



第十四届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA

数智赋能 共筑未来



北京国际会议中心 | 2023/8/16-18



MySQL的数据隐私与安全最佳实践

徐轶韬

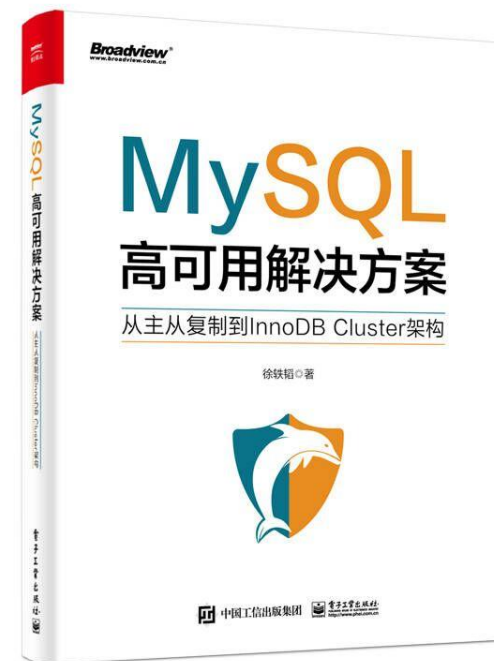
甲骨文公司MySQL解决方案首席工程师

ORACLE

徐轶韬

甲骨文公司MySQL解决方案首席工程师

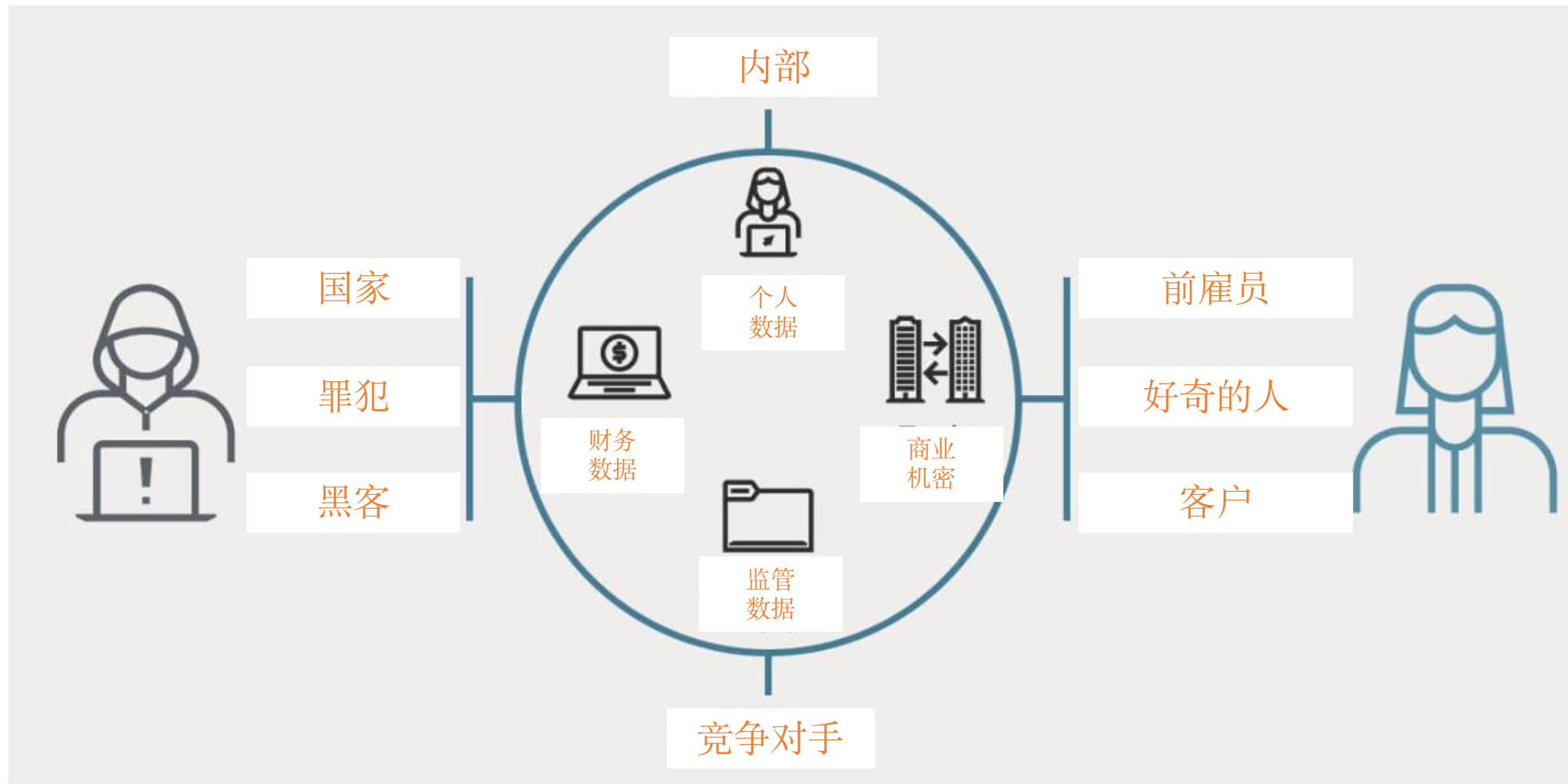
公众号“MySQL解决方案工程师”的内容作者和运营者。



《MySQL高可用解决方案——从主从复制到InnoDB Cluster架构》作者

“93%的安全漏洞可以预防”
—— Online Trust Alliance (Internet Society)

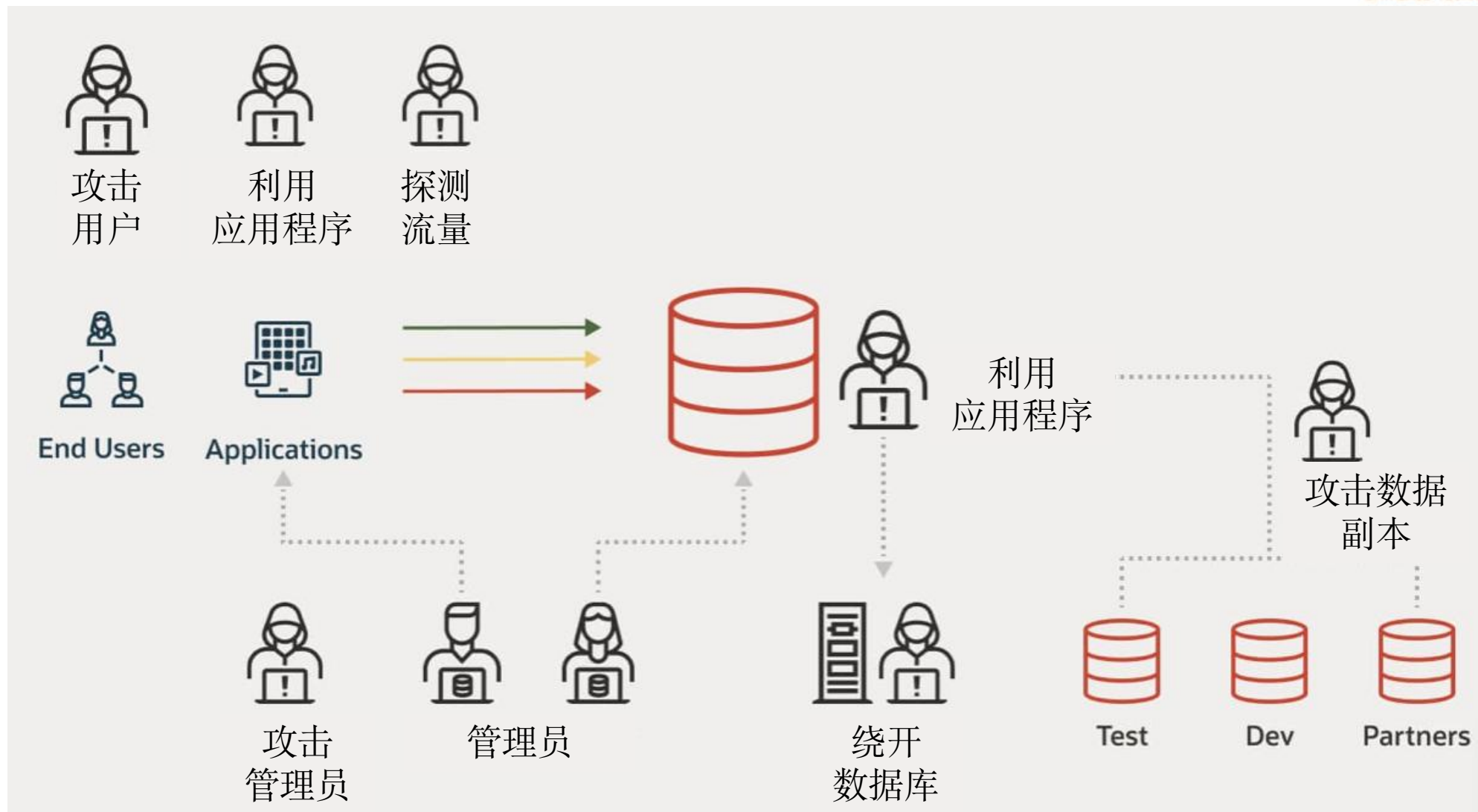
数据库面临的安全挑战



数据库可以在哪里遭受攻击?

DTCC 2023

第十四届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2023



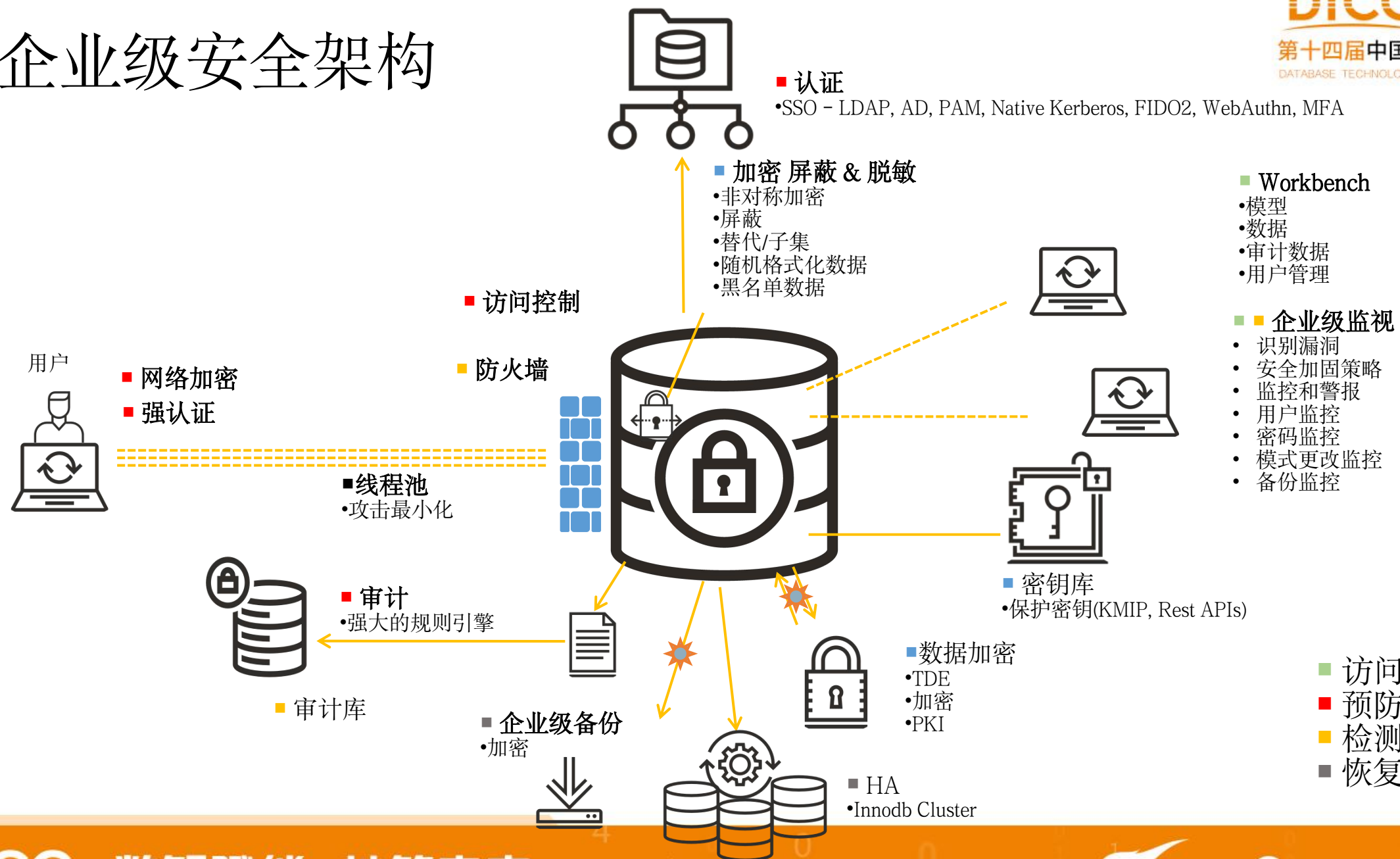
数据库遭受的攻击方式

- SQL注入
 - 防范方法：数据库防火墙、白名单、输入验证
- 缓冲区溢出
 - 防范方法：经常更新数据库软件、数据库防火墙、白名单、输入验证
- 内幕滥用
 - 防范方法：严格的访问控制、用户特定的身份验证、审计、监控、加密
- 蛮力破解
 - 防范方法：在指定次数的错误尝试后锁定帐户
- 网络窃听
 - 防范方法：所有连接和传输都需要SSL/TLS
- 恶意软件
 - 防范方法：严格访问控制、有限的网络IP访问、更改默认设置、加密

访问数据库的恶意行为

- 信息披露：获取信用卡及其他个人信息
 - 防御：数据和网络加密，执行更严格的访问控制
- 拒绝服务：运行资源密集型查询
 - 防御：资源使用限制——设置各种限制包括最大连接、会话、超时, ...
- 提升权限：检索并使用管理员权限
 - 防御：更强的身份验证、访问控制、审计
- 欺骗：检索并使用其他凭证
 - 防御：更强的帐户和密码策略
- 篡改：更改数据库中的数据，删除事务记录
 - 防御：更严格的访问控制、审计、监控、备份

企业级安全架构



安全威胁形势

82% 漏洞利用了被盗
或薄弱的凭据

13% 勒索软件漏洞增加了13%，比过
去5年的总和还多

安全模式进化
零信任安全理念

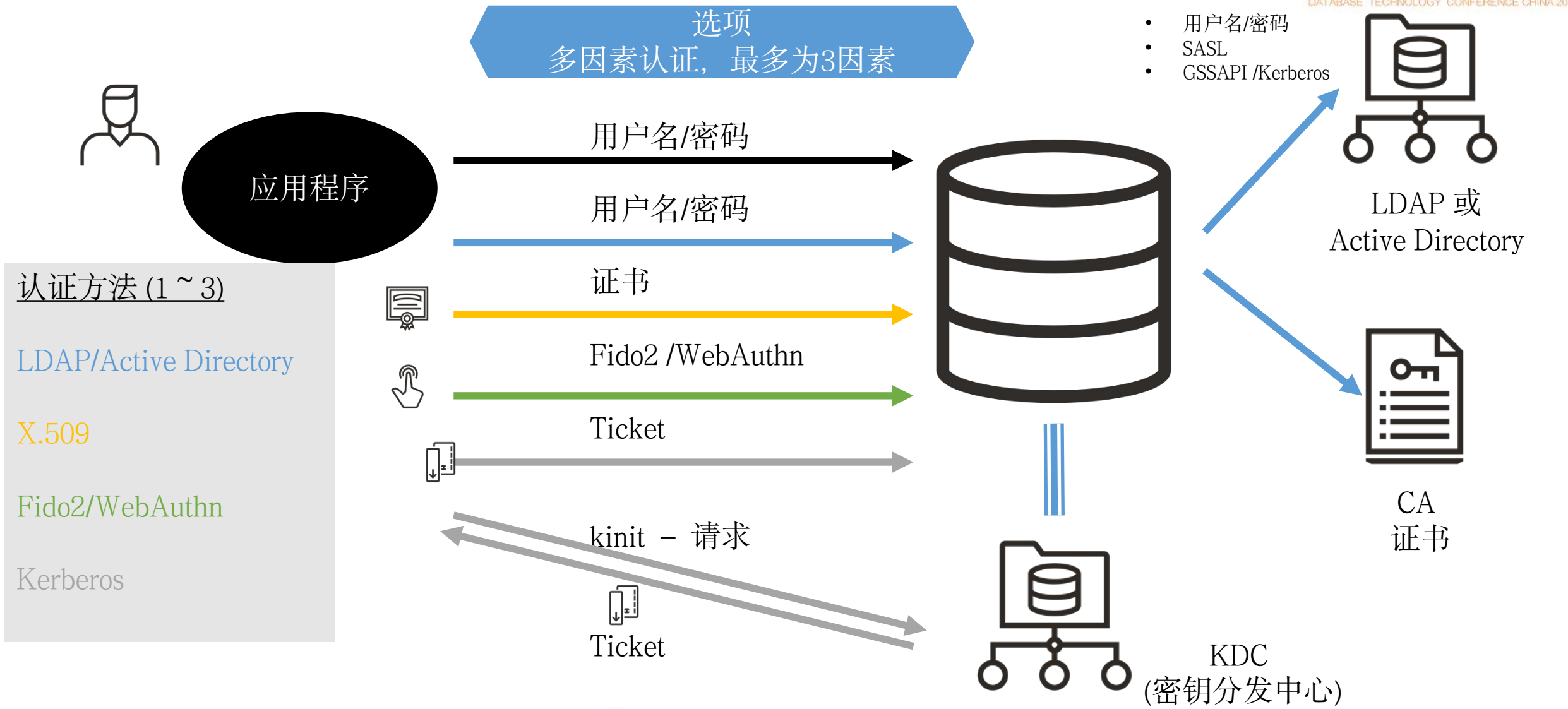




- 与集中式身份验证基础设施集成
 - 集中账户管理
 - 密码策略管理
 - 组 & 角色
 - 多重认证
 - X.509
- 验证类型
 - MySQL原生
 - 用户/密码SHA2, 可配置哈希轮数(默认5000)
 - LDAP原生
 - 通过原生LDAP 服务认证
 - Kerberos原生 – User/Pass, SASL, GSSAPI/Kerberos
 - Windows – AD
 - 访问Windows服务, 用于通过Windows Active Directory或本机主机对用户进行身份验证
 - FIDO2
 - Linux PAM 标准接口

MySQL与现有的安全基础设施集成

MySQL 客户端认证选项



密码

- 不使用明文存储——使用秘钥库
- 经常轮换，使用双重密码方式

令牌

- 使用 Kerberos 令牌 – 具有TTL

证书

- X509 有效期间短、自动轮换

主机名

- 可能的情况下进行限制——注意，通过VPN等通常不再可行

对应用程序账户授权访问

- 尽可能限制

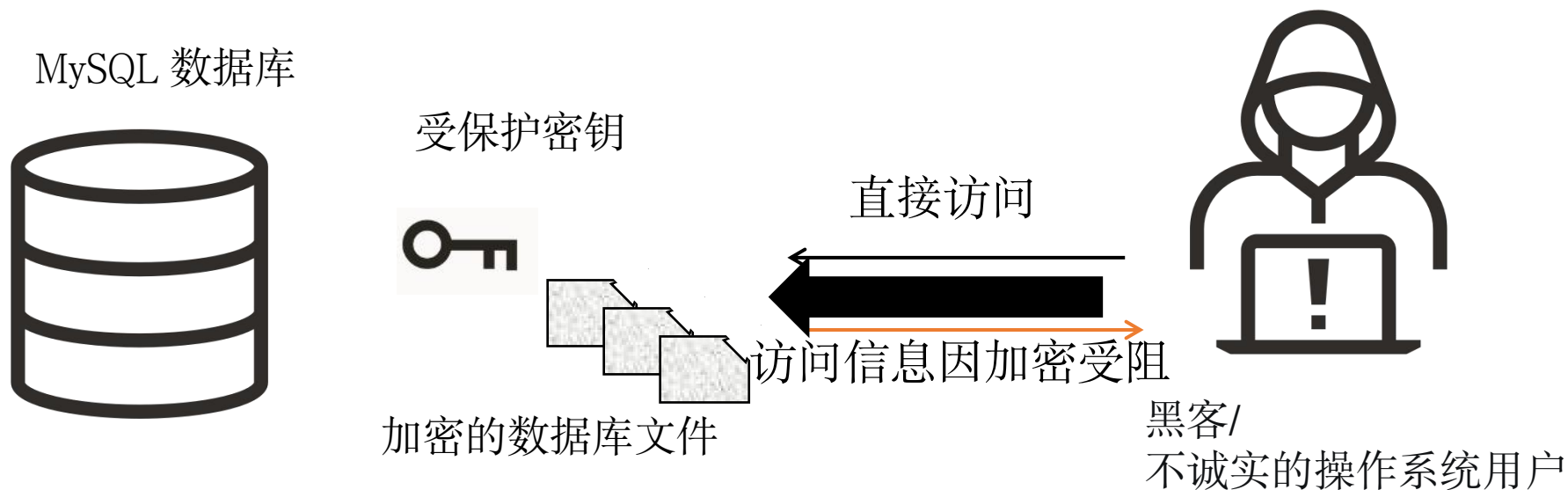
角色

- 为升级/降级的需求定义多个角色并更改角色
- SET ROLE role [, role]

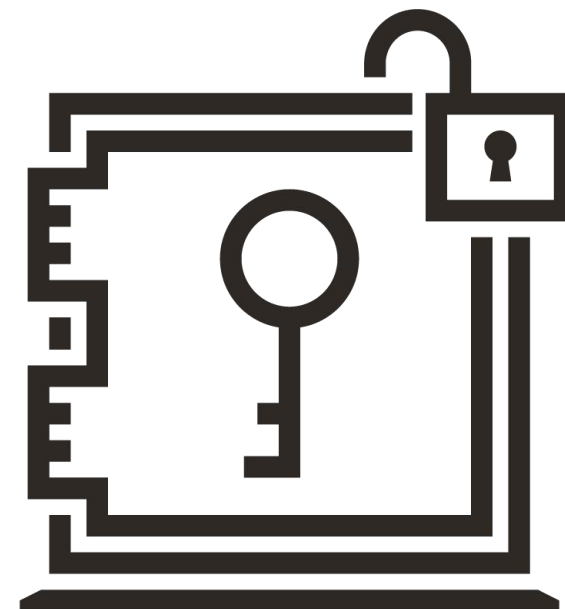
- 密码管理

- 双重密码修改
- 对于每个受影响的帐户，在服务器上建立一个新的主密码，保留当前密码作为辅助密码
- 密码更改传播到所有服务器后，修改使用受影响帐户的应用程序，使其使用帐户主密码进行连接
- 当所有应用程序从备用密码迁移到主密码后，不再需要备用密码，可以丢弃。在此更改传播到所有服务器之后，只能使用每个帐户的主密码进行连接

```
# Create the new password
ALTER USER 'appuser' '@' localhost IDENTIFIED BY 'newpass' RETAIN CURRENT PASSWORD;
# Wait for the password change to replicate to all slave servers
# Modify each application that uses the appuser1 account so that it connects to the
# servers using a password of 'newpass' rather than 'oldpass'
# Discard the old password
ALTER USER 'appuser' '@' localhost DISCARD OLD PASSWORD;
```



OKV 或
KMIP 兼容密钥库



取得/发送MySQL密钥
至MySQL密钥环



密匙环上的密钥只能被内部组件、内部代码或内部插件访问

钥匙环不是持久化的，而是在内存中受保护的

ACLs – 密钥是为谁而设的？例如，InnoDB表空间

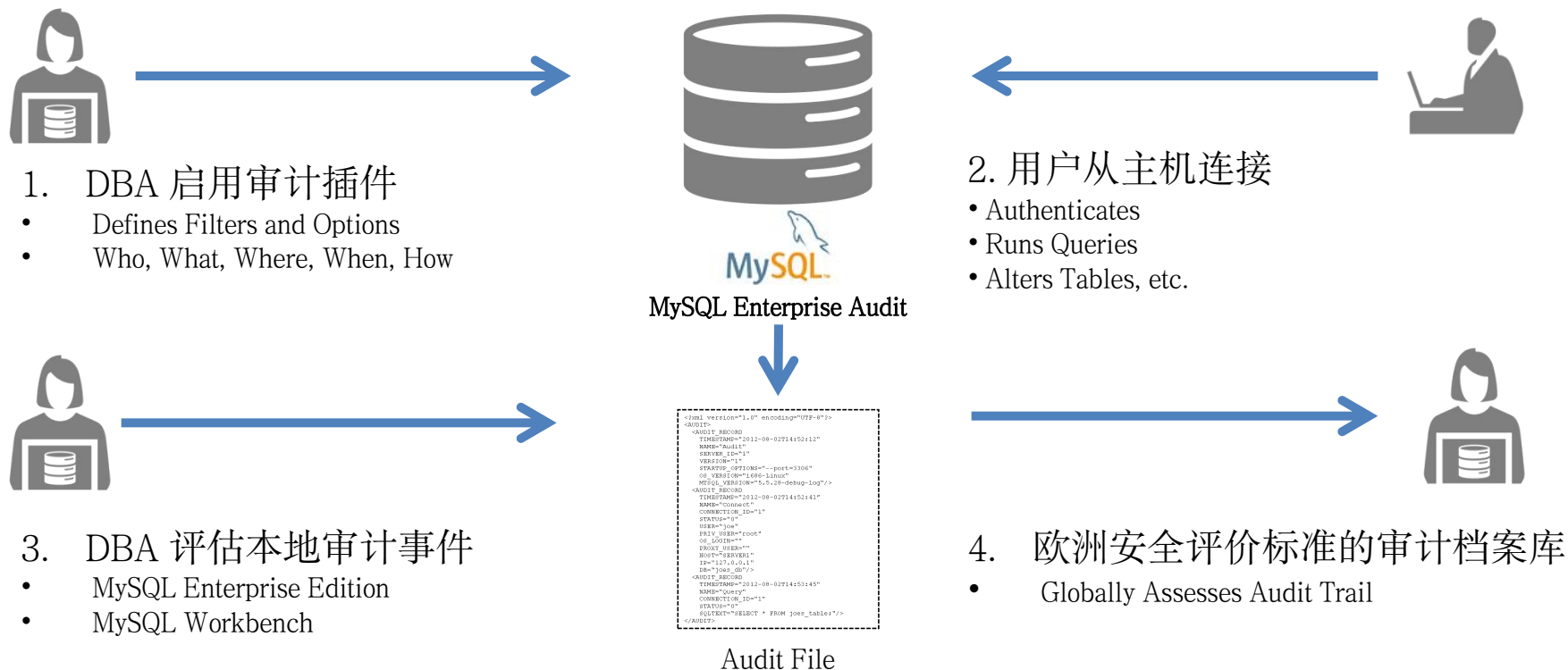
MySQL的透明数据加密

- CREATE TABLE 的选项
- ENCRYPTION= “Y”
- 轮换密钥的方法简单
- ALTER INSTANCE ROTATE INNODB MASTER KEY
- Keyring 插件
- 用于检索密钥
- 组件/插件
- 插件类型 : keyring
- 能够在InnoDB初始化之前加载--early-plugin-load
- MySQL 文件
- 支持表空间文件、撤销日志、重做日志、二进制日志, 及审计日志加密
- IMPORT/EXPORT 加密表
- 支持主密钥轮换



- 审计跟踪是基本的安全性最佳实践
- “信任但要验证”的安全方法
 - 确保具有强权限的用户不会滥用这些权限
- 业务审计——数据有效性
 - 数据库数据准确/正确的证明
 - 证明数据没有被篡改
- 入侵分析——作为纵深防御策略的组成部分
 - 主动——数据库被入侵
 - 反应——数据库是如何被入侵的？被改变了什么？拿走了什么？等等。

MySQL Enterprise Audit: 工作流程

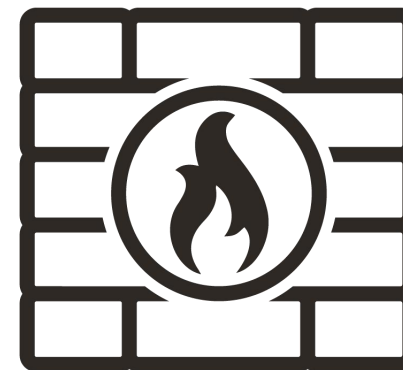


数据库防火墙

- SQL注入攻击
 - #1 Web应用程序漏洞
 - 77%的网站存在漏洞

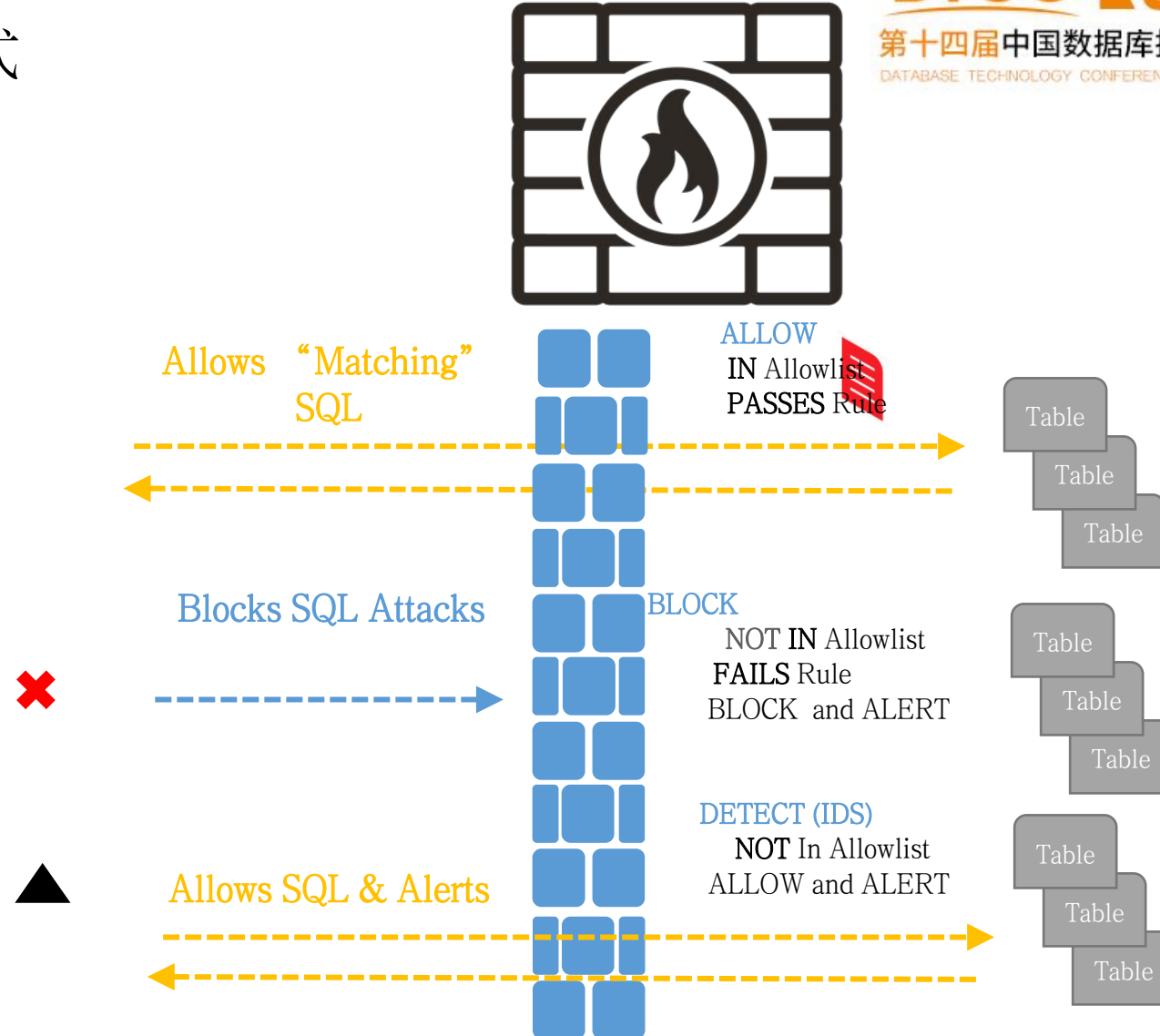
MySQL Enterprise Firewall

- 实时监控数据库语句
- 自动允许列表“规则”为任何应用程序生成
- 定义阻断防火墙规则
- 阻断SQL注入攻击
- 入侵检测系统



MySQL Enterprise Firewall: 操作模式

- 1 ALLOW – 执行 SQL
 - SQL 匹配白名单
 - SQL 通过规则
- 2 BLOCK – 阻挡请求
 - 不在白名单内
 - SQL 不符合规则
 - 阻挡模式
- 3 DETECT – 执行SQL & 警告
 - 不在白名单内
 - SQL 不符合规则
 - 警告模式



- 数据屏蔽
- 数据屏蔽是一种通过替换真实值来隐藏敏感信息的方法

Employee 表

ID	Last	First	SSN
1111	Smith	John	555-12-5555
1112	Templeton	Richard	444-12-4444



随机生成数据

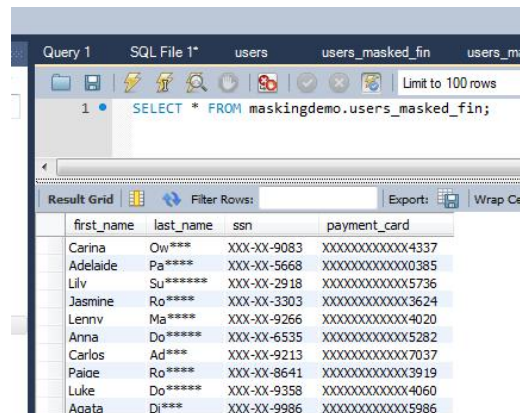
ID	Last	First	SSN
2874	Smith	John	XXX-XX-5555
3281	Templeton	Richard	XXX-XX-4444



屏蔽后的视图

- 数据屏蔽 & 生成随机数据
- 数据屏蔽
 - 字符串屏蔽
 - 基于字典替换
 - 指定屏蔽
 - SSN
 - 支付卡: 严格/宽松
- 生成随机数据
 - 范围内随机数字
 - Email
 - 支付卡 (符合Luhn 算法检查)
 - SSN
 - 基于字典

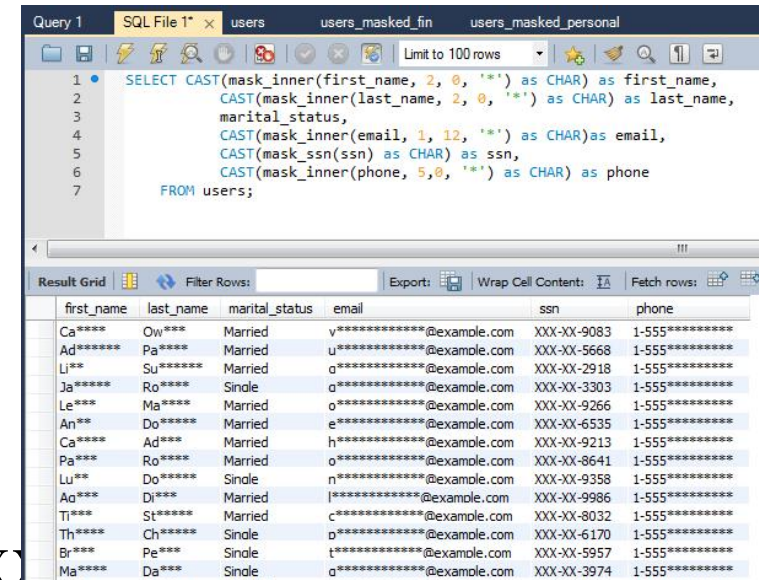
- 屏蔽字符串
 - 屏蔽部分字符: ArthXXXXnt
 - 屏蔽起始字符:
 - XXthurDeXX
- SSN 屏蔽: XXXX-XX-1234
- 支付卡屏蔽
 - 严格: XXXXXXXXXXXXXXXX7395, 宽松: 493812XXXXXXXXXX
- 基于字典屏蔽
 - gen_blocklist(“007”, “00designations”, “Cover_identity”) => Universal Exports



Query 1 SQL File 1* users users_masked_fin users_mas

```
1 • SELECT * FROM maskingdemo.users_masked_fin;
```

first_name	last_name	ssn	payment_card
Carina	Ow***	XXX-XX-9083	XXXXXXXXXXXX4337
Adelaide	Pa****	XXX-XX-5668	XXXXXXXXXXXX0385
Lilv	Su*****	XXX-XX-2918	XXXXXXXXXXXX5736
Jasmine	Ro****	XXX-XX-3303	XXXXXXXXXXXX3624
Lennv	Ma****	XXX-XX-9266	XXXXXXXXXXXX4020
Anna	Do****	XXX-XX-6535	XXXXXXXXXXXX5282
Carlos	Ad****	XXX-XX-9213	XXXXXXXXXXXX7037
Paide	Ro****	XXX-XX-8641	XXXXXXXXXXXX3919
Luke	Do****	XXX-XX-9358	XXXXXXXXXXXX4060
Aaata	Di***	XXX-XX-9986	XXXXXXXXXXXX5986



Query 1 SQL File 1* users users_masked_fin users_masked_personal

```
1 • SELECT CAST(mask_inner(first_name, 2, 0, '*') as CHAR) as first_name,  
2 CAST(mask_inner(last_name, 2, 0, '*') as CHAR) as last_name,  
3 marital_status,  
4 CAST(mask_inner(email, 1, 12, '*') as CHAR) as email,  
5 CAST(mask_ssn(ssn) as CHAR) as ssn,  
6 CAST(mask_inner(phone, 5, 0, '*') as CHAR) as phone  
7 FROM users;
```

first_name	last_name	marital_status	email	ssn	phone
Ca****	Ow***	Married	v*****@example.com	XXX-XX-9083	1-555*****
Ad*****	Pa****	Married	u*****@example.com	XXX-XX-5668	1-555*****
Lj***	Su*****	Married	q*****@example.com	XXX-XX-2918	1-555*****
Ja****	Ro****	Single	q*****@example.com	XXX-XX-3303	1-555*****
Le***	Ma****	Married	o*****@example.com	XXX-XX-9266	1-555*****
An**	Do****	Married	e*****@example.com	XXX-XX-6535	1-555*****
Ca****	Ad****	Married	h*****@example.com	XXX-XX-9213	1-555*****
Pa***	Ro****	Married	o*****@example.com	XXX-XX-8641	1-555*****
Lu**	Do****	Single	n*****@example.com	XXX-XX-9358	1-555*****
Ao***	Di****	Married	l*****@example.com	XXX-XX-9986	1-555*****
Tj***	St****	Married	c*****@example.com	XXX-XX-8032	1-555*****
Th****	Ch*****	Single	p*****@example.com	XXX-XX-6170	1-555*****
Br***	Pe***	Single	t*****@example.com	XXX-XX-5957	1-555*****
Ma****	Da***	Single	q*****@example.com	XXX-XX-3974	1-555*****

MySQL Enterprise Masking and De-Identification

- 生成随机数据
- 范围内生成:
- `gen_rnd(10000, 20000) => 12503`
- Email: `kajsm.hamskdk@example.com`
- 支付卡: `7389026626032990`
 - 可配置长度: 12 ~ 19 位数字

```
UPDATE users SET designation = CAST(gen_dictionary("Designations") as CHAR) where id > 0 ;
UPDATE users SET ssn = CAST(gen_rnd_ssn() as CHAR) where id > 0;
UPDATE users SET salary = gen_range(40000, 55000) WHERE designation in ('Developer') and id > 0;
UPDATE users SET salary = gen_range(60000, 70000) WHERE designation in ('Senior Developer') and id > 0;
UPDATE users SET salary = gen_range(75000, 90000) WHERE designation in ('Principal Developer', 'Senior Developer') and id > 0;
UPDATE users SET salary = gen_range(95000, 120000) WHERE designation in ('Architect', 'Senior Manager') and id > 0;
UPDATE users SET salary = gen_range(125000, 150000) WHERE designation in ('Director') and id > 0;
UPDATE users SET salary = gen_range(160000, 200000) WHERE designation in ('Senior Director') and id > 0;
UPDATE users SET salary = gen_range(220000, 250000) WHERE designation in ('Vice President') and id > 0;
UPDATE users SET email = gen_rnd_email() where id > 0;
UPDATE users SET phone = gen_rnd_us_phone() where id > 0;
UPDATE users SET payment_card = gen_rnd_pan() where id > 0;
```

- 软件升级和补丁
 - MySQL
 - 开发框架和ORM（对象关系映射）
 - 应用程序库
- 高可用、容灾、安全性，及备份齐头并进，使用集群、副本集、集群集、运行备份
- MySQL安全性日常检查及再次检查
- 监视及审阅变更
- 审阅嫌疑动作的审计数据
- 审阅应用程序代码
 - 配置
 - 如何连接
 - 连接信息在哪里？
 - 数据有效性——是否检查输入数据的类型、长度等等
 - 使用预处理语句，绑定动态数据

- CIS Benchmark for MySQL 8.0 EE
- https://www.cisecurity.org/benchmark/oracle_mysql/
- CIS建议被美国国防部云计算安全推荐为指南(SRG)、支付卡行业数据安全标准(PCI DSS)、健康保险可移植性和责任法案(HIPAA)、联邦信息安全管理法(FISMA)、联邦风险和授权管理计划(FedRAMP)和国家标准与技术研究所(NIST)认可为安全配置标准。





- Oracle安全实践
 - 关键补丁更新、安全警报、公告
 - 在支持期间处理敏感的“私人/个人”信息
 - 源代码保护
 - 安全编码标准
 - 安全分析与测试
 - 员工筛选及教育
 - 架构安全性审查
 - 受信任的安装包存储库
- MySQL 安全性指南
- MySQL 提供安全性方面的知识库

Oracle Support, Security, and Compliance References

Oracle's corporate security

- <https://www.oracle.com/corporate/security-practices/>

Oracle's cloud compliance

- <https://www.oracle.com/cloud/compliance/>

The Critical Patch Updates and Security Alerts Page

- <https://www.oracle.com/security-alerts/>

Instructions on how to report security vulnerabilities

- <https://www.oracle.com/corporate/security-practices/assurance/vulnerability/reporting.html>

Oracle Software Technical Support Policies

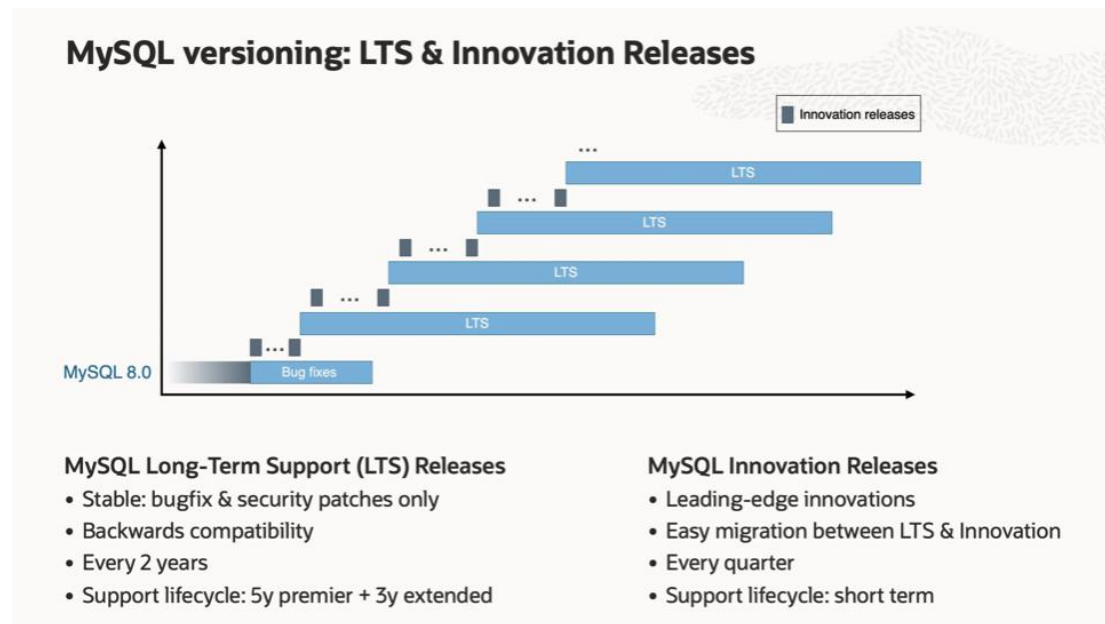
- <https://www.oracle.com/us/assets/057419.pdf>

The agreements for Oracle Cloud (including the Data Processing Agreement for Oracle Services) <http://www.oracle.com/corporate/contracts/cloud-services/index.html>

MySQL版本——长期稳定版与创新版 (LTS & Innovation Releases)

<https://blogs.oracle.com/mysql/post/introducing-mysql-innovation-and-longterm-support-lts-versions>

- 长期稳定版将与8.0.xx一起发布
 - 8.0.xx+ 仅包含错误修复、不会引入非必要功能
- 创新版
 - 包含错误修复和新特性，与现有的8.0类似



MySQL 8.1.0 是第一个创新版本，
MySQL 8.0.34+ 将仅包含错误修复直至 8.0 (EOL)2026年4月

THANKS

TDDL

DistributedTable

DBproxy

HBase

PostgreSQL

SSD

MongoDB

GreatDB

Cassandra

Hyperbase

Hubble

DataCenter

VisualDataPlatform

Blockchain

ArgoDB

Distributed

DatabaseKernel

TemporalData

CloudnativeData

AIalgorithm