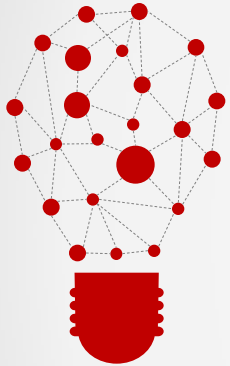
《数据安全》

【第一章　数据库安全】

翁正秋

本章要点



- **目的与要求：**
  - 了解数据库安全涉及的范畴，包括数据库安全性控制、数据库脱敏、数据库漏洞扫描、数据库防火墙等内容；
  - 掌握数据库安全控制的常用方法与技术。
- **重点与难点：**
  - 数据库安全性控制（重点）
  - 数据库脱敏（重难点）
  - 数据库漏洞扫描（重难点）
  - 数据库防火墙（重点）

资料整理
火龙果软件 荐

本章要点

1.1 案例

1.2 数据库安全性控制

1.3 数据库脱敏

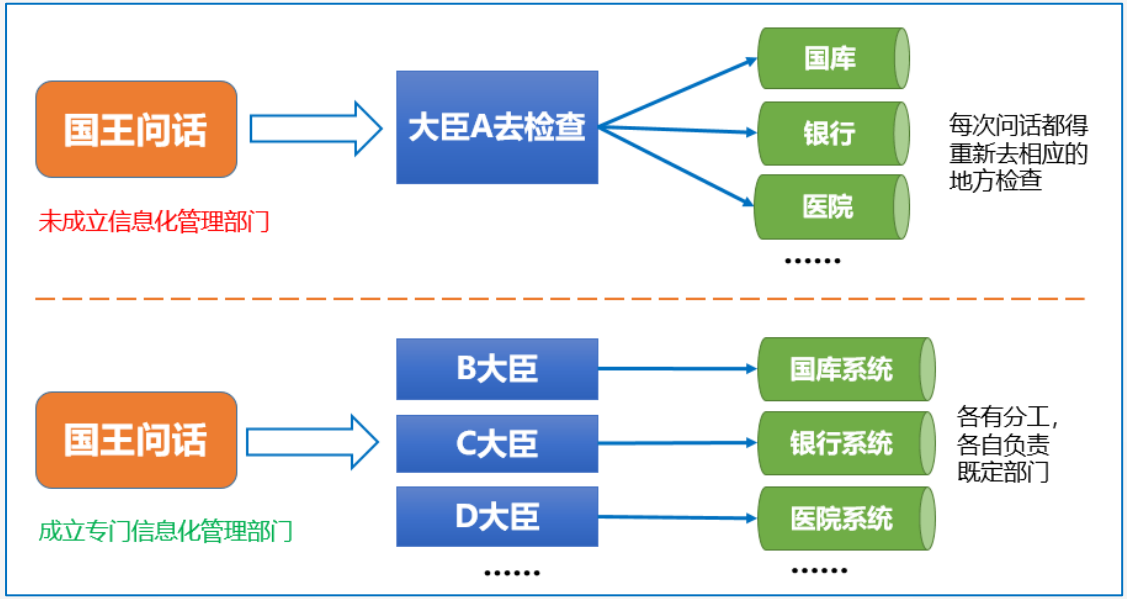
1.4 数据库漏洞扫描

1.5 数据库防火墙

资料整理
火龙果软件
荐

案例 【案例1：基于视图的访问控制】

通篇案例导入：X国



资料整理
火龙果软件
荐

案例解析

假如把案例中的系统看成一个大型的数据库，那么案例中的故事我们可以用数据库中的视图机制更好地去理解。

1 视图的使用

案例中的大臣BCD，都拥有不同的视图View_B、View_C、View_D，不同的大臣查看到的内容是不同的，这就是利用视图进行数据安全访问的好处。

视图只是**定义了一个查询**，视图中的数据是从**基表**中获取，这些数据在视图被引用时**动态的生成**。由于视图基于数据库中的其他对象，因此一个视图只需占用数据字典中保存其定义的空间，而**无需额外的存储空间**，视图也会随着基表的变化而变化。

资料整理
火龙果软件 荐

2 视图的优点

优点1：使用视图可以隐藏查询的复杂性

比如，要查询用户在银行中的信用卡数量，
已知用户表User，信用卡表CreditCard。
一般写法如下：

```
SELECT a.dno, count(*)  As  CreditCard_Num;
FROM User a, CreditCard c
WHERE a.dno=c.dno
GROUP BY a.dno;
```

采取视图的方式

前提是已创建了基于上面查询的视图my_view

```
SELECT dname, CreditCard_Num
FROM my_view
```

通过my_view能够将复杂的查询简化成简单的查询语句，让终端用户或者应用程序使用查询非常方便。

资料整理
火龙果软件 荐

案例 【案例1：基于视图的访问控制】 案例解析

2 视图的优点

优点2：视图可以加强安全性

假设国王只想要用户大臣C查看CreditCard表中的用户姓名UName和信用卡号CardNo两个列。

如果用下面的语句对大臣C进行查看select权限的授权：

GRANT select ON CreditCard TO 大臣C;

大臣C将会看到CreditCard表的全部内容

只看到两列

```
CREATE VIEW C_CreditCard
AS SELECT UName,CardNo
FROM CreditCard;
GRANT select ON C_CreditCard TO 大臣C;
```

只看到自己的

假设想要每个人都可以查询CreditCard表，但是只能看到用户自己的记录

```
CREATE VIEW my_CreditCard
AS SELECT * FROM CreditCard
WHERE UName=USER;
GRANT select ON my_CreditCard TO public;
```

资料整理
火龙果软件
存

案例 【案例2：数据库数据推理与数据营销】

案例描述：

以X国银行系统消费数据为例，有一天，一个用户接到银行客服的电话，直接开门见山，向用户推出本月信用卡额度临时调整业务，可以调整到1万的额度，有效期限本月，并提供很低的利息进行下个月的分期还款，问是否选择调整。

该用户目前的**信用额度为5000元，月初时已经用了4000元的额度，本月还剩20多天**，而日常大部分的生活消费都是用**信用卡支付**的，并且往往都是**月底**进行大部分生活支出，而该用户的**工资水平并不高**，如果不提供分期还款的话，提高额度也同样会带来下个月全额还款的压力。很显然该用户调整额度的需求是非常迫切的。

那么为什么是**调整到1万的额度而且有效期限本月**，并提供**很低的利息**进行**下个月的分期还款**呢？这背后的推理逻辑和营销策略，为何会**如此精准**呢？

资料整理
火龙果软件
荐

案例解析

生活中的推理现象：当看到天空乌云密布、燕子低飞、蚂蚁搬家等现象时，我们会得到一个推理判断：天要下雨了。

1 数据库推理与营销

解析1：用户数据积累

从用户第一天办理信用卡开始，用户的数据就进入银行数据库，其中的数据大致包含了：**姓名、所在城市、区域、年龄、性别、身份证号、电话号、公司、工龄、年薪、信用额度、银行卡号，消费记录、接待客服**等大量的数据。

资料整理
火龙果软件 荐

案例 【案例2：数据库数据推理与数据营销】 案例解析

1 数据库推理与营销

解析2：用户数据挖掘与模型建立

“物以类聚，人以群分”

以本案例的用户为例，特征是：**青年人，刚刚参加工作，工资偏低、日常生活消费为主**等，同时**讲究生活品质，定期购买高端商品，信用额度经常不够用**。

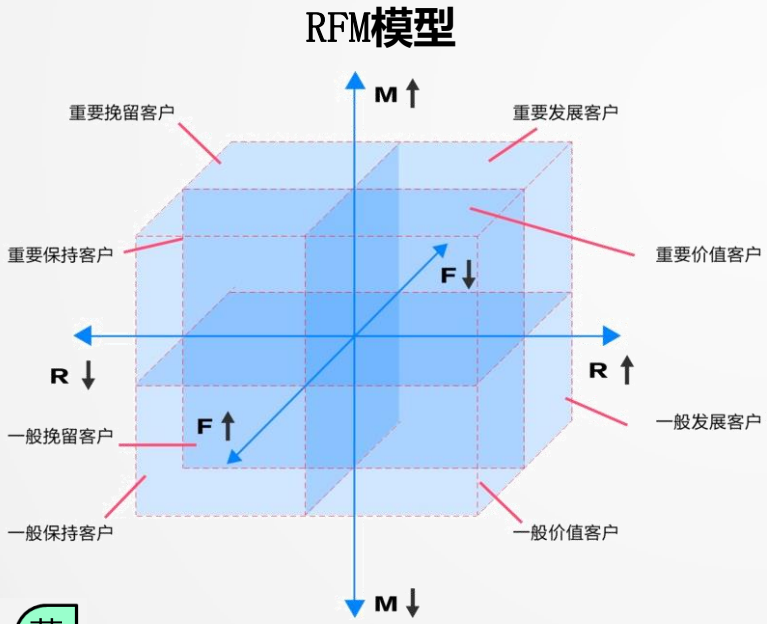
于是这样的用户就极有可能成为信用额度提升、推广分期付款的营销对象。

资料整理
火龙果软件 荐

案例 【案例2：数据库数据推理与数据营销】 案例解析

1 数据库推理与营销

解析3：制订营销方案



（1）最近一次消费(**Recency**)：最近消费，如果面对用户一个月的消费，那么一周作为一个分段，分为4段。

（2）消费频率(**Frenquency**)：消费频率从1次到10次，那么每2次作为一个分段，分为5段。

（3）消费金额(**Monetary**)：消费金额从1元到5000元，那么1000元就是一个等级点。分为5段。

按照上面的3个维度，可以把顾客分成4*5*5 = 100类，对其进行数据分析，然后制定出营销策略。对于那些近期消费用户，频率较高且金额较大的，一定是高端的优质用户，区分出用户后，排好优先级进行营销。

资料整理
火龙果软件 荐

1 数据库推理与营销

解析4：营销执行

细分出精准用户，量身打造的产品，接着就是按照用户名单的优先级别进行多种方式的营销执行。比如对于RFM模型中高端用户，采用电话营销，提高转化率。

解析5：效果评估与模型优化

根据不同渠道影响的用户消费数据，判断各种营销手段的有效性，譬如RFM模型的高端用户，采用A/B测试方式，是发短信的效率高，还是电话营销的效率高?

各种不同的信用卡产品，究竟是否会被持续使用？有无可能提供更贴近用户习惯，又能充分利用资源的产品？

现有的聚类方式是否合理？RFM模型分段是否合理？都需要进行用户行为的后期数据跟踪进行判断优化。

资料整理
火龙果软件
荐

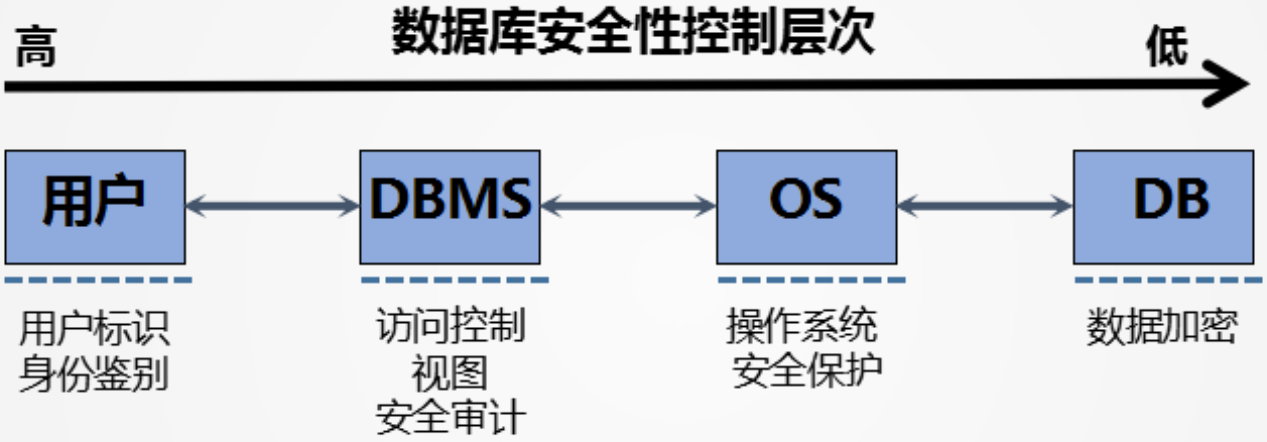
案例 【案例2：数据库数据推理与数据营销】 案例解析

2 数据库安全中的推理问题

一个数据库用户可以通过**一条合法的数据库访问命令**获取数据库中的**低密级数据**，然后再通过**数据库推理**来推断出**高敏感性的数据库信息**，这就是存在于数据库系统中的隐蔽通道，即**数据库推理通道**。

数据库安全中的推理问题是**用户利用数据之间的相互联系窃取其不能直接访问的数据**，从而造成敏感数据泄露的一种安全问题。

资料整理
火龙果软件 荐



安全性控制就是用于实现**数据库各种安全策略**的功能集合，通过这些安全机制来构建安全模型，从而实现确保数据库系统安全的目的。

资料整理
火龙果软件
荐

知识点【用户标识与身份鉴别】 【第一章 数据库安全】

用户标识和身份鉴别的目的是保证只有合法的用户才能存取数据库资源。

用户标识

最简单的方法可以通过用户ID和密码进行标识。作为用户的身份证明要求保证标识的唯一性，每位用户通过这个唯一标志才能进入系统。

通过用户ID和密码的方式虽然简单易行，但是容易被他人窃取。

身份鉴别

身份鉴别是指系统对进入系统的每个用户身份进行鉴别，用于确保用户身份的合法性。

通行字认证

通行字也称为“口令”或“密码”。比较常见的有最小长度限制、次数限定、选择字符、有效期、双通行字和封锁用户系统等

数字证书认证

数字证书是证书认证中心颁发并进行数字签名的数字凭证，它提供了实体身份的鉴别与认证、信息完整性验证、机密性和不可否认性等安全服务。

智能卡认证

智能卡（如有源卡、IC卡或Smart卡等）可以用来验证个人身份，典型智能卡主要由微处理器、存储器、输入输出接口、安全逻辑及运算处理器等组成。

个人特征识别

通过判断被授权用户的个人特征来进行身份识别是一种可信度更高的验证方法，个人特征识别归功于生物统计学的研究成果，即利用个人具有人体唯一的、可靠的、稳定的生物特征（指纹、虹膜、脸部、掌纹等）来实现。

资料整理
火龙果软件
荐

身份鉴别 **【例1-1】以Oracle数据库为例，Oracle数据库提供了两种鉴别方法。**

为了防止非授权用户使用数据库，Oracle数据库提供了两种鉴别方法：操作系统鉴别和Oracle数据库鉴别。

①操作系统鉴别：

Oracle可使用操作系统所维护的信息来鉴别用户。由操作系统鉴定用户具有以下优点：用户可以更方便地链接到Oracle数据库，而不需要指定的用户名和口令；对用户授权的控制集中在操作系统，Oracle不需要存储和管理用户口令。那么同时也带来了以下安全风险：一旦攻击者控制并访问到操作系统，就意味着可以完全操控数据库，那么数据库将面临严重的安全威胁；操作系统级别的访问，不利于多用户在线访问数据库，同时也不利于用户权限的控制。

②Oracle数据库鉴别：

Oracle使用存储在数据库中的信息来对试图连接到数据库的用户进行鉴定。用户申请与数据库建立连接时，首先要输入用户名，系统需要确认用户是否是合法用户；如果是合法用户，则继续要求用户输入口令密码。同时，为了安全起见，用户在终端所输入的口令默认是不显示的。通过口令的核对，数据库系统就可以进一步的核实用户的身份。

资料整理
火龙果软件 荐

访问控制：其目的是确保用户对数据库只能进行经过授权的有关操作。

在存取控制机制中，“客体”指的是被访问的资源对象，“主体”指的是以用户名义进行资源访问的进程、事务等实体。

访问控制机制由两部分组成，分别为定义存取权限和检查存取权限。

定义存取权限

在数据库系统中，为了保证用户只能访问其有权存取的数据，必须预先对每个用户定义存取权限。

检查存取权限

对于通过鉴定获得权限的用户（即确认为合法用户），系统根据其存取权限定义对其各种操作请求进行控制，确保该用户只执行合法操作。

- **自主存取控制（Discretionary Access Control，简称DAC)**
- **强制存取控制（Mandatory Access Control，简称 MAC**
- **基于角色的存取控制（Role-based Access Control，简称 RBAC)**
- **基于属性的权限验证（Attribute-Based Access Control，简称 ABAC)**

资料整理
火龙果软件 荐

自主存取控制DAC

自主存取控制允许用户（如数据库管理员、普通数据用户等）自主对数据库对象的操作权限进行控制。

【例1-2】以SQL Server数据库为例，三种权限状态。

权限分为3种状态：授予、拒绝、撤销，可以使用如下的语句来修改权限的状态。

（1）授予权限（GRANT）：授予权限以执行相关的操作。通过角色，所有该角色的成员继承此权限。

（2）撤销权限（REVOKE）：撤销授予的权限，但不会显式阻止用户或角色执行操作。用户或角色仍然能继承其他角色的GRANT权限。

（3）拒绝权限（DENY）：显式拒绝执行操作的权限，并阻止用户或角色继承权限，该语句优先于其他授予的权限。

资料整理
火龙果软件 荐

自主存取控制DAC

自主存取控制允许用户（如数据库管理员、普通数据用户等）自主对数据库对象的操作权限进行控制。

【例1-3】以SQL Server数据库为例，使用GRANT命令授予角色“po_mag”对”PO”数据库中“物料”表的DELETE、INSERT、UPDATE权限。

```
GRANT  DELETE, INSERT, UPDATE
ON 物料
TO  po_mag
```

资料整理
火龙果软件 荐

自主存取控制DAC

自主存取控制允许用户（如数据库管理员、普通数据用户等）自主对数据库对象的操作权限进行控制。

【例1-4】以SQL Server数据库为例，使用REVOKE语句撤销 “po_mag”角色对“物料”表所拥有的DELETE、INSERT、UPDATE权限。

```
REVOKE  DELETE, INSERT, UPDATE
ON 物料
FROM  po_mag  CASCADE
```

资料整理
火龙果软件 荐

自主存取控制DAC

自主存取控制允许用户（如数据库管理员、普通数据用户等）自主对数据库对象的操作权限进行控制。

【例1-5】以SQL Server数据库为例，在数据库“PO”的“物料”表中执行INSERT操作的权限授予了public角色，然后于拒绝用户guest拥有该项权限。

```
USE PO
GO
GRANT INSERT ON 物料 TO public
GO
DENY INSERT ON 物料 TO guest
GO
```

资料整理
火龙果软件 荐

自主存取控制DAC

自主存取控制的优缺点

优点：能够通过授权机制有效地控制其他用户对敏感数据的存取。

缺点：对权限控制比较分散，不便于管理，可能存在数据的“无意泄露”。

原因：这种机制仅仅通过对数据的存取权限来进行安全控制，而数据本身并无安全性标记。

解决：实施强制存取控制策略。

资料整理
火龙果软件 荐

强制存取控制MAC

每一个数据对象被标以一定的密级；每一个用户也被授予某一个级别的许可证；对于任意一个对象，只有具有合法许可证的用户才可以存取。

强制存取控制的优缺点

优点：用户能否对该对象进行操作取决于双方的权限标识的关系，通常由系统硬性限制。适合机密机构或者其他等级观念强烈的行业。

缺点：不够灵活，不适用于商业服务系统。

原因：这种机制仅仅通过对数据的存取权限来进行安全控制，而数据本身并无安全性标记。

解决：实施强制存取控制策略。

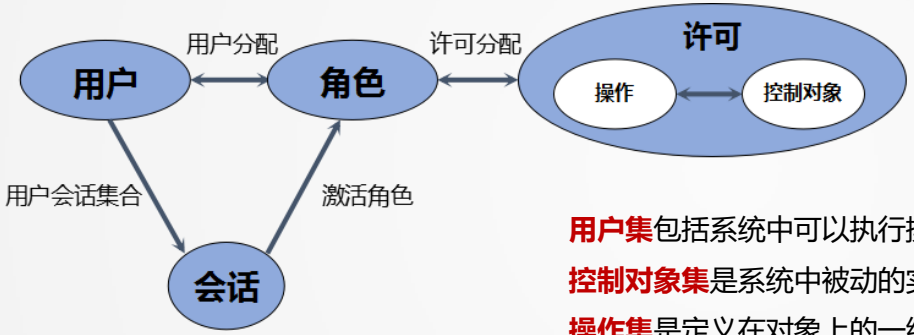
与DAC相比，MAC机制比较严格，强制访问控制（MAC）弥补了自主访问控制（DAC）权限控制过于分散的缺陷。

比如在“谍战”影片中经常看到某个特工在查询机密文件时，系统会提示“权限不足，请提供一级安全许可”等字样，而这样的权限标识往往都是需要硬件进行控制，普通的用户并不具有。

资料整理
火龙果软件 荐

基于角色的存取控制RBAC

RBAC其实就是用户关联角色、角色关联权限，并且，RBAC可以模拟出DAC和MAC的效果。RBAC是目前普遍应用的一种权限设计模型。



用户集包括系统中可以执行操作的用户，是主动的实体；

控制对象集是系统中被动的实体，包含系统需要控制的各类信息；

操作集是定义在对象上的一组操作，对象上的一组操作构成了一个许可；

角色则是RBAC模型的核心，通过用户分配和许可分配使用户与许可关联起来。

RBAC核心模型包含五个基本的静态集合，即用户集（users）、角色集（roles）、许可集（perms）（包括操作集（operators）和控制对象集（objects）），以及一个运行过程中动态维护的集合，即会话集（sessions）

资料整理
火龙果软件 荐

基于角色的存取控制RBAC

RBAC属于策略中立型的存取控制模型，既可以实现自主存取控制策略，又可以实现强制存取控制策略。

【例1-6】以MongoDB数据库为例

① MongoDB便是采用RBAC模型，对数据库的操作都划分成了权限

权限标识	说明
find	具有此权限的用户可以运行所有和查询有关的命令，如：aggregate、checkShardingIndex、count等。
insert	具有权限的用户可以运行所有和新建数据有关的命令：insert和create等。
collStats	具有此权限的用户可以指定database或collection执行collStats命令。
viewRole	具有此权限的用户可以查看指定database的角色信息。
.....	

资料整理
火龙果软件 荐

基于角色的存取控制RBAC

【例1-6】以MongoDB数据库为例

② 基于这些权限，MongoDB提供了一些预定义的角色

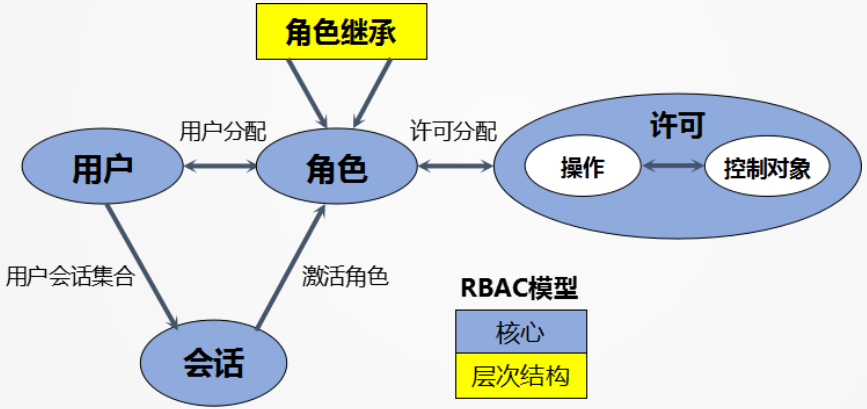
角色	find	insert	collStats	viewRole	……
read	✔		✔		……
readWrite	✔	✔	✔		……
dbAdmin	✔		✔		……
userAdmin				✔	……

资料整理
火龙果软件 荐

基于角色的存取控制RBAC

【例1-6】以MongoDB数据库为例

③ RBAC规范还提供了扩展模式，如角色继承(Hierarchical Role)和职责分离(Separation of Duty)。



角色继承就是指角色可以继承于其他角色，在拥有其他角色权限的同时，还可以关联额外的权限。这种设计可以给角色分组和分层，一定程度简化了权限管理工作

资料整理
火龙果软件 荐

基于角色的存取控制RBAC

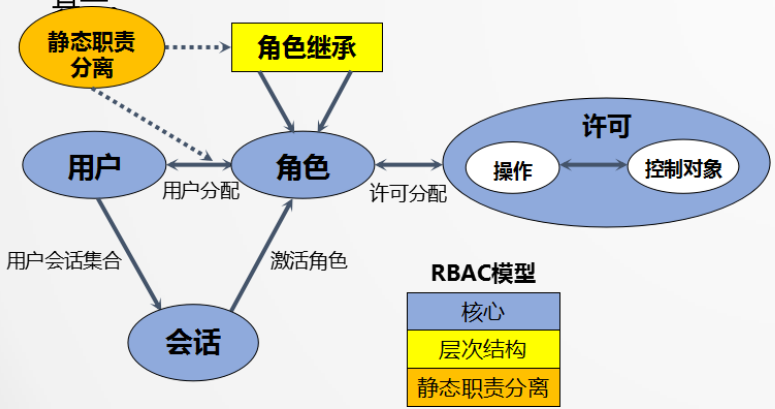
【例1-6】以MongoDB数据库为例

职责分离是为了避免用户拥有过多权限而产生利益冲突，例如一个篮球运动员同时拥有裁判的权限。

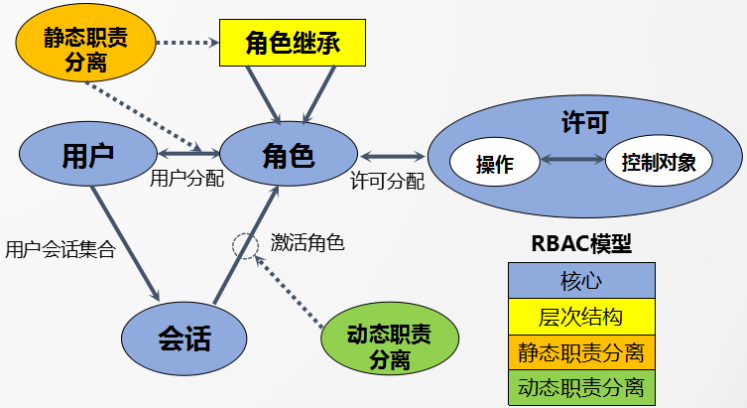
职责分离有以下两种模式。

静态职责分离(Static Separation of Duty)：用户无法同时被赋予有冲突的角色。

动态职责分离(Dynamic Separation of Duty)：用户在一次会话（Session）中不能同时激活自身所拥有的、互相有冲突的角色，只能选择其一



静态职责分离



动态职责分离

资料整理
火龙果软件 荐

知识点【视图定义与查询修改】 【第一章 数据库安全】

视图是一种数据库对象，是从一个或者多个数据表或视图中导出的虚表，视图的结构和数据是对数据表进行查询的结果。

视图的特点

1. 视图能够简化用户的操作，从而简化查询语句；
2. 视图使用户能以多种角度看待同一数据，增加可读性；
3. 视图对重构数据库提供了一定程度的逻辑独立性；
4. 视图能够对机密数据提供安全保护；
5. 适当的利用视图可以更清晰的表达查询；

视图机制使系统具有数据安全性、数据逻辑独立性和操作简便。

资料整理
火龙果软件 荐

创建视图

【例1-7】以SQL Server数据库为例，创建一个视图

```
CREATE  VIEW  v_Item_1
AS
SELECT 物料代码,物料名称
FROM  物料表
GO
```

资料整理
火龙果软件 荐

创建视图

【例1-8】以SQL Server数据库为例，创建一个视图，用于查看物料类别为1的物料的代码和名称，并要求进行修改和插入操作时仍需保证该视图只显示物料类别为1的物料。

```
CREATE  VIEW  v_Item_3 (物料号, 物料名)
AS
SELECT 物料代码,物料名称
FROM 物料表
WHERE 物料类别=1
/*进行修改和插入操作时仍需保证该视图只显示物料类别为1的物料*/
WITH CHECK OPTION
```

资料整理
火龙果软件 荐

创建基于多个基表的视图

【例1-9】以SQL Server数据库为例，创建基于多个基表的视图，用于查看由供应商名为“海蓝电子”供货的采购订单信息。

```
CREATE VIEW v_Item_4
AS
SELECT 采购订单头表.*
FROM 采购订单头表,供应商表
WHERE 采购订单头表.供应商代码=供应商表.供应商代码
And 供应商表.供应商名称= '海蓝电子'
GO
```

资料整理
火龙果软件 荐

创建基于视图的视图

【例1-10】以SQL Server数据库为例，创建基于视图的视图，用于查看由供应商名为“海蓝电子”供货，并且单据类型编号为1的采购订单信息。

```
CREATE  VIEW  v_Item_6
AS
SELECT  *
FROM  v_Item_4   /*在视图v_Item_4的基础上*/
WHERE 单据类型 = 1
GO
```

资料整理
火龙果软件 荐

调用视图

【例1-11】以SQL Server数据库为例，利用视图查询数据，直接调用v_Item_4中的视图，用于查看由供应商名为“海蓝电子”供货的采购订单号为SO001的采购订单信息

```
SELECT *
FROM v_Item_5
WHERE 采购订单号 = 'SO001'
```

资料整理
火龙果软件 荐

更新视图

【例1-12】以SQL Server数据库为例，利用视图插入数据，向视图v_Item_3中插入一个新的物料记录，物料号为1001，物料名为鼠标。

```
Insert into v_Item_3
Values('1001','鼠标')
```

由于视图不一定包括表中的所有字段，所以在插入记录时可能会遇到问题。

视图中那些没有出现的字段无法显式插入数据，假如这些字段不接受系统指派的null值，那么插入操作将失败。

由于视图v_Item_3是用于查看物料类别为1的物料的代码和名称，并要求进行修改和插入操作时仍需保证该视图只显示物料类别为1的物料。

所以进行插入操作的时候，如果插入的数据不满足条件，则会报错。

资料整理
火龙果软件 荐

更新视图

【例1-13】以SQL Server数据库为例，利用视图更新数据，将视图v_Item_3中物料号为“2001”的物料名改为“电源线”

```
Update  v_Item_3
        set  物料名称='电源线'
        where 物料代码='2001'
等价于:
Update  Item
        set  物料名称 = '电源线'
        where 物料代码= '2001'
```

资料整理
火龙果软件 荐

删除视图

【例1-14】以SQL Server数据库为例，利用视图删除数据，将视图v_Item_3中物料号为“2001”的物料名改为“电源线”。

```
DELETE
FROM  v_Item_3
WHERE  物料代码 = '2001'
等价于:
DELETE
FROM  Item
WHERE  物料代码 = '2001' AND 物料类别=1
```

使用DELETE语句可以通过视图删除基本表的数据。但对于依赖于多个基本表的视图，不能使用DELETE语句。

资料整理
火龙果软件
荐

加密的基本思想：根据一定的算法将原始数据（明文：Plain text）变换为不可直接识别的格式（密文：Cipher text）。

一个好的数据库加密系统应该：

① 保证长时间且大量数据不被破译。

② 加密后的数据库存储量没有明显的增加。

③ 加解密速度足够快，数据操作响应时间尽量短。

④ 加解密对数据库的合法用户操作（如数据的增、删、改、查等）是透明的。

⑤ 灵活的密钥管理机制，加解密密钥存储安全，方便使用。

资料整理
火龙果软件 荐

【例1-15】以SQL Server数据库为例，使用证书加密数据。

```
--使用证书加密数据
create database TestDB1
go
use TestDB1
go
--建测试表，data字段为要加密的列，数据类型要为varbinary
--因为加密后的数据是二进制数据
create table TestT1
(
id int identity(1,1),
  data varbinary(5000)
)
--创建主密钥
create MASTER KEY ENCRYPTION BY PASSWORD ='ABC123'
--使用数据库主密钥
open MASTER KEY DECRYPTION BY PASSWORD ='ABC123'
```

```
--用密码ABC123创建证书Cert1
create CERTIFICATE Cert1
ENCRYPTION BY PASSWORD ='ABC123'
WITH SUBJECT ='ABC123 test certificate', start_date='01/09/2020',
EXPIRY_DATE ='01/09/2021';
GO
--向测试表中写入一条测试数据
insert into TestT1(data)
select encryptbycert(cert_id('Cert1'),'证书cert1加密的内容')
go
select * from TestT1
Go
--提取加密后数据
SELECT id,cast(DecryptByCert(Cert_Id('Cert1'),data,N'ABC123') as
varchar(20)) from TestT1
```

数据加密的主要内容：

1. 数据库加密的常见实现技术
2. 数据库数据加密的粒度
3. 加密算法
4. 密钥管理
5. 数据库加密的局限性

资料整理
火龙果软件 荐

1、数据库加密的常见实现技术

① 前置代理及加密网关方式

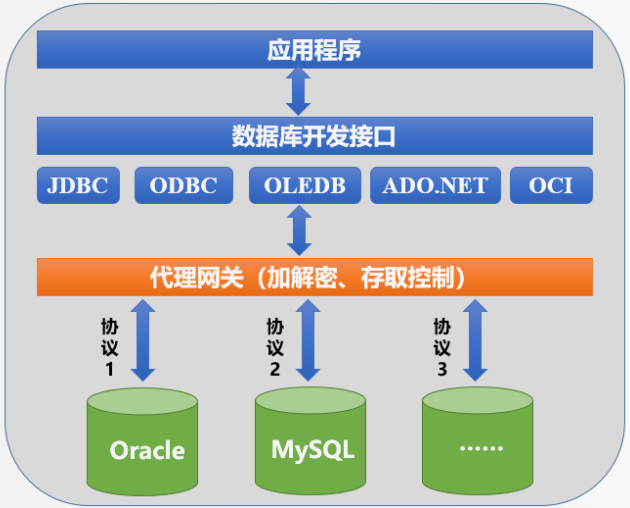
② 应用层改造加密技术

③ 基于文件级的加解密技术

④ 基于视图及触发器的后置代理技术

资料整理
火龙果软件 荐

1 前置代理及加密网关方式



前置代理及加密网关技术主要存在如下几个问题：

① 数据存储不一致问题

② 数据联合检索问题

③ 开发不透明问题

④ 数据库的优化处理、事务处理、并发处理等特性都无法使用

⑤ 存在巨大的开发工作量及很高的技术复杂度

资料整理
火龙果软件 荐

2 应用层改造加密技术

应用层改造加密技术通过**加密API(JDBC，ODBC，CAPI等)**对**敏感数据**进行加密，数据库的**底层文件**中存储加密数据；

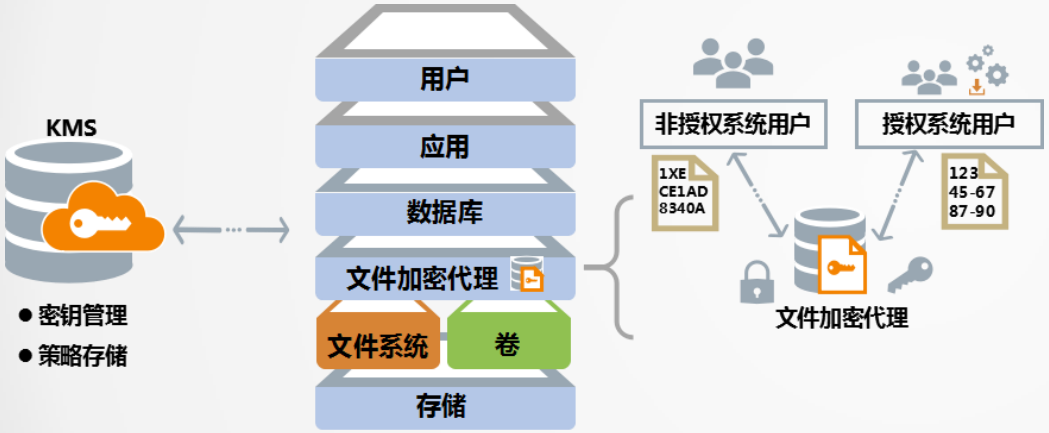
在进行数据检索时，将**密文数据返回到客户端**，再进行解密，**应用系统自行管理密钥体系**。

不足在于：应用程序必须对数据进行加解密，增加编程复杂度，而且无法对现有系统做到透明，应用程序必须进行大规模改造。

这种技术**无法利用数据库的索引机制**，加密后数据的检索性能大幅下降。

资料整理
火龙果软件 荐

3 基于文件级的加解密技术



主要存在如下几个问题：

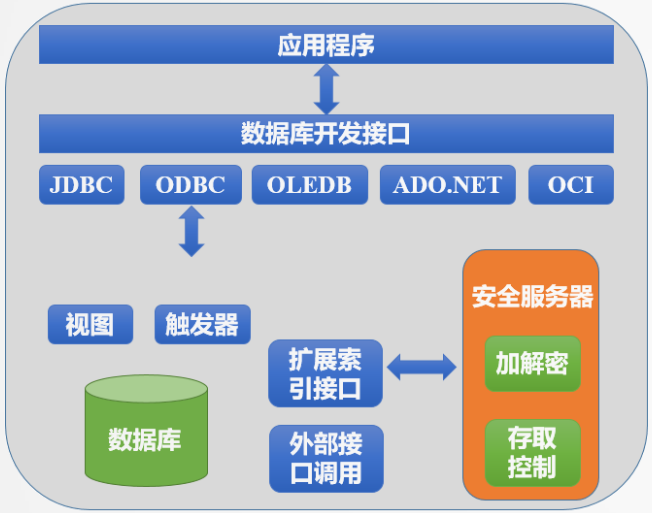
① 操作系统或文件系统级加解密，很难适应不同平台，而且与内核绑定过重，一旦出现程序故障，会对操作系统造成较大影响，容易导致业务中断。

② 操作系统或文件级加解密的控制权在系统层，因此对不同数据库账号的访问权限的设置无法单独完成，需要和其他产品与技术进行组合设置。

③ 操作系统或文件级的加解密针对落地文件进行操作，无法针对列级进行加密，加解密粒度较粗糙。

资料整理
火龙果软件 荐

4 基于视图及触发器的后置代理技术



基于视图及触发器的后置代理技术原理主要有以下四个方面：

① 通过视图实现加密数据透明查询处理

② 通过触发器实现数据加密插入与更新处理

③ 通过数据库的扩展索引接口实现加密索引

④ 通过外部接口调用实现独立于数据库的权限控制和国产加密算法

资料整理
火龙果软件 荐

数据加密的主要内容：

1. 数据库加密的常见实现技术
2. 数据库数据加密的粒度
3. 加密算法
4. 密钥管理
5. 数据库加密的局限性

1. 数据库级加密
2. 表级加密
3. 属性级加密
4. 记录级加密
5. 数据元素级加密

资料整理
火龙果软件 荐

数据加密的主要内容：

1. 数据库加密的常见实现技术
2. 数据库数据加密的粒度
3. 加密算法
4. 密钥管理
5. 数据库加密的局限性

1. 基于“消息摘要”的算法
2. 对称/非对称密钥加密算法

资料整理
火龙果软件 荐

1 基于“消息摘要”的算法

“消息摘要”(Message Digest) 是一种能产生特殊输出格式的算法，该加密算法的特点是：无论什么长度的原始数据，经过计算后输出的密文都是固定长度的。

算法的原理

根据一定的运算规则对原数据进行某种形式的提取，这种提取就是“摘要”。

消息摘要算法是“**不可逆**”的，理论上无法通过反向运算取得原数据内容，因此它通常只能被用来做**数据完整性验证**，而**不能作为原数据内容的加密方案**使用。

消息摘要算法最经典的 **MD5 算法**，是最强健的加密算法之一，常用于**文件**完整性保护和用户的密码保护。

资料整理
火龙果软件 荐

2 对称/非对称密钥加密算法

对称加密算法也称为秘密密钥算法或单密钥算法。其使用的是数据可逆的加密算法，其需要通过一个“密钥”进行数据加密处理，接收方通过一个“密钥”进行解密。加密密钥和解密密钥可以是相同的，或者可以相互推算出来。

算法的原理

非对称加密算法数据加密和解密的密钥是不同的，即算法产生两个密钥，一个是公钥，对其他用户公开；另一个是私钥，仅为自己所有。只能通过私钥才能解密用户公钥加密过的信息。

基于“非对称密钥”的加密算法主要有RSA、Diffie-Hellman等。

受RSA密钥的加密长度影响，加密速度会变慢，并且运算复杂。

经典的 MD5算法虽然具有速度快、加密程度强的优点，但是数据库中的敏感数据需要涉及查询、更新等操作，要求“可逆”，即可以解密；对称和非对称加密都需要保存密钥，数据库中又增加了需要进一步保密的数据，即密钥，增加了数据的冗余。

资料整理
火龙果软件 荐

2 对称/非对称密钥加密算法

对称加密算法也称为秘密密钥算法或单密钥算法。其使用的是数据可逆的加密算法，其需要通过一个“密钥”进行数据加密处理，接收方通过一个“密钥”进行解密。加密密钥和解密密钥可以是相同的，或者可以相互推算出来。

算法的原理

非对称加密算法数据加密和解密的密钥是不同的，即算法产生两个密钥，一个是公钥，对其他用户公开；另一个是私钥，仅为自己所有。只能通过私钥才能解密用户公钥加密过的信息。

基于“非对称密钥”的加密算法主要有RSA、Diffie-Hellman等。

受RSA密钥的加密长度影响，加密速度会变慢，并且运算复杂。

经典的 MD5算法虽然具有速度快、加密程度强的优点，但是数据库中的敏感数据需要涉及查询、更新等操作，要求“可逆”，即可以解密；对称和非对称加密都需要保存密钥，数据库中又增加了需要进一步保密的数据，即密钥，增加了数据的冗余。

资料整理
火龙果软件 荐

数据加密的主要内容：

1. 数据库加密的常见实现技术
2. 数据库数据加密的粒度
3. 加密算法
4. 密钥管理
5. 数据库加密的局限性

1. 集中密钥管理
2. 多级密钥管理

资料整理
火龙果软件 荐

数据加密的主要内容：

1. 数据库加密的常见实现技术
2. 数据库数据加密的粒度
3. 加密算法
4. 密钥管理
5. 数据库加密的局限性

1. 系统运行效率受到影响
2. 难以实现对数据完整性约束
3. 对数据的SQL语言及SQL函数受到制约
4. 密文数据容易成为攻击目标

资料整理
火龙果软件 荐

数据库审计是指**监视并记录**用户对数据库所做的**各种操作行为**的机制。

数据库安全审计系统通过对网络数据进行分析，实时并智能地识别出对数据库服务器的各种操作，并将审计结果存入审计数据库中，方便日后进行查询与分析，实现对目标数据库系统的用户操作的监控和审计。

1. 审计跟踪包括：

① 操作类型；

② 主机和用户标识；

③ 日期时间；

④ 操作的数据；

⑤ 数据的前后值。

2. 数据库安全审计系统功能：

① 实时监测并智能地分析、还原各种数据库操作过程。

② 根据规则设定及时阻断违规操作，保护重要的数据库表和视图。

③ 实现对数据库系统漏洞、登录帐号、登录工具和数据操作过程的跟踪，发现对数据库系统的异常使用。

④ 支持对登录用户、数据库表名、字段名及关键字等内容进行多种条件组合的规则设定，形成灵活的审计策略。

⑤ 提供包括记录、报警、中断和向网管系统报警等多种响应措施。

⑥ 具备强大的查询统计功能，可生成专业化的报表

资料整理
火龙果软件 荐

3. 数据库安全审计系统主要特点：

① 采用旁路技术，不影响被保护数据库的性能。

② 使用简单，不需要对被保护数据库进行任何设置。

③ 支持SQL-92标准，适用面广，可以支持Oracle、SQL Server、Sybase、Informix等多类数据库。

④ 可审计并还原SQL操作语句，审计精细度高。

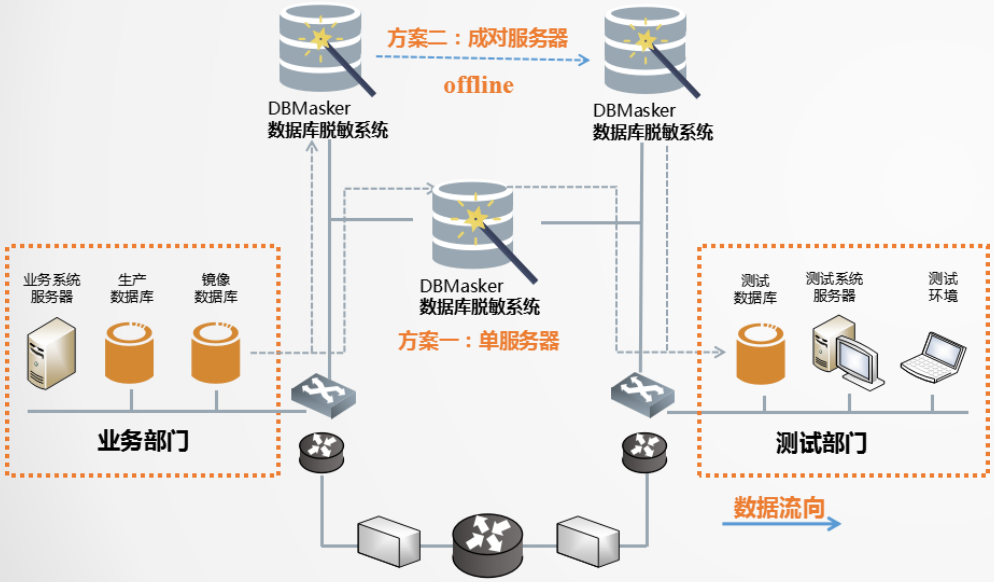
⑤ 采用分布式监控与集中式管理的结构，易于扩展。

⑥ 完备的“三权分立”管理体系，适应对敏感内容审计的管理要求。

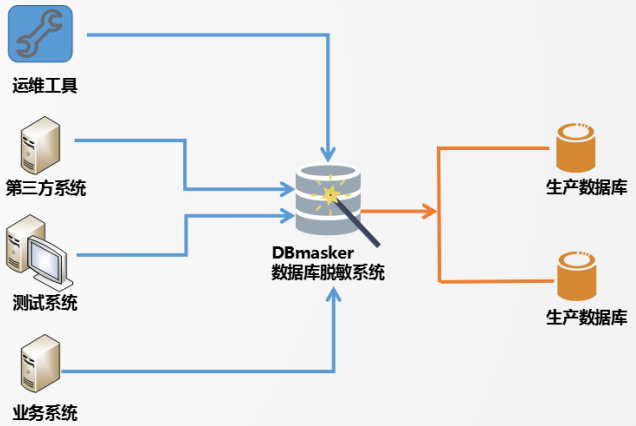
⑦ 审计功能自动记录用户对数据库的所有操作，并且存入审计日志。事后可以利用这些信息重现导致数据库现有状况的一系列事件，提供分析攻击者线索的依据。

资料整理
火龙果软件 荐

数据脱敏的基本原理是通过脱敏算法将敏感数据进行隐藏或变形，先降低敏感级别，然后对外使用。



静态脱敏部署图



动态脱敏部署图

资料整理
火龙果软件
荐

针对数据库系统的安全漏洞，建议利用漏洞扫描软件定期地扫描数据库，特别是检查数据库特有的安全漏洞，全面评估可能存在的安全漏洞和认证、授权、完整性方面的问题。

以Oracle数据库的漏洞扫描产品为例，Oracle数据库扫描扫描方式：远程扫描和登录扫描。

远程扫描

通过远程网络访问的方式，获取被扫描对象的版本、指纹等信息，判断是否受漏洞影响。

它的**优点**是能够**快速、大范围识别**网络中各系统存在的漏洞，这种方式**容易产生误报**的情况。

远程扫描结果可以作为快速初步定位的手段，但不能作为确认依据。

如果需要进一步确认，就应该进行登录扫描。

登录扫描

1、**系统登录扫描**：登录操作系统（Windows或者Linux），全面获取数据库的补丁信息，可以精准的判断是否存在漏洞。

2、**数据库登录扫描**：这种扫描方式获取的补丁信息有限，没有系统登录获取补丁的信息全面，判断漏洞存在还有局限。但是登录数据库后可以从数据表中查询到组件等相关信息。

资料整理
火龙果软件 荐

数据库防火墙作为数据库安全的设备，其主要作用是隔离来自外部的威胁。

数据库防火墙在攻击者入侵在到数据库之前就将其进行攻击阻断。

数据库防火墙关键能力

1、高可用性和高性能
2、准入控制
3、入侵防护功能
4、访问控制
5、SQL白名单
6、风险监控
7、告警
8、审计分析与追踪

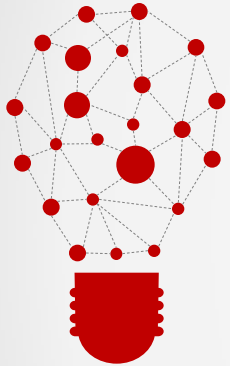
资料整理
火龙果软件 荐

数据库防火墙应用场景

应用场景	描述
1、SQL注入攻击	SQL注入攻击是数据库防火墙的核心应用场景，甚至可以说数据库防火墙就是为了防御SQL注入攻击而存在的。SQL注入攻击的发生不是由于数据库的漏洞导致，而是因为应用程序漏洞和缺陷导致，但是受到伤害和影响的则是数据库。
2、CC攻击	即使一个没有任何缺陷的应用程序也可以简单的发起cc攻击。每个应用程序都会存在资源消耗特别高的某些操作，入侵者只要同时调度这些高资源消耗的操作，就会导致数据库服务器失去响应。
3、非预期的大量数据返回	由于应用程序缺陷，在某些操作中返回了计划之外的大量数据。大量数据返回很容易引起安全性问题。
4、敏感数据未脱敏	由于历史原因，现有应用程序很少对于敏感数据进行脱敏显示。为了遵循新的安全法规和规则，为了更好的保护客户和公司，在很多情况下我们需要对于应用程序返回数据进行脱敏。
5、频繁的同类操作	通过应用程序不断的频繁获取敏感信息资料是敏感信息泄露的主要通道之一，数据库防火墙可以通过延迟，通知等响应方式来降低此类数据泄露风险。
6、超级敏感操作控制	很多应用程序往往存在着权限控制漏洞，无法控制某些敏感操作。比如统方，比如绝密资料的获取等等。
7、身份盗用和撞库攻击	撞库攻击是互联网最大的安全风险之一，绝大部分撞库攻击都是为了身份盗用。
8、验证绕行和会话劫持	由于应用程序缺陷导致起验证安全机制没有生效，比如验证码等，或者会话被劫持导致业务应用程序被非法控制。
9、业务逻辑混乱	由于应用程序漏洞导致业务逻辑混乱，比如在审批中不检查前置流程的存在性和合规性，直接触发下一个流程。

资料整理
火龙果软件
存

本章小结



- **案例：**
  - 第一个案例通过视图的原理导入数据库安全性控制的概念；
  - 第二个案例通过一个数据库数据推理与数据营销的故事，提出了通过数据推理技术导致敏感数据泄露的一种安全问题。
- **要点：**
  - **安全性控制：**包括用户标识与身份鉴别、访问控制、视图定义与查询修改、数据加密以及安全审计等；
  - **数据库脱敏技：**包括静态脱敏和动态脱敏等相关内容；
  - **数据库漏洞扫描技术：**包括远程扫描与登录扫描；
  - **数据库防火墙技术：**包括八种数据库防火墙关键能力、九种数据库防火墙的常见应用场景以及防御SQL注入攻击等内容。

资料整理
火龙果软件 荐