THE UNIVERSITY OF MELBOURNE

Database Security and Backups

Database Systems & Information Modelling
INFO90002

Week 10 – Database security

Dr Tanya Linden
David Eccles

This Lecture Discusses

- Technical safeguards
  - Types of access control
  - Firewall
- Data Safeguards
  - Encryption
  - Backups. Types of backups
- Reducing risk of data loss

Technical Safeguards – Access Control

1. Access Control Policy (People and Procedures)
   - A high level set of rules to grant, revoke and or deny access to the database
2. Access Control Model (Procedure)
   - The model is the formalised policy of the rules
     - Identifying the role of the person/process attempting to perform a task
     - Checking the permissions of the role
     - Granting or denying requested access
3. Access Control Mechanism (Data and Technical)
   - The mechanism is the means to enforce the policy

Technical Safeguards - Access Control System

Policy content example: Only HR & Payroll managers should be able see the salary of employees. Only HR staff should be able to see an employee's date of birth

Model:

- Everybody who is a HR Manager or Payroll Manager will be able to see the employee table, all other job roles such as HR Staff, or Payroll Clerk will have to access the employee table via a view which omits the salary information. Only HR staff should be able to see an employee's date of birth

Mechanism: Role Based AC

```
CREATE VIEW V_EMPLOYEE
AS
SELECT employeeid, firstname, lastname,
departmentid, bossid,dateofbirth
FROM employee;

GRANT select on V_EMPLOYEE to HRstaff;
```

Staff	Role	EmpID
Helen Jones	HR Manager	56
Van Nguyen	Payroll Manager	23
Cathy Bates	HR Staff	101
Wolfgang Tuck	Payroll Clerk	27

Access Control Types

Types of Access Control

- Discretionary Access Control
- Mandatory Access Control
- Role Based Access Control

Access Control Types

Discretionary Access Control - DAC

- Based on the identity of the user requesting access
- Explicitly states which user (subject) can perform which action (action) on which resource (object)
- DAC mechanism controls are defined by user identification with supplied credentials during authentication
- Data owners (or any users authorised to control data) can define access permissions for specific users or groups of users (e.g. sharing objects on Google Drive, OneDrive)

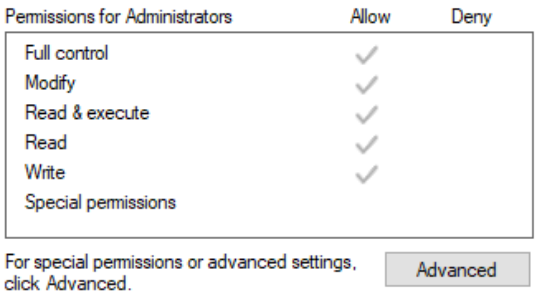
- Authorisation is triple:
  - Subject (User)
  - Object (Table)
  - Action (DML, DDL, DQL)
- Types of DAC
  - Authorisation Table
  - Access Control List
  - Capability (Owner determines access rights to the objects they own)

important point: owner can delegate some of their rights on other people. instead of accessing object directly

Discretionary Access Control

DAC is used in UNIX, Windows, Linux, and many other network operating systems.

A user may give access to their file or directory to other users or groups. The user decides on the type of control (read/write/execute…)



DBA (or sysadmin) could perform the following types of discretionary access control

1. Control who can create databases
2. Prevent unauthorised users from registering user-defined routines

Example:

```
## Grants for lindent@% ##

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP,
RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES,
        SHOW DATABASES,
```

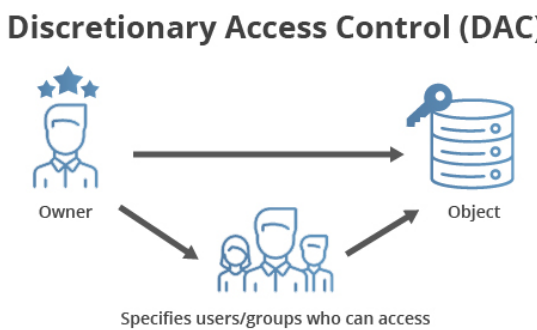


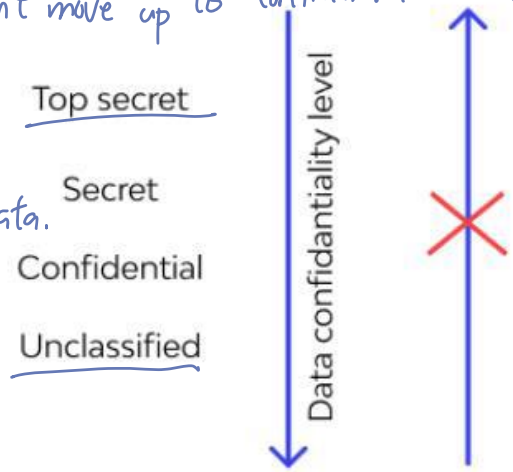
Diagram source
https://www.ekransystem.com/en/blog/mac-vs-dac

Access Control: Other types

Mandatory access control (MAC) is a model of access control where the operating system provides users with access based on data confidentiality and user clearance levels.

In this model, access is granted on a need to know basis.

- Single Sign On
  - an authentication scheme that allows a user to log in with a single ID to any of several related, yet independent, software systems.
- Active Directory



Examples (military systems)

- An individual data owner does not decide who has a top-secret clearance
- The owner of an object cannot change the classification of an object from top-secret to secret

Role Based Access Control (RBAC)

- Unix / Linux / Mac Groups

MySQL Access Control

System administration. Two possibilities:

- Privileges per user (see example)
- Associate certain privileges with certain roles, and assign roles to users

Example of privileges per user:

```
## Grants for lindent@% ##
GRANT SELECT, INSERT, UPDATE, DELETE,
CREATE, DROP, RELOAD, SHUTDOWN,
PROCESS, FILE, REFERENCES, INDEX, ALTER,
SHOW DATABASES,
```

Grant privileges at:

- Global Level (all databases)
- Database Level (on one database level (like access to one database only))
- Table Level
- Column Level

(Optional reading http://www.br8dba.com/tag/how-to-grant-user-privileges-at-the-column-level-on-mysql/)

Technical Safeguard - Firewalls

protection layer between internet and your local network or your computer

In big company, firewall is a seperate server protecting like a God.

In our laptop, it's just a piece of software.

Protective layer between your LAN and the WAN / Internet

Software or dedicated hardware-software unit selectively blocks or allows data packets

All network traffic is quarantined and authenticated

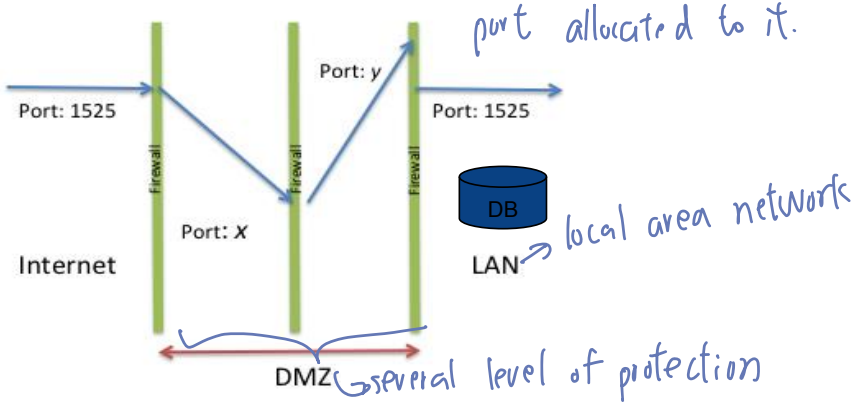
Often several layers and types of firewall

DMZ – Demilitarized Zone

MySQL – Port 3306

what happens is that depending on what kind of messages are coming, they are allowed only on certain port.

And every popular type of product has particular port allocated to it.



local area networks

several level of protection

Data Safeguard - Encryption

When normal readable text becomes jumbled into sth unreadable
But the actual text is not changed, it's just a masked, its representation has changed.

In What app : encrypts your data from point to point, which means it's encrypted in transit so no one can grab it on its way to another mobile

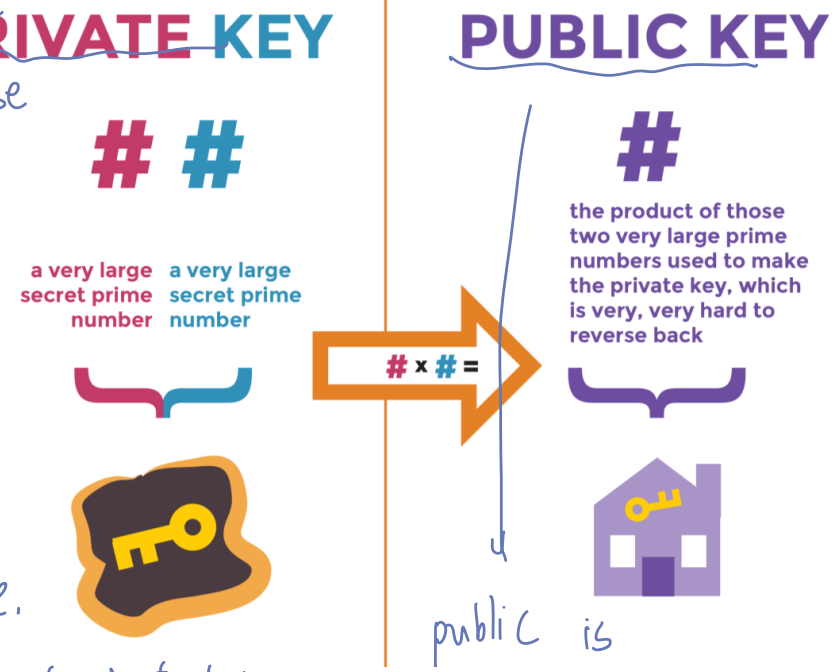
but we don't which algorithm they use

Encryption turns "clear text" into "*k4#h2nsk7"

Involves very big prime number calculations used to scramble clear text

- Then "Salting" adding a byte or two to the encrypted string

Encryption *does not* hide data - it masks data



encryption is often done by public key and private key. Public key is available for everyone. Private key can used for decryption and without private key it cannot be decrypted.

public is

Data Safeguard - Encryption

and it's used to encrytec data.

- Public Key and a Private Key
  - Public key is broadcast software doing encryted.
  - Private key is required to unencrypt

And the software on another end knows what is the private key to decryted it.

(A encrypts sensitive information using B's **public key** and sends it across. B can only access that information and decrypt it using their corresponding **private key**)

Encryption

Encryption-at-Rest vs In-Transit

- Encryption-at-Rest - Protect stored, static data (e.g. on disk)
- In-Transit - Protect data moving from one location to another (such as across the Internet, through a private network, or between services)

Types of Data Encryption-at-Rest

- Application-level encryption: The app that modifies or generates data also performs encryption.
- Database encryption: Entire database, or some of its parts such as certain tables, are entirely encrypted.
- File system encryption: Encrypt only selected file systems (or folders within a file system). Anyone can boot up the device with this encryption but accessing the protected file system(s) requires a passphrase.
- Full disk encryption: Converts data on the entire hard drive into a nonsensical form. The only way to boot up the device is to provide a password.

Backups

What and why

Types of Backups:

- Physical vs. Logical
- Online vs. Offline
- Full vs. Incremental
- Onsite vs. Offsite

Data Safeguards – What is a Backup?

precious things.

A backup is a copy of your data

- there are several types of backup

If data becomes corrupted or deleted or held to ransom it can be restored from the backup copy

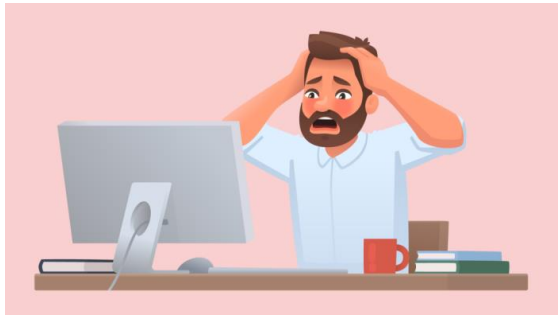
A backup and recovery strategy is needed

- To plan how data is backed up
- To plan how it will be recovered

Backups protect data from …

Human error

- e.g. accidental drop or delete
- example: https://7news.com.au/business/banking/nab-blames-human-error-for-personal-data-breach-affecting-13000-customers-c-368105



Hardware or software malfunction

- bug in application
- hard drive (failure or corruption)
- CPU
- memory

Back-ups protection

Backups could protect against

- malicious activity 惡意的
  - security compromise
    - server, database, application

prevent can use backup

- natural or man-made disasters
  - consider the *scale* of the damages

(physical disaster)

Backups help comply with

- government regulation
  - historical archiving rules
  - Metadata collection (AUS)
  - Privacy Rules

Security

Texas cops lose evidence going back eight years in ransomware attack

We have to get very, very tough on cyber and cyber warfare… and backups?

By Alexander J Martin 27 Jan 2017 at 16:57 36 SHARE ▼

♫ I hacked the sheriff, but I did not hack his deputy ♫

Updated Cockrell Hill, Texas has a population of just over 4,000 souls and a police force that managed to lose eight years of evidence when a departmental server was compromised by ransomware.

In a public statement, the department said the malware had been introduced to the department's systems through email. Specifically, it arrived "from a cloned email address imitating a department issued email address" and after taking root, requested 4 Bitcoin in ransom, worth about $3,600 today, or "nearly $4,000" as the department put it.

Categories of Failure

Failures can be divided into the following categories:

1. Statement failure
   - Syntactically incorrect (misspelt)

```
SELECT employeeid, firstname, lastname, celery
FROM employee;
```

salary

2. User Process failure
   - The process doing the work fails (errors, dies)

ex: sth happened in operating system (crashing on computer)

3. Network failure
   - Network failure between the user and the database

ex: internet server crash (ex: Optus cable broken)

Unable to connect to info90002db.eng.unimelb.edu.au

4. User error (ex: student accidentially dropped the schema)
   - User accidentally drops the rows, table, database

```
-- What does this do?
DROP table employee;
Rollback;
```

drop can not be restored.

5. Memory failure ex: RAM corrupt
   - Memory fails, becomes corrupt

6. Media Failure
   - Disk failure, corruption, deletion

Different types of backup.

Backups – Physical vs. Logical

Physical

(ex: copy assignment files from a hard disc to the cloud drive. It's on the level of operating system. It's not database operation)

1. Raw copies of files and directories
2. Suitable for large databases that need fast recovery
3. Database is preferably offline ("cold" backup) when backup occurs
   - MySQL Enterprise automatically handles *file* locking, so database is not wholly offline
4. Backup = exact copies of the database directories and files
5. Backup should include logs
6. Backup is only portable to machines with a similar configuration

To restore

- shut down DBMS
- copy backup over current structure on disk

We hope to have the latest version, it's prefer that you do the backup when your database is offline.

If don't maybe I will lose the data, which we have done after save.

Because if there are interrupted operations, the log will show it that certain transaction didn't run to the end.

When you are restore the database, the system will know that it needs to update certain table, according to the latest log.

shut down the system temporarily and back up everything and later on if things go wrong and you need to restore it. basically you need backups only for restore.

Logical

1. Backup completed through SQL queries
2. Slower than physical
   - SQL SELECTs rather than OS copy
3. Output is larger than physical
4. Doesn't include log or config files
5. Machine independent
6. Server is available during the backup

In MySQL can use the backup using

- Mysqldump
- SELECT ... INTO OUTFILE

To restore

- use mysqlimport, or LOAD DATA INFILE within the MySQL client

Logical backups are just SQL queries.
It's a group of select quries to read data from the disc and save it in the format so that later on these statements. If can be used in create statement to create schema and insert statments to insert data back to tables.

It doesn't include log or config files, which means whatever data you copied. if some transaction was broken in the middle, you don't have a log file to know that you have damage the database consistency.

ex: files that you get for the assignment2 to install the database. This is like a backup of some database. And everyone can get their own version to install it.

not in exam

server is available during the backup:

=> No major conflict. Because it's select statment. They're reading tables even when other transaction may also reading data from the same table.

physical backup, you can done both online or offline. But logical backup you can only done for online. Because if you database shut down. You cannot do any transact statment.

Backups – Offline vs. Online

Online (LIVE) or HOT

1. Backups occur when the database is "live"
2. Clients don't realise a backup is in progress → so it is possible for the client that it will slow down the database or because backup locking some tables
3. Need to have appropriate locking to ensure integrity of data columns or row. A client might be told that the operating system is temporarily unavailable.
4. No downtime or outage
5. Physical and Logical backups

It can be used in physical and logical backup.

Offline (Shutdown) COLD

1. Backups occur when the database is stopped
2. Avoid the risk of copying data that might be in the process of being updated (application in the hospital cannot use this one, these one cannot have downtime)
3. Simpler to perform
4. Offline backup is preferable, but not available in all situations,
   e.g. applications without downtime
5. Cannot be interrupted by a virus or hacker or power surge
6. Physical backups only

what is downtime and outage mean in the database online backup

In the context of database management, particularly when dealing with online backups, the terms "downtime" and "outage" have specific implications:

Downtime

Downtime refers to the period during which a database (or a system) is unavailable or not operational. This can be due to planned maintenance, such as software upgrades, hardware replacements, or other scheduled tasks that require the database to be temporarily offline or in a non-operational state. Downtime can also occur unexpectedly due to system failures, crashes, or other unforeseen incidents that cause the database services to be disrupted.

In the scenario of an online backup, the goal is typically to minimize downtime so that the impact on users and services that depend on the database is reduced. Online backup strategies are designed to allow backups to be performed without taking the database offline, thereby reducing downtime.

Outage

Outage generally refers to an unplanned interruption or failure in the database service, which results in the system being unavailable for use. Outages can be caused by a variety of issues including hardware failures, software bugs, power failures, and more. Outages are a form of unexpected downtime and are often critical issues that require immediate attention to restore service and ensure data integrity.

Online Backup Context

When it comes to online backups, these terms become crucial because the ability to backup data without affecting the availability of the database is a significant advantage. Here's how they relate to online backups:

- **Minimizing Downtime**: Online backups are designed to run while the database is still operational, which ideally minimizes downtime. This means that routine backup operations should not cause the database to become unavailable to its users. The main challenge in online backups is to ensure that the backup process does not overly degrade the performance of the database while it is still serving user requests.

- **Preventing Outages**: Effective backup strategies are critical for preventing outages by ensuring that there is always a recent backup available to restore from in case of a failure. If an outage occurs, a recent online backup can significantly reduce the recovery time by providing a recent restore point, thus bringing the system back online more quickly and with minimal data loss.

In summary, while online backups aim to prevent outages and reduce downtime, the terms themselves are generally used to describe undesirable states where the database is either not operational or unexpectedly unavailable. Online backup techniques strive to keep the system operational during backups and to provide a fallback in case an unexpected failure leads to an outage.

backing up the whole database. physical backup is a full backup. because you're copying the whole files that are there.

Backups – Full vs. Incremental

Full backup

A full backup is where the complete database is backed up

- Physical (online or offline)
- Logical (online)

It includes everything you need to get the database operational in the event of a failure but it's time consuming

You do full backup once in a while. (eg: you full backup once a week(). And then you do incremental backup for serval times a day.

Incremental Backup using logical backup. → to make note of change since the last backup.

Only the changes since last backup are backed up

For most databases this means only backup log files

To restore: (not the ransom one)

1. stop the database,
2. copy backed up log files to disk,
3. start the database,
4. tell it to redo the log files

(ransom one) So, if sth goes wrong, you want to reformat, you use physical backup to restore and the you run logical backup commands one after another to restore to the latest possible state of the database.

Don't need to reformat everything to restore. first you stop the database. And you copy log files to restore only

the last few operations. if it wasn't sth major that stopped your database.

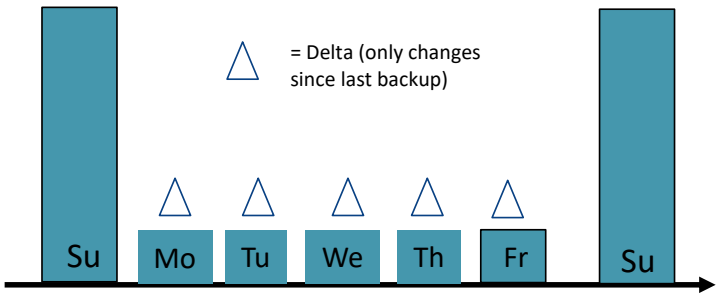
Backup Strategy

ex: doing physical backup on Sunday, Every Sunday operation are down. And other backups on every other day.

Backup strategy is usually a combination of full and incremental backups

For example:

- weekly full backup
- weekday incremental backup



Conduct backups when database load is low

If you replicate the database, use the mirror database for backups to negate any performance concerns with the main database

TEST your backup before you **NEED** your backup!

Important things: not just to create a backup, but actually test that files you are created is readable. So backup need to be tested.

Offsite Backup

Use it → Because if you keep your backup files on the same server. If the server crashes, the backup will die with the main database. The point is that the backup should be separately.

Motivation: hackers could still potentially get into your backups if they're connected to your network

Offsite means company backups are not stored in the organisation's building

Enables disaster recovery and business continuity

- Must be at remote site (e.g. ASIC require 100 km away)

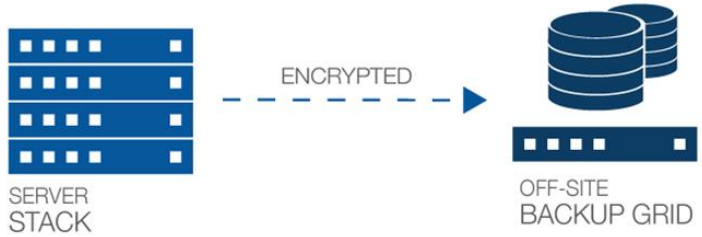
Backup tapes transported to underground vault

- NAB Knox City vault (15 feet silicon wall)

Remote mirror database maintained via replication

- Telstra Data Centres (Melbourne and Sydney)

Backup to Cloud

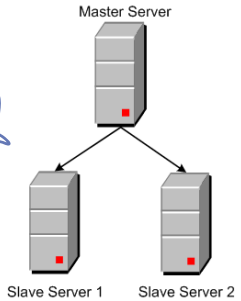


THE UNIVERSITY OF MELBOURNE

Other ways to reduce risk of data loss

Other ways to reduce risk of data loss

Server replication



Problem	Protection?
accidental drop or delete	data loss!
server failure	protected
security compromise	limited protection

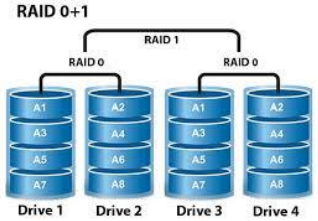
if you delete sth, it'll update on all other server. So, it does not protect this one.

Server cluster

Problem	Protection?
accidental drop or delete	data loss!
server failure	protected
security compromise	limited protection

RAID

It's a group of cheap disc



Problem	Protection?
accidental drop or delete	data loss!
server failure	data may be lost!
security compromise	all data compromised!

if this one fail, it fails everywhere.

Database Hardening - checklist

DB Physical Hardening (not easy to enter the room)

- harder to get to the server room

Firewalls for DB Servers And there also have logical protection, like separate firewalls. for database server.

Database Software; App / Web Server and App Code

- regularly patched, constant checking for vulnerabilities

Client Workstations / Browsers

- least privilege rule

Admin SU (Super User) accounts, permissions and passwords

User roles, permissions, passwords and reporting

Change Management

Auditing

Backup and Recovery

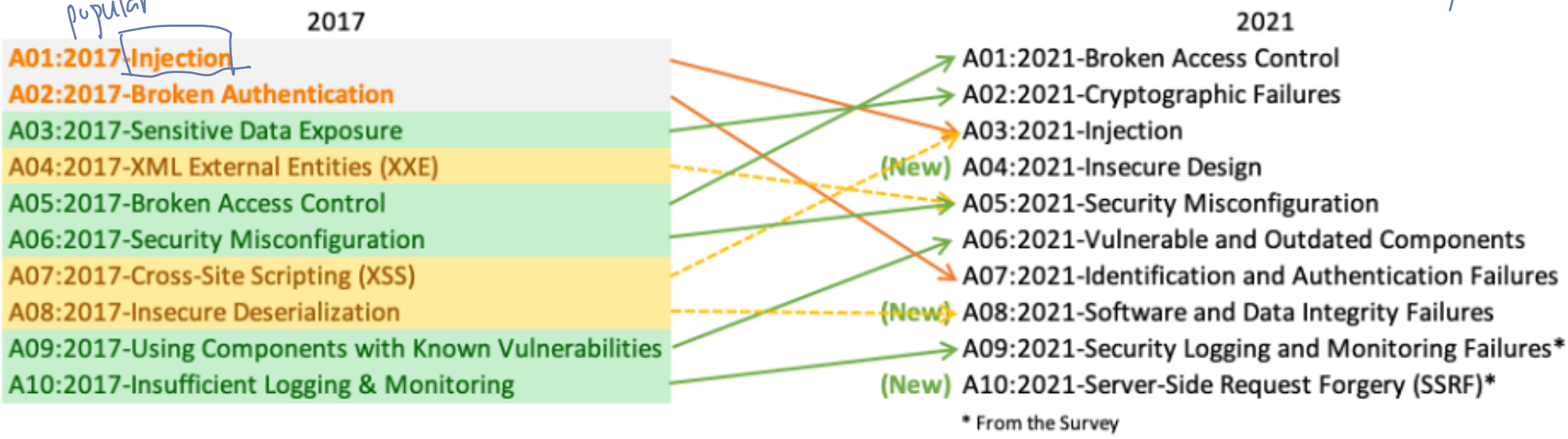
Web security

there are different vulnerabilities

OWASP top ten web app vulnerabilities

How to get into that database on the web. Web server and what can go wrong.

popular



https://owasp.org/www-project-top-ten/

OWASP = *Open Web Application Security Project*

SQL injection

This is a type of attack which exploit how we provide input to the database

SQL Injection attacks

- a technique used to exploit web applications that use *user input within database queries*
- malicious code is entered into a data entry field in such a way that it becomes part of SQL commands that are run against the database

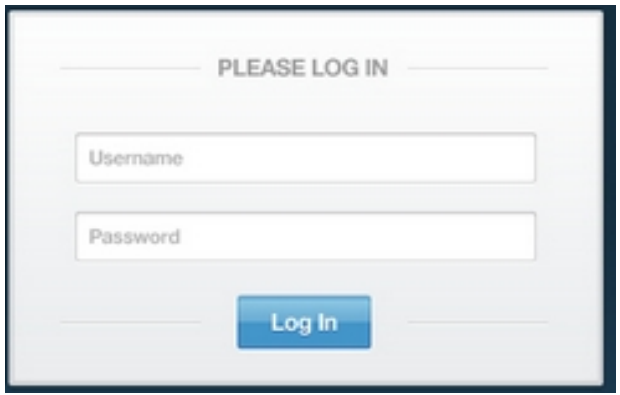
Hacker's goals

- Understand the database structure (table names, column names, etc.) → and they will able to add another row to the database
- Access and view data in important tables (e.g. customer accounts details)
- Add data to tables (e.g. new accounts or usernames and passwords)
- How to prevent:
  - sanitize user inputs (e.g. block OR, UNION, and similar in SQL statements with user input)
  - pass inputs as parameters to a stored procedure, rather than directly building the SQL string in the code

SQL injection

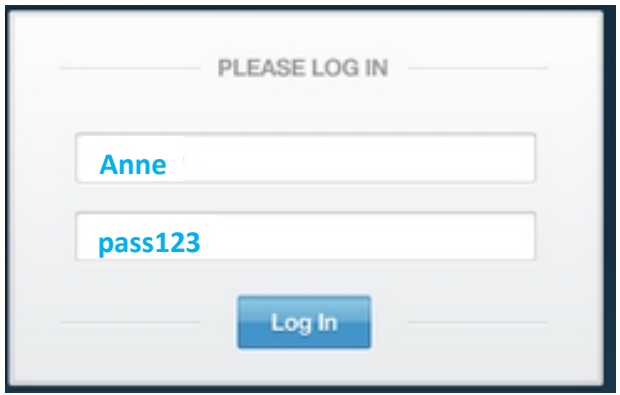
User inputs are used to form an SQL statement

Login



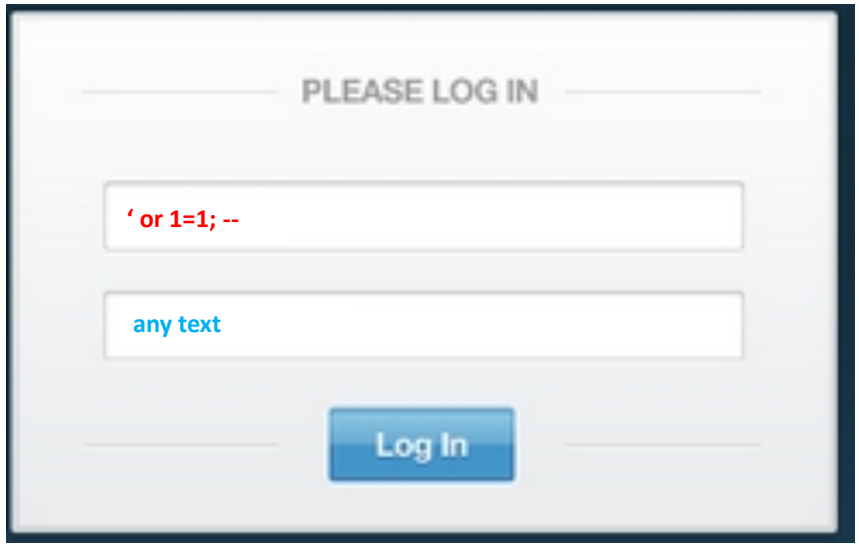

```
SELECT *
FROM User
WHERE username = ' @name '
and password = ' @pw ';
```

Programmer wants:




```
SELECT *
FROM User
WHERE username = 'Anne'
and password = 'pass123';
```

SQL injection: malicious input



Text entered in @name string now

- closes the string
- adds a condition that is always true
- ends the SQL statement
- begins a comment with '- - ' to neutralise the rest of the SQL

```
SELECT *
FROM User
WHERE username = '' or 1=1; --
```

ignore everything afterwards. so the password equals sth else will be completely commented out.

is always True

the where username will be useless

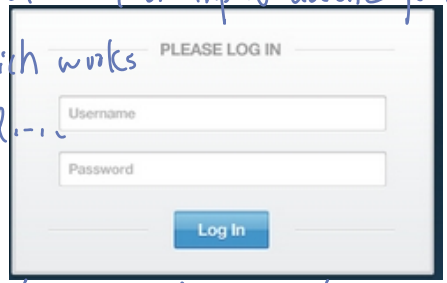
See an interactive demo
https://www.hacksplaining.com/exercises/sql-injection

Empty string username - not going to happen, OR 1=1 - always true

SQL injection: prevention

What are the major defense against SQL injections?
→ The companys use parameterized statement and stored procedures.

which means that the input is validated before it is sent to the execution. Your input doesn't go directly into that SQL statement, which works for search for item. keyword...



Primary defences:

1. • Prepared Statements (parameterised queries)
2. • Stored Procedures
   – (both mean SQL is no longer 'dynamic')
- i.e. "escape" all user input
  – turns SQL special characters like ' ; -- into ordinary characters

Also remove some type of special character, like single quote, double quote or especially commenting out dash dash, converting them into ordinary characters rather than part of SQL statment.

Additional defences:

3. • Principle of Least Privilege
   – don't give application accounts DBA privileges
4. • White List input validation
   – check input is from a list of acceptable values

Additional: ① making sure that application have least privilege. (strict limitation for what you can get in the database)

② check user input before sending it into the database. for wether it's acceptable or not.

Interesting read:
https://www.pcworld.com/article/485770/lulzsec_anonymous_hacks_were_avoidable_report_says.html

Source: OWASP SQL Injection Prevention Cheat Sheet

What's examinable

- Access control
- Technical safeguards
- Data safeguards
- Types of back-ups
- Reducing risk of data loss
- SQL injections

THE UNIVERSITY OF MELBOURNE

Thank you