UNIVERSITY
OF TRENTO - Italy

eit Digital
MASTER SCHOOL

Security Engineering
Fall 2015

Lecture 15 – DB Security

Fabio Massacci

(Most material courtesy of D. Gollmann)

Lecture Outline

- ***DB security***
  - General threats
  - Relational databases countermeasures
  - Computation on encrypted data
- ***Statistical DB security***
  - Inferential threats
  - Statistical databases countermeasures
- ***New challenges***

Database Security: Statistics

- ***621 out of 47000+ security incidents lead to data disclosures and to 44 million+ compromised records***
- ***Main threats***
  - Malware 40%
  - Hacking 52%
  - Social 29%
  - Misuse 13%
  - Physical 35%
  - Error 2%
  - Environmental 0%

Database Security

- ***A database is a collection of data***
- ***DBMS organizes the data and gives users the means to retrieve information***
- ***Database Security:***
  - protection of sensitive data and mechanisms that allow users to retrieve information in a controlled manner
  - Provide controlled, protected access to the contents of a database as well preserve the integrity, consistency and overall quality of the data

Design Requirements

- ***Precision:***
  - protect sensitive information while revealing as much non-sensitive information as possible
- ***Internal consistency:***
  - the entries in the database obey some prescribed rules
    - E.g., stock levels cannot fall below zero.
- ***External consistency:***
  - The entries in the database are correct
    - E.g., stock levels given in the database match stock levels in the warehouse
  - DBMS alone cannot keep the database in a consistent state → need organizational controls!
  - This property is also called accuracy

Top Ten Database Security Threats

- ***(lack of) organzational measures***
  1. Excessive and Unused Privileges
  2. Abuse Privileges
  8. Unmanaged Sensitive Data
  10. Limited Security Expertise and Education
- ***(presence of) software problems***
  3. SQL Injection
  4. Malware
- ***5. A bit of both***
  5. Weak Audit Trail
  6. Storage Media Exposure
  7. Exploitation of (unpatched) Vulnerabilities
  9. Denial of Service

T1: Excessive and Unused Privileges

- ***When someone is granted database privileges that exceed the requirements of their job function, these privileges can be abused***
- ***Access control mechanisms for job roles have not been well defined or maintained***
  - users may be granted generic or default access privileges that far exceed their specific job requirements
- ***Example***
  - a bank employee whose job requires the ability to change only account holder contact information may take advantage of excessive database privileges and increase the account balance of a colleague's savings account

T2: Abuse Privileges

- ***Users will abuse legitimate database privileges for unauthorized purposes***
- ***Example***
  - Consider an internal healthcare application used to view individual patient records via a custom Web interface
  - The Web application normally limits users to viewing an individual patient's healthcare history
  - However, a rogue user might be able to circumvent these restrictions and copy electronic healthcare records on his laptop

T8: Unmanaged Sensitive Data

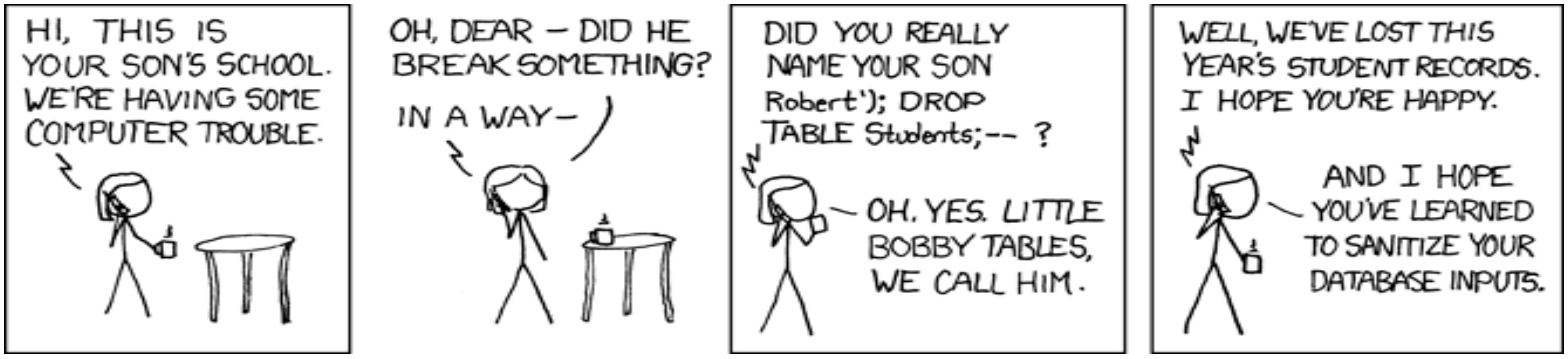
- ***Many companies struggle to maintain an accurate inventory of their databases***
  - Forgotten databases may contain sensitive information
  - New databases can emerge
- ***Sensitive data in these databases will be exposed to threats if the required controls and permissions are not implemented***
  - You cannot control what you don't "know" it exists

T10: Limited Security Expertise and Education

- ***Internal security controls are not keeping pace with data growth and many organizations are ill-equipped to deal with a security breach***
- ***Due to the lack of expertise required to implement security controls, policies, and training***
- ***According to PWC's 2012 Information Security Breaches Survey***
  - 75% of the organizations surveyed experienced staff-related breaches when a security policy was poorly understood
  - 54% of small businesses did not have a program for educating their staff about security risks

T3: SQL Injection

- ***Already discussed for application security***
  - SQL injection involves inserting (or "injecting") unauthorized or malicious database statements into a vulnerable SQL data channel such as a Web application or stored procedure
  - If these injected statements are executed by the database, critical data stores can be viewed, copied, and altered
- ***Normally this happens by exploiting the polyglottism of the web application on top of the DB***
  - http://xkcd.com/327



T4: Malware

- ***Cybercriminals, state-sponsored hackers, and spies use advanced attacks to penetrate organizations***
  - spear phishing emails and malware
- ***Legitimate users become a conduit for these groups to access networks and sensitive data***
  - Users are unaware that malware has infected their device

T5: Weak Audit Trail

- ***Failure to collect detailed audit records of database activity***
  - Organizations with weak (or sometimes non-existent) database audit mechanisms will increasingly find that they are at odds with industry and government regulatory requirements
- ***Many enterprises will turn to native audit tools provided by their database vendors or rely on ad-hoc and manual solutions***
  - These approaches do not record details necessary to support auditing, attack detection, and forensics.
- ***Finally, users with administrative access to the database, either legitimately or maliciously obtained, can turn off native database auditing to hide fraudulent activity***
  - Audit duties should ideally be separate from both database administrators and the database server platform to ensure strong separation of duties policies

T6: Storage Media Exposure

- ***Backup storage media is often completely unprotected from attack***
  - Numerous security breaches have involved the theft of database backup disks and tapes
- ***Failure to audit and monitor the activities of administrators who have low-level access to sensitive information can put your data at risk***
  - Taking the appropriate measures to protect backup copies of sensitive data
  - Monitor your most highly privileged users is not only a data security best practice, but also mandated by many regulations

T7: Exploitation of Vulnerabilities

- ***It is common to find vulnerable and un-patched databases, or discover databases that still have default accounts and configuration parameters***
  - Attackers know how to exploit these vulnerabilities to launch attacks against your organization
- ***Maintenance is hard***
  - Organizations often struggle to stay on-top of maintaining database configurations even when patches are available
  - It generally takes organizations months to patch databases once a patch is available
  - Sometimes licensing is the issue, sometimes interoperability with legacy software

T9: Denial of Service

- ***DoS conditions can be created via many techniques***
- ***Most common technique***
  - overload server resources such as memory and CPU by flooding the network with database queries that ultimately cause the server to crash
- ***Motivations behind DoS attacks***
  - often linked to extortion scams in which a remote attacker will repeatedly crash servers until the victim meets their demands
- ***Software or organizational decision***
  - Not enough “power” to meet excess demand

Relational Databases

- ***A relational database is a database that is perceived by its users as a collection of tables***
- ***A relation R is a subset of $D_1 \times \cdots \times D_n$ where $D_1, \ldots, D_n$ are the domains on n attributes.***
- ***The elements in the relation are n-tuples $(v_1, \ldots, v_n)$ with $v_i \in D_i$; the value of the i-th attribute has to be an element from D_i.***
- ***Elements in a tuple are often called fields***
- ***A special null value indicates that a field does not contain any value***

Example

Name	Day	Flight	Status
Alice	Mon	GR123	private
Bob	Mon	YL011	business
Bob	Wed	BX201	
Carol	Tue	BX201	business
Alice	Thu	FL9700	business

Flight	Destination	Departs	Days
GR123	THU	7: 55	1- - 4 - -
YL011	ATL	8:10	12345 - 7
BX201	SLA	9: 20	1 - 3 - 5 - -
FL9700	SLA	14: 00	- 2 – 4 – 6 -
GR127	THU	14: 55	-2 - 5-

SQL

- ***Structured Query Language (SQL):***
  - standard language for describing how information in a relational database can be retrieved and updated.
- ***SQL operations:***
  - SELECT: retrieves data from a relation.
  - UPDATE: update fields in a relation.
  - DELETE: deletes tuples from a relation.
  - INSERT: adds tuples to a relation.

Examples

- ***SELECT Name, Status***

  FROM Diary

  WHERE Day = 'Mon'

- ***UPDATE Diary***

  SET Status = private

  WHERE Day = 'Sun'

- ***DELETE FROM Diary***

  WHERE Name = 'Alice'

- ***INSERT INTO Flights (Flight, Destination, Days)***

  VALUES ('GR005', 'GOH', '12-45-')

Types of Relations

- ***Base relations (real relations):***
  - named, autonomous relations; exist in their own right, are not derived from other relations,
  - have 'their own' stored data
- ***Views:***
  - named, derived relations, defined in terms of other named relations;
  - no stored data of their own.
- ***Snapshots:***
  - named, derived relations, defined in terms of other named relations;
  - have stored data of their own.
- ***Query results:***
  - may or may not have a name;
  - no persistent existence in the database per se

Database Keys

- ***Tuples in a relation must be uniquely identifiable***
- ***A primary key K of a relation R must be:***
  - Unique: at any time, no tuples of R have the same value for K;
  - Minimal: if K is composite, no component of K can be omitted without destroying uniqueness.
- ***Every relation must have a primary key***
- ***A primary key of one relation that is an attribute in some other relation is a foreign key in that relation***

Integrity Rules

- ***Entity Integrity Rule:***
  - no component of primary key of a base relation is allowed to accept nulls
- ***Referential Integrity Rule:***
  - DB must not contain unmatched foreign key values
- ***Application specific integrity rules:***
  - Field checks: to prevent errors on data entry
  - Scope checks
  - Consistency checks

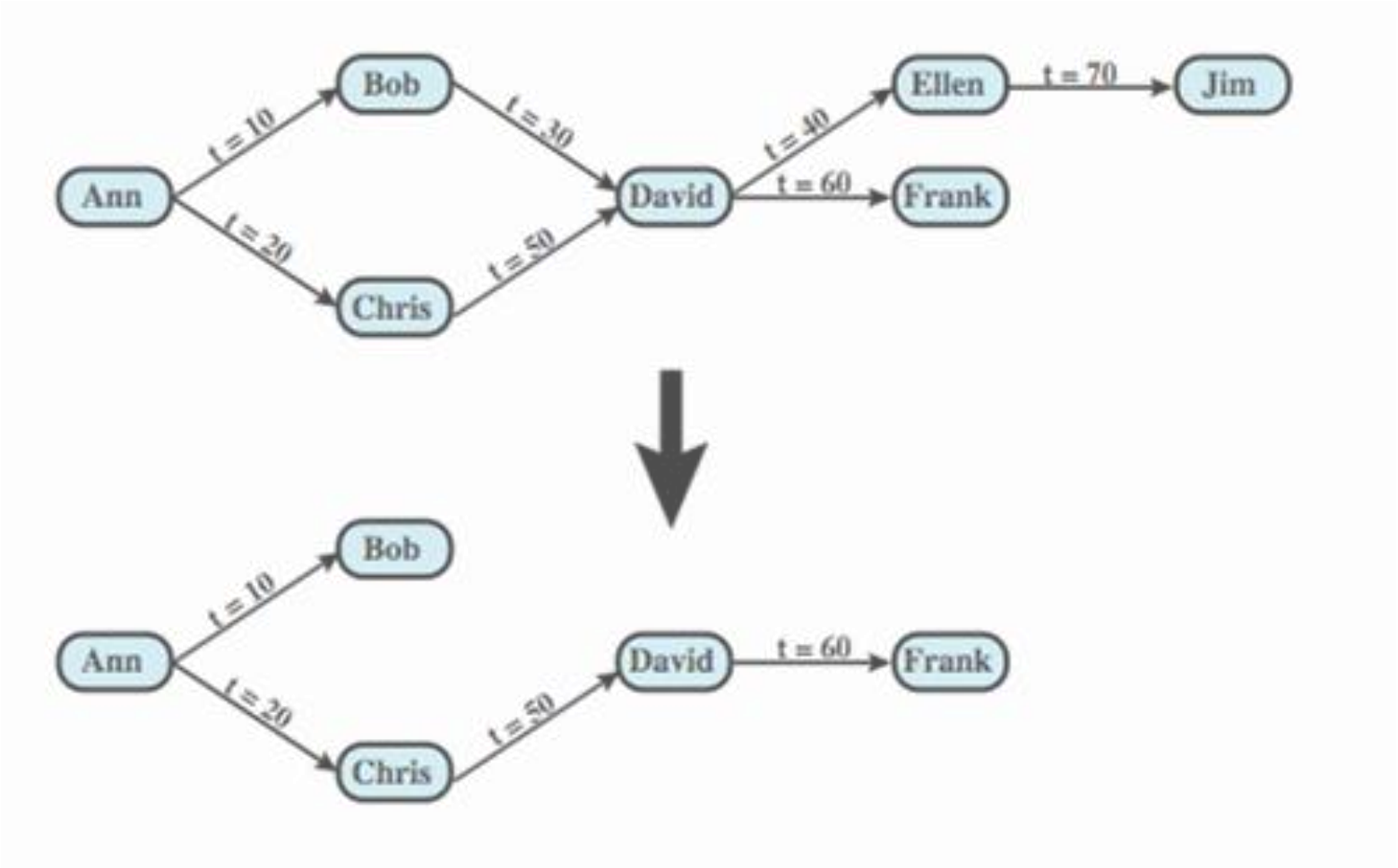
Access Control

- ***Discretionary access control using privileges and views, based on:***
  - users: authenticated during logon;
  - actions: include SELECT, UPDATE, DELETE, and INSERT;
  - objects: tables, views, columns (attributes) of tables and views;
- ***Users invoke actions on objects***
- ***DBMS decides whether to permit the requested action***
- ***When an object is created, it is assigned an owner; initially only the owner has access to the object; other users have to be issued with a privilege:***
  - (grantor, grantee, object, action, grantable)

Granting & Revoking Privileges

- ***Privileges managed with GRANT and REVOKE***
  - GRANT SELECT, UPDATE (Day, Flight)
  - ON TABLE Diary
  - TO Art, Zoe
- ***Selective revocation of privileges:***
  - REVOKE UPDATE
  - ON TABLE Diary
  - FROM Art
- ***Right to delegate privileges given through GRANT option:***
  - GRANT SELECT
  - ON TABLE Diary
  - TO Art
  - WITH GRANT OPTION

Granting and Revoking Privileges



Role-Based Access Control

- ***Role-based access control work well for DBMS***
  - eases admin burden, improves security
- ***categories of database users:***
  - application owner
  - end user
  - administrator
- ***DB RBAC must manage roles and their users***
  - cf. RBAC on Microsoft's SQL Server
- ***It can be implemented using GRANT and REVOKE***

Row Level Access Control

- ***Views: derived relations, created by***
  - CREATE VIEW view_name [(column [, column] ...)]
  - AS subquery
  - [WITH CHECK OPTION]
- ***Many security policies better expressed by privileges on views than by privileges on base relations***
- ***Access conditions described through subquery in the view definition***
  - CREATE VIEW business_trips AS
  - SELECT * FROM Diary
  - WHERE Status = `business'
  - WITH CHECK OPTION;

Advantages

- ***Views are flexible and allow access control policies to be defined at a level of description that is close to the application requirements***
- ***Views can enforce context-dependent and data-dependent security policies.***
- ***Views can implement controlled invocation***
- ***Secure views can replace security labels***
- ***Data can be easily reclassified***

Examples

- ***Application-level access control***

```
CREATE VIEW Top_of_the_Class AS
  SELECT *
  FROM Students
  WHERE Grade >(SE
                  FROM Students
                 WHERE Name =

CREATE VIEW My_Journeys AS
       SELECT * FROM Diary
        WHERE Customer = current_user());
```

displays students whose grade average is higher than that of the person using the view

display journeys booked by the customer using the view

CHECK Option

- ***INSERT and UPDATE can interfere with view-based access control***
- ***Views may not be updatable because they do not contain the information that is needed to maintain the integrity of the corresponding base relation***
  - E.g., a view that does not contain the primary key of an underlying base relation cannot be used for updates
- ***Blind writes:***
  - updates that overwrite an existing entry
  - For views defined WITH CHECK OPTION, UPDATE and INSERT can only write entries to the database that meet the definition of the view
- ***Blind writes possible if CHECK option is omitted***

CHECK Option: Example

- **UPDATE business_trips AS**
- ***CREATE VIEW business_trips AS***

  SET Status = ‘private’

  WHERE Name = ‘Alice’ AND Day =‘Thu’

  SELECT * FROM Diary

  WHERE Status = ‘business’

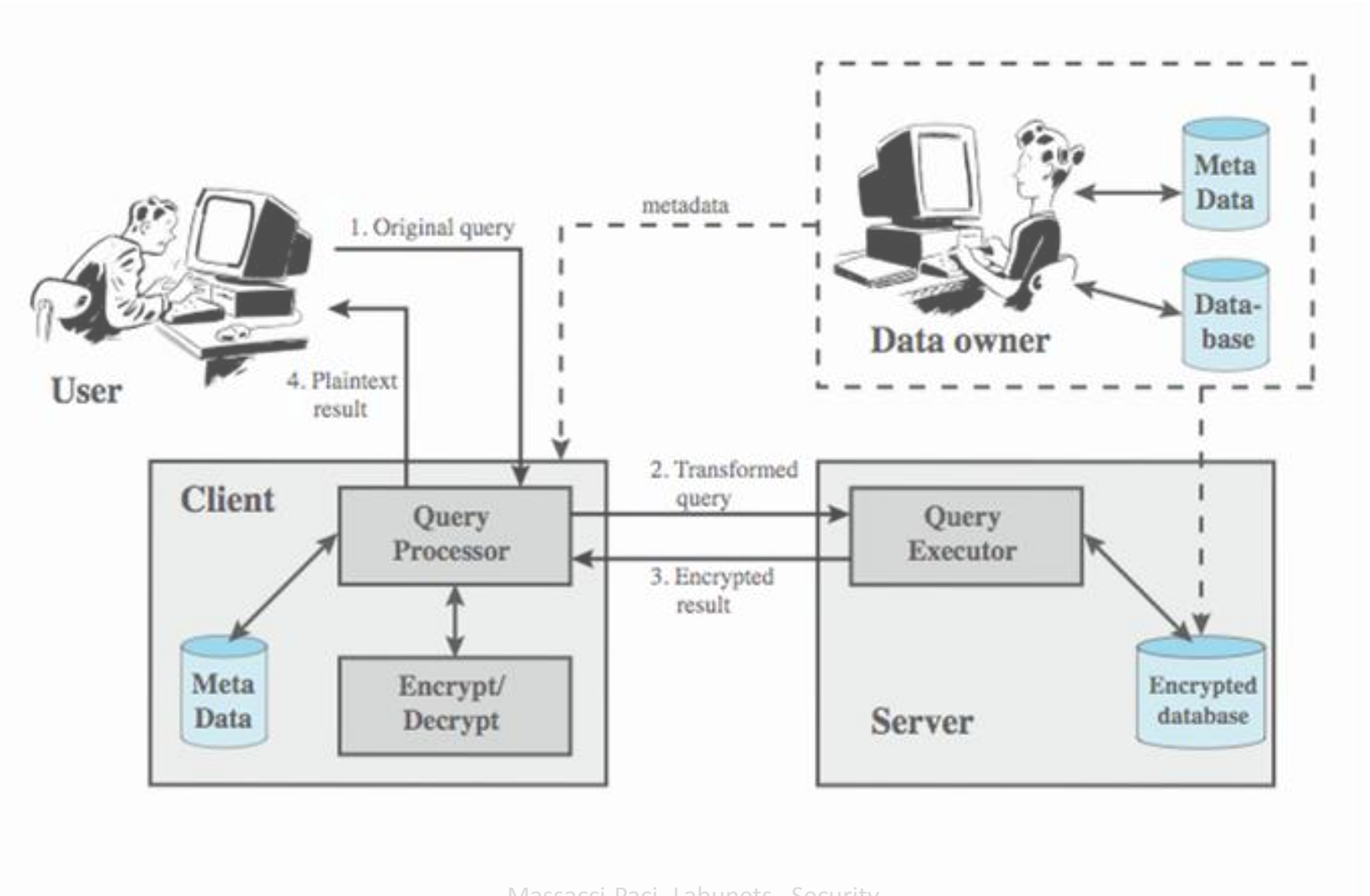
  WITH CHECK OPTION

Name	Day	Flight	Status
Bob	Mon	YL011	business
Carol	Tue	BX201	business
Alice	Thu	FL9700	business

Database Encryption

- ***Databases are a valuable information resource***
  - protected by multiple layers of security: firewalls, authentication, O/S access control systems, DB access control systems, and database encryption
- ***Encryption strategies***
  - entire database - very inflexible and inefficient
  - individual fields - simple but inflexible
  - records (rows) or columns (attributes) - best
    - also need attribute indexes to help data retrieval
- ***Issues***
  - Key Management
  - Query Execution

Database Encryption



Dtabase encryption functionalities

- ***Fragments of SQL can be captured by different kind of encryptions***
  - Order-preserving encryption preserves the order of plaintexts, i.e.
    - $x \leq y \Rightarrow E_{OP}(x) \leq E_{OP}(y)$
  - Deterministic encryption preserves the equality of plaintexts, i.e.
    - $x=y \Rightarrow E_{DET}(x) = E_{DET}(y)$
  - In (additively) homomorphic encryption multiplication of ciphertexts (modulo a key-dependent constant) maps to addition of the plaintexts,
    - $D_{HOM}(E_{HOM}(x)*E_{HOM}(y)) = x+y$
- ***Unfortunately not all types of encryptions can be combined***
  - SELECT x FROM T GROUP BY y HAVING SUM(z) > 100
  - Requires complex constructions data is layered in onions of encryption
- ***Not all encryption schemes equally secure (why?)***
  - OP ≤ DET ≤ HOM
  - However not everything needs to be encrypted with less secure scheme (but makes DB very dependent on type of query, and cannot change later)
- ***Performance is still painful***
  - SAP performance Data (GIS 2014)
    - Exact Search – 1.2x, Equi-Join 1.5x, Group&Sum – 11.7x, OrderbySum – 15.4x
- ***Harder with "standard" access control on the DB for multiple users***
  - Different users have access to different fragments of the DB which therefore must use different keys
  - SAP implementation by F. Kershbaum is +37% over single encrypted users

Statistical Database Security

- ***Statistical database***
  - information retrieved by means of statistical (aggregate) queries on attributes (columns) of a table.
- ***Aggregate functions in SQL***
  - COUNT: the number of values in a column,
  - SUM: the sum of the values in a column,
  - AVG: the average of the values in a column,
  - MAX: the largest value in a column,
  - MIN: the smallest value in a column.
- ***Query predicate of a statistical query***
  - specifies the tuples used for computing the aggregate value
- ***Query set***
  - tuples matching the query predicate

Threats

- ***Key difference with normal databases: user is authorized to ask stuff but not too much***
- ***Aggregation***
  - sensitivity level of an aggregate computed over a group of values may differ from the sensitivity levels of the individual elements;
    - e.g., an aggregate may be sensitive information derived from a collection of less sensitive business data
- ***Inference***
  - Derivation of sensitive information from non-sensitive data
  - Direct Attack:
    - aggregate computed over a small sample so that information about individual data items is leaked
  - Indirect Attack:
    - combine information relating to several aggregates
  - Tracker Attack:
    - a particularly effective type of indirect attack

Direct Attack

Name	Sex	Program	Units	Grade Ave.
Alma	F	MBA	8	63
Bill	M	CS	15	58
Carol	F	CS	16	70
Don	M	MIS	22	75
Errol	M	CS	8	66
Flora	F	MIS	16	81
Gala	F	MBA	23	68
Homer	M	CS	7	50
Igor	M	MIS	21	70

Direct Attack

- *Assume we know Carol is a female CS student*

```
Q1 : SELECT COUNT(*)
     FROM Students
     WHERE Sex = 'F' AND
```

Returns count 1

```
Q2 : SELECT AVG(Grade Ave.)
     FROM Stud
     WHERE Se                    m = 'CS'
```

Returns 70: average for a single student

Tracker Attack

```
Q3 : SELECT COUNT(*)
     FROM   Students
     WHERE  Programme = 'CS'

Q4 : SELECT COUNT(*)
     FROM   Students
     WHERE  Programme = 'CS' AND Sex = 'M'

Q5 : SELECT AVG(Grade Ave.)
     FROM   Students
     WHERE  Program = 'CS'

Q6 : SELECT AVG(Grade Ave.)
     FROM   Students
     WHERE  Pro
```

Returns count 4

Returns count 3

Returns average 61

Returns average 58

Carol's grade average:
$4 \cdot 61 - 3 \cdot 58 = 70$

General Attacks

- ***Individual tracker for a given tuple***
  - Query R that allows to derive information about our tuple r
- ***General tracker***
  - Query T used to find the answer to any inadmissible query.
- ***How to combine them?***
  - T chosen so that the query set and its complement are large enough for the query to be permitted
  - Make two queries to the database with the predicates $R \vee T$ and $R \vee \neg T$;
    - the target r is the only tuple used by both queries
- ***'Add' the two results and 'subtract' the result of a query over the entire database***
- ***Only the target is left***

General Tracker

```
Q7 : SELECT SUM(Units)
     FROM   Students
     WHERE  Name = 'Carol' OR Program = 'MIS'
```

Returns sum 75

```
Q8 : SELECT SUM(Units)
     FROM   Students
     WHERE  Name = 'Carol' OR NOT (Program = 'MIS')
```

Returns sum 77

```
Q9 : SELECT SUM(Units)
     FROM   Students
```

Returns sum 136

Carol has passed
(75 + 77) - 136 = 16 units

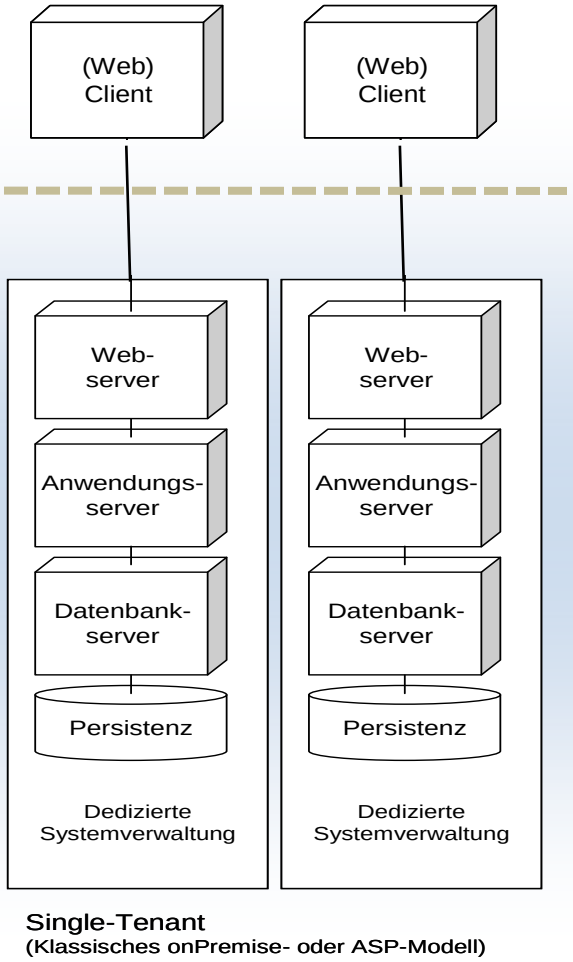
Countermeasures

- ***Suppress obviously sensitive information***
- ***Disguise the data***
  - Randomly modify entries in the database so that an individual query will give a wrong result although the statistical queries still would be correct
    - R. Agrawal, R. Srikant: Privacy-Preserving Data Mining. SIGMOD 2000.
  - Add small random perturbations to query result so that the value returned is close to the real value but not quite correct
  - Drawback: reduced precision and usability
- ***Better design of the database schema***
- ***Track what the user knows***
  - user actions recorded in an audit log, a query analysis checks for suspicious sequences of queries

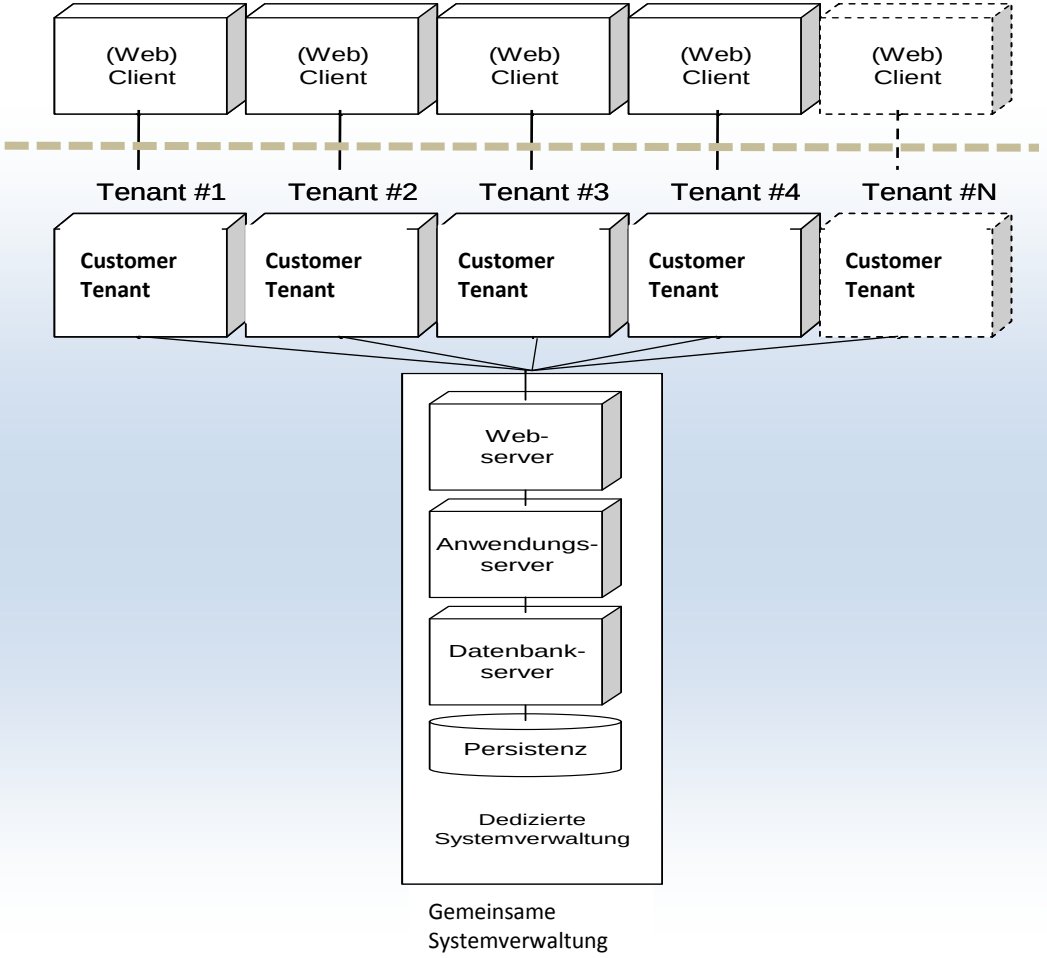
Life is complicated

- ***Protecting access to services requires protecting all layers***
  - Web server
  - Application Server
  - DBMS Server
  - OS System + Hw Infrastructure
- ***They used to be isolated into a silo for different customers***
  - The “Apache” model
  - Things change with cloud models

From ASP to Multi-Tenancy



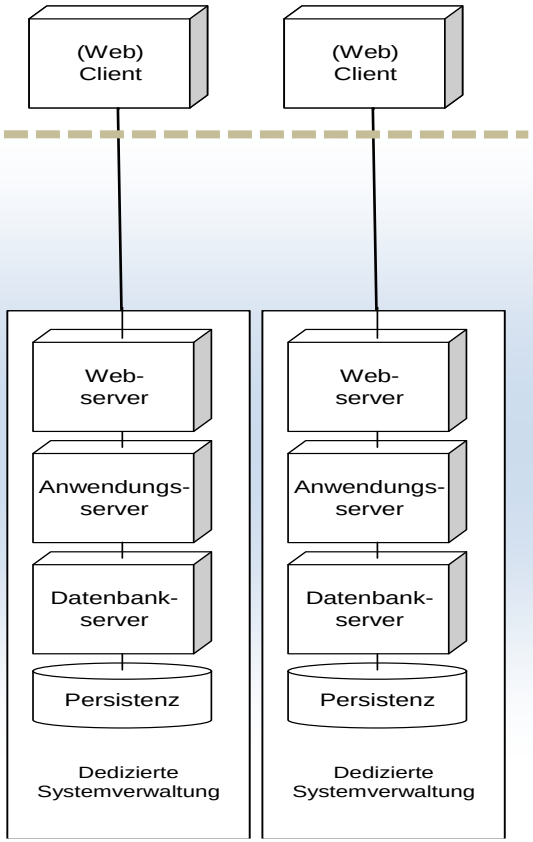
Single Tenancy
(classical on-premise or ASP model)



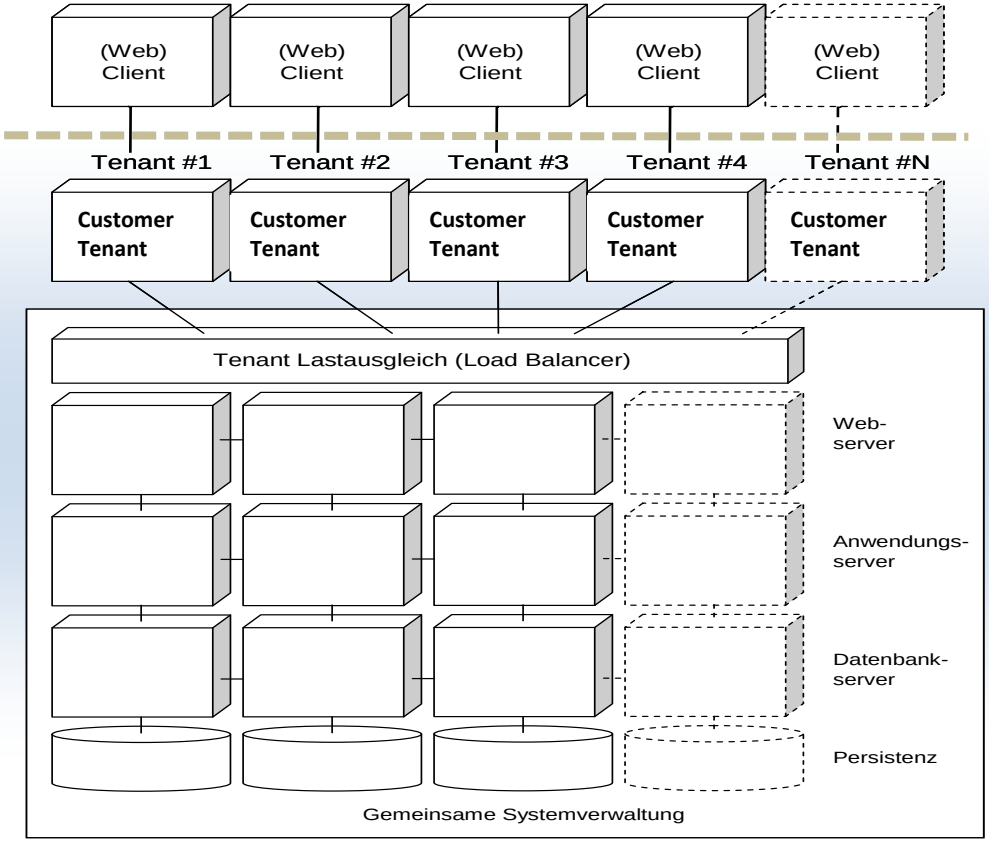
Multi Tenancy

Source: SAP

Efficient & Sclabale Multi-Tenancy



Single Tenancy
(classical on-premise or ASP model)



Multi Tenancy

Source: SAP

Additional readings

- ***Chapter 9. D. Gollmann – Computer Security***
- ***Chapter 5 W. Stallings & L. Brown – Computer Security - Principles and Practices***
- ***Animated Database Courseware***
  - http://adbc.kennesaw.edu/
- ***Top 10 Database Security Threats***
  - http://www.imperva.com/docs/WP_TopTen_Database_Threats.pdf
- ***Searching on Encrypted data***
  - http://www.fkerschbaum.org/seed.html
- ***Search Google for DataCenter Security***
  - http://www.youtube.com/watch?v=1SCZzgfdTBo