Java** and similar functions in other languages to avoid ***SQL Injection*** attacks;
- Possibly use **high level virtualized languages such as Java, C#**;
- Low level languages like C and C++ are more exposed to buffer overflow exploits;
- Always deploy your application in the authentic Appstore.

Good practices

- Use **Public Key Cryptography (AES, DES)** to do effective encryption.
- Encrypt passwords with **PGP**, **GnuPG**, **RSA** or other encryption tools; store them in a secure place;
- Facilitate password strength and prompt to change old passwords.
- **Identification & Authentication** have to be done over **encrypted channels (e.g., HTTPS)**
- Do security testing

Thank You