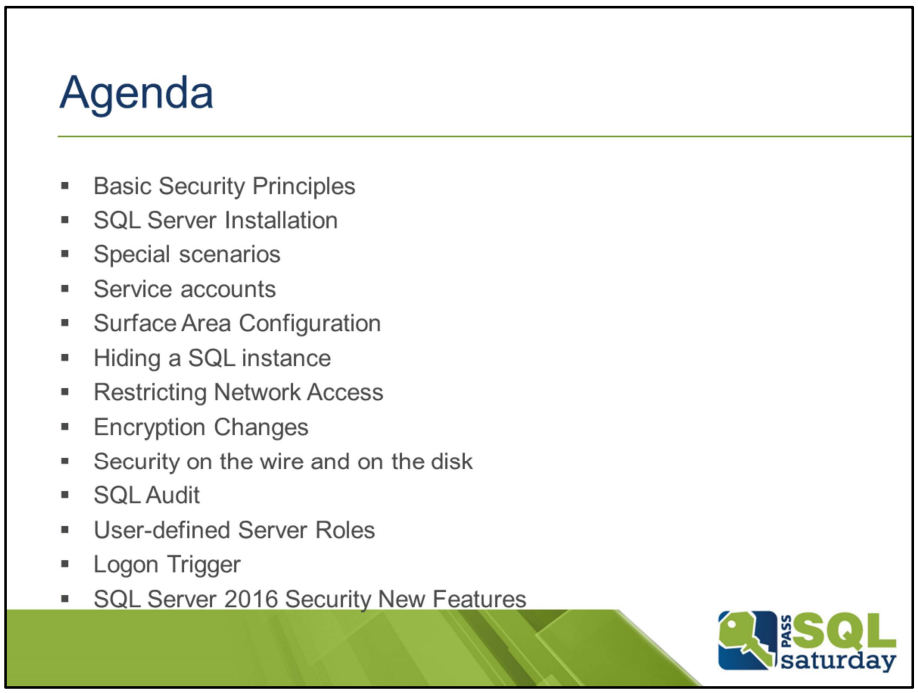
SQL Server 2014 Security Hardening

Diego Miranda
Premier Field Engineer - Microsoft
diegomiranda1@hotmail.com

PASS SQL saturday
#424 | SÃO PAULO 2015

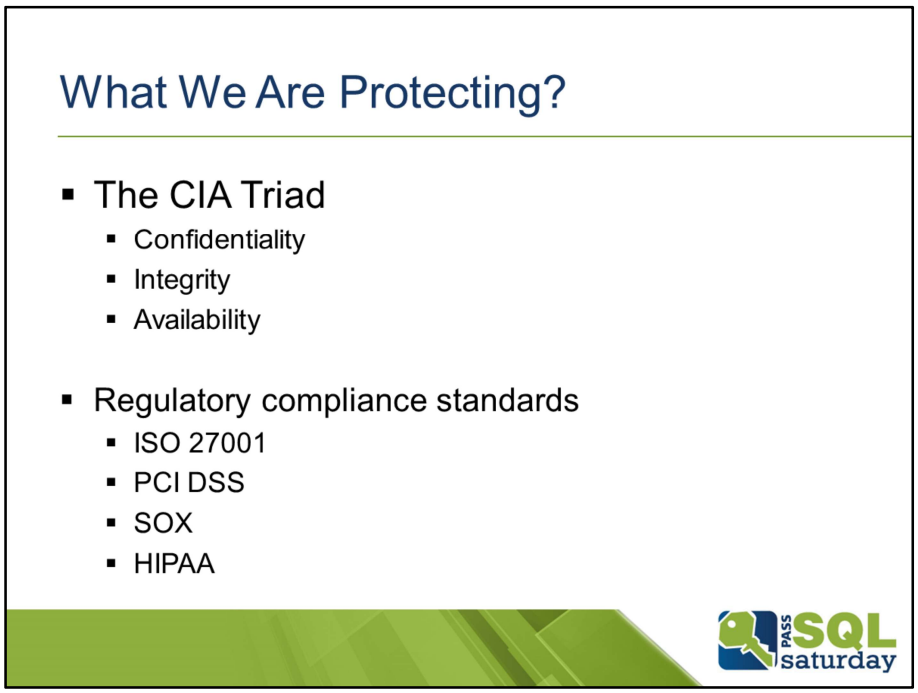


SQL Server 2012 Security Best Practices - Operational and Administrative Tasks
http://download.microsoft.com/download/8/F/A/8FABACD7-803E-40FC-ADF8-355E7D218F4C/SQL_Server_2012_Security_Best_Practice_Whitepaper_Apr2012.docx

Security Center for SQL Server Database Engine and Azure SQL Database
https://msdn.microsoft.com/en-us/library/bb510589(v=sql.120).aspx

Falem que o evento é gratuito pra todos, e que os patrocinadores são os responsáveis por garantir o coffee e a infra-estrutura basica pro evento acontecer. Em contra partida, eles querem mostrar seus produtos e serviços, é legal aos participantes conhecerem o que eles tem a oferecer e aceitar receber contatos deles via email.

Neste slide é pra convidar os participantes pro PASS Summit, em Seattle no próximo mês. Os códigos de promoção que dão 150USD de desconto é do SQLManiacs (Código da esquerda) e SQLServerDF (Código da direita). Ambos dão o mesmo desconto, o cara fica a vontade pra usar qualquer um.



ISO/IEC 27001 - Information security management
http://www.iso.org/iso/home/standards/management-standards/iso27001.htm

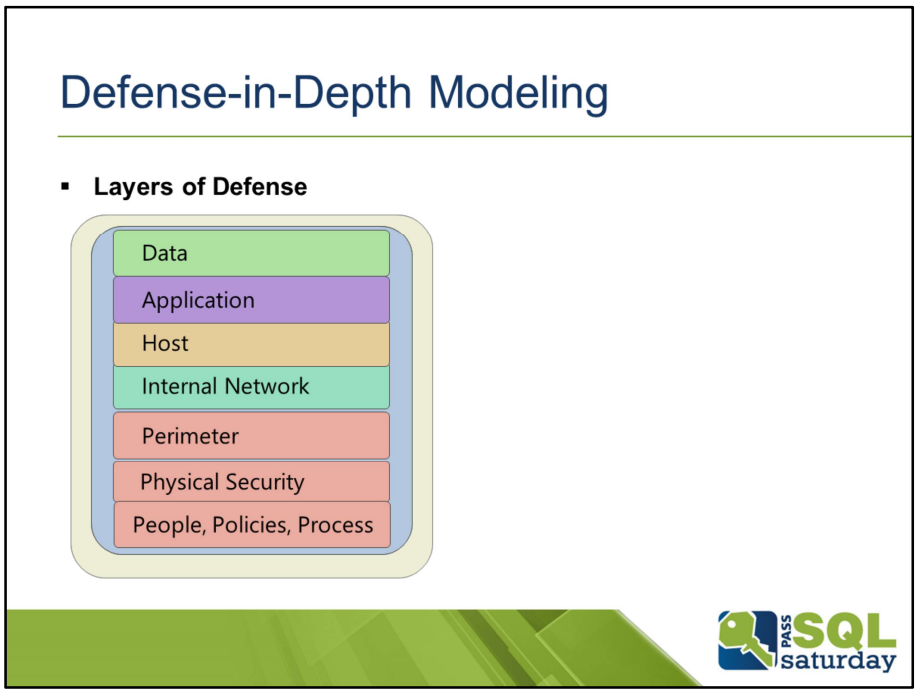
ISO 27001 - Details
https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1:en

Why We Protect It?

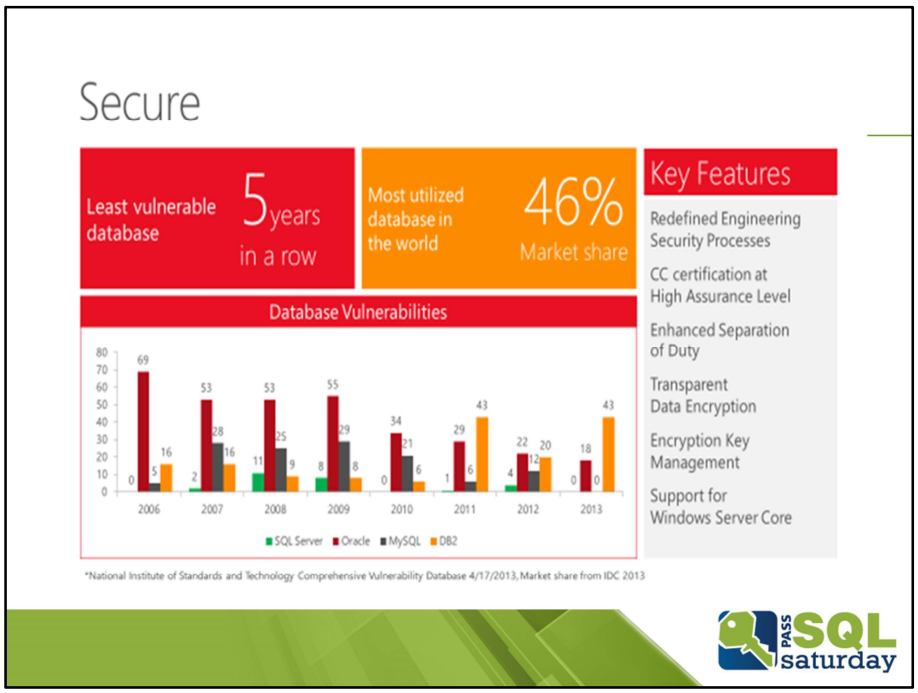
- Security decisions are risk management decisions
  - Risk can never be reduced to zero
- Assess Risk
  - Identify and prioritize risks to the business
- Implement Controls
  - Deploy and operate control solutions to reduce risk to the business.
- Measure Effectiveness
  - Verify that controls are providing the expected degree of protection

How We Protect It?

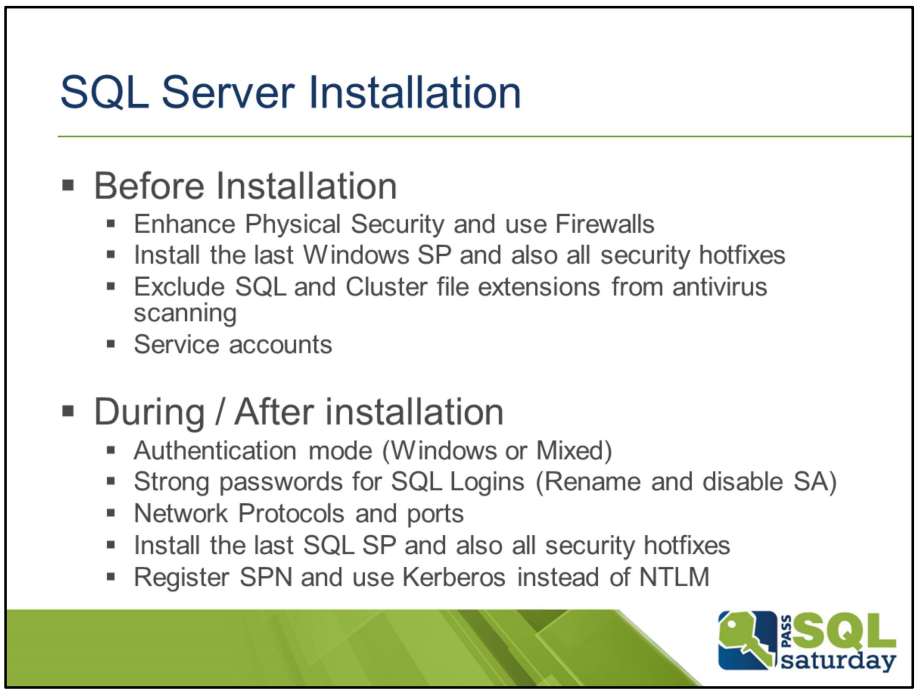
- Principle of Least Privilege (POLP)
  - Access varies based on minimum amount of privilege for the requirement
- Reduce the Attack Surface
  - Lower attack surface directly reduces the probability of a successful attack
- Defense-in-Depth modeling



Security Content Overview
http://technet.microsoft.com/en-us/library/cc767969.aspx



https://nvd.nist.gov/



SP = Service Pack
Configure Windows Service Accounts and Permissions
https://msdn.microsoft.com/en-us/library/ms143504(v=sql.120).aspx

How to choose antivirus software to run on computers that are running SQL Server
http://support.microsoft.com/kb/309422/en-us

Security Considerations for a SQL Server Installation
http://msdn.microsoft.com/en-us/library/ms144228.aspx

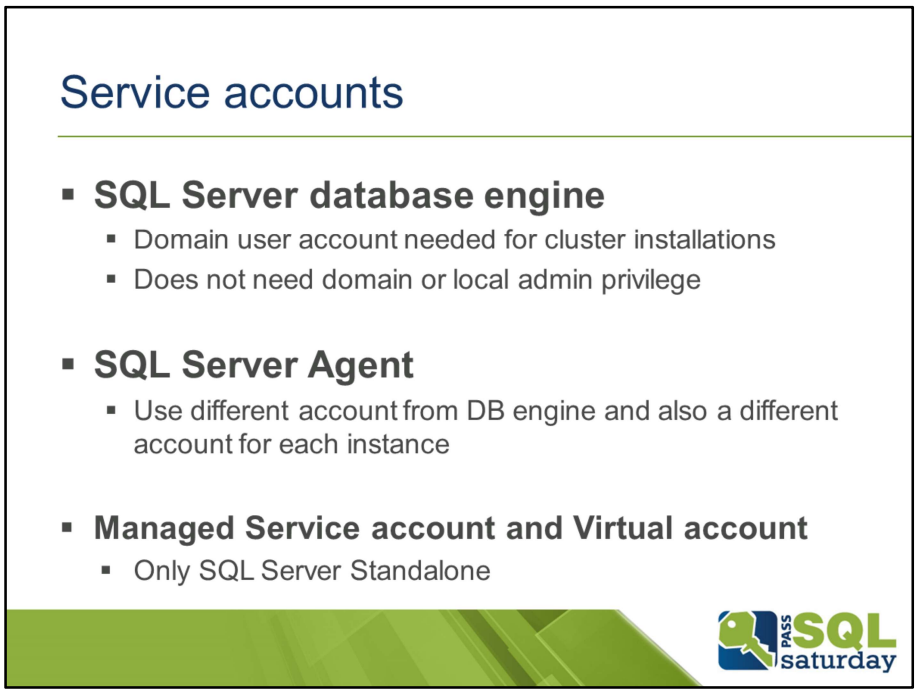
Securing SQL Server
http://msdn.microsoft.com/en-us/library/bb283235.aspx

Register a Service Principal Name for Kerberos Connections
http://msdn.microsoft.com/en-us/library/ms191153.aspx

Configure the Windows Firewall to Allow SQL Server Access
http://technet.microsoft.com/en-us/library/cc646023.aspx

Special scenarios

- DMZ
  - Not recommended
  - Why open many TCP ports with internal network?
- Domain Controller
  - Possible but not recommended
  - If needs SQL Cluster, not supported



Changing SQL service account: Always use SQL Server Configuration Manager

Configure Windows Service Accounts and Permissions
http://msdn.microsoft.com/en-us/library/ms143504.aspx

NT AUTHORITY\SYSTEM account

Do not delete this account or remove it from the SYSADMIN fixed server role (Only until SQL Server 2008 R2, however, from SQL Server 2012 to above NT AUTHORITY\SYSTEM doesn't have sysadmin permission anymore)

Used by

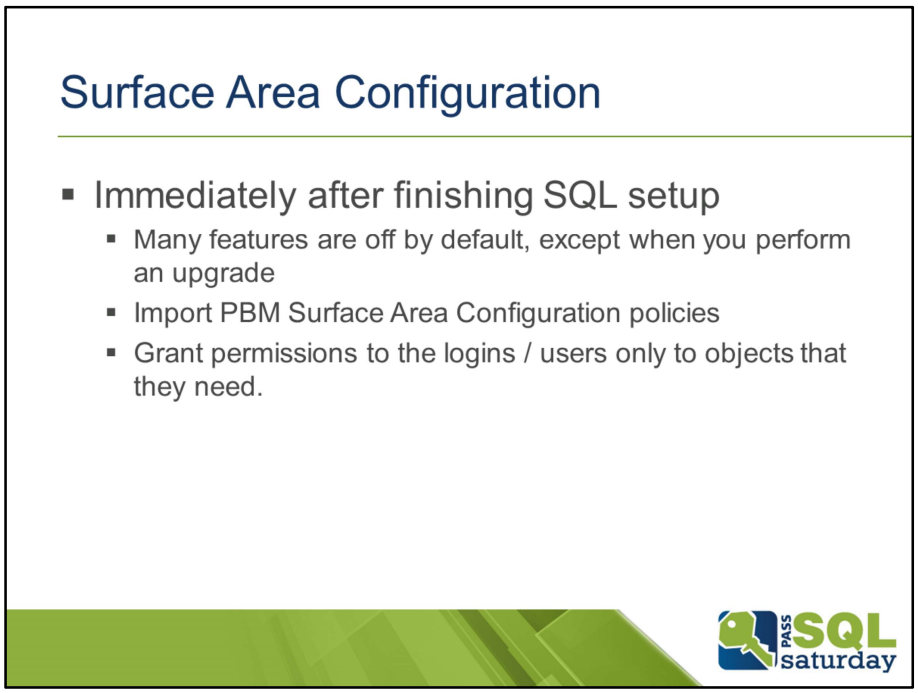
Cluster service to check SQL server service health !!!

Microsoft Update and to apply service packs and hotfixes

SQL Writer Service (VSS provider)

How to make unwanted access to SQL Server 2005 by an operating system administrator more difficult
http://support.microsoft.com/kb/932881/en-us

What's New in SQL Server Installation
https://technet.microsoft.com/en-us/library/bb500459(v=sql.110).aspx



Only turn on features if you really need them. (Xp_cmdshell and Ad Hoc Distributed Queries parameters, TRUSTWORTHY Database Property)

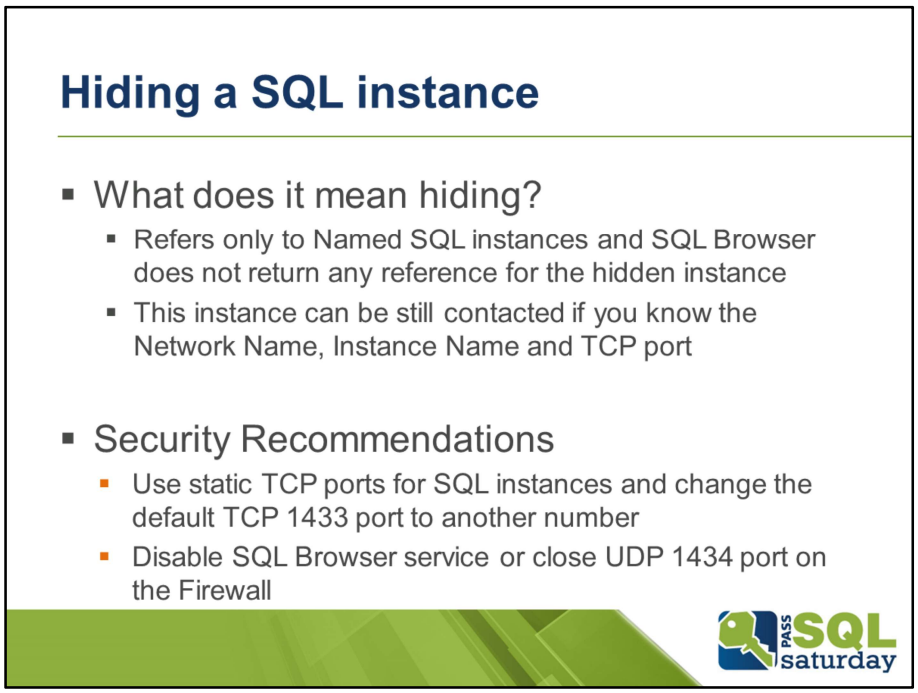
Administer Servers by Using Policy-Based Management
http://msdn.microsoft.com/en-us/library/bb510667.aspx

PBM - Facets
http://blogs.msdn.com/b/sqlpbm/archive/2008/05/24/facets.aspx

Enterprise Policy Management Framework
http://sqlblog.com/blogs/lara_rubbelke/archive/2008/11/16/enterprise-policy-management-framework-on-codeplex.aspx

Don't give sysadmin and db_owner to the app login.
Give specific access such as:
DDL_admin (user_db)
DB_DataReader e Db_DataWriter (User_db)
View definition (User_db)
Connect (User_db)

Show plan (User_db)
View database state (User_db)
exec procedures (User_db)
Grant select “SQL_DEPENDENCIES e
SQL_EXPRESSION_DEPENDENCIES” (user_DB)



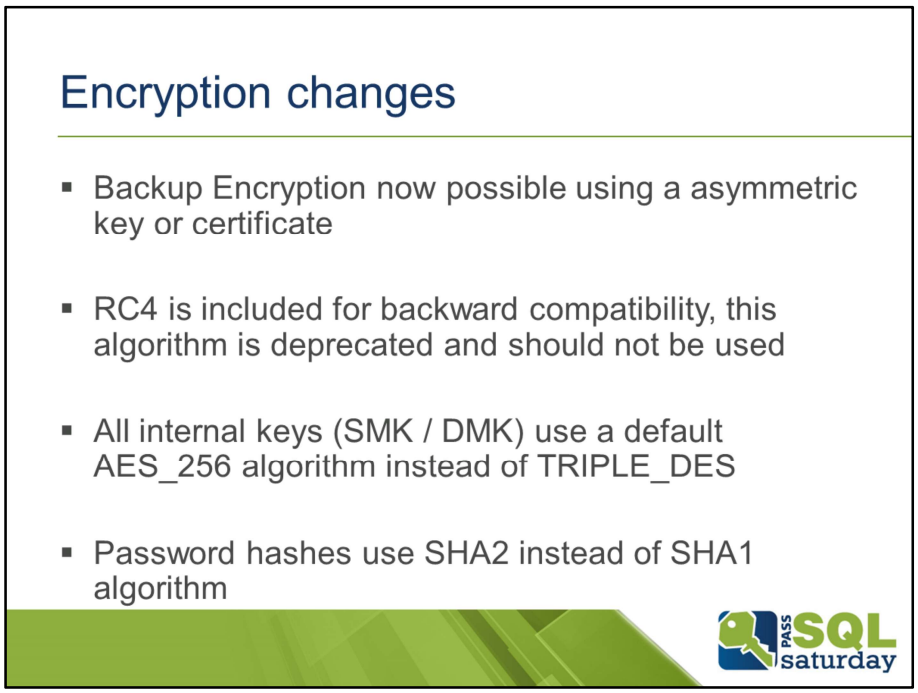
Think about how to manager the connection strings from all the apps after hiding SQL Server instance. Suggestion: Centralized location, access control and encryption

Network Connectivity

- Disable the protocols that will not be used
  - Standalone: Shared Memory and TCP
  - Cluster: Shared Memory, TCP and Named Pipes
- All logins are granted permission on ENDPOINTS for standard protocols
  - Explicit DENYs can be used to block specific logins
  - Endpoints created for: All enabled protocols, DAC, Mirroring, Service Broker, Http

Restricting Network Access

- Everyone group has the right “Access the computer from the network”
  - Use the Local Security Policy or domain GPO in order to remove this right from Everyone group
- Restrict which users have interactive logon access to the SQL Server machine
  - Non admin users doesn’t need rights “Allow log on locally” and “Allow log on through Remote Desktop Services”



SQL 2008 uses the TRIPLE_DES encryption algorithm to protect the service master key (SMK) and the database master key (DMK). The Service Master Key (SMK) is a server-wide encryption key that sits at the top of the SQL Server Encryption Hierarchy. The SMK is created automatically the first time it's needed to encrypt a Database Master Key or other secret. SQL 2012 uses AES_256 algorithm by default to protect SMK and DMK. AES_256 is a stronger algorithm than TRIPLE_DES.

See section “Encrypting Connections to SQL Server ” in BooksOnLine
http://technet.microsoft.com/en-us/library/ms189067(v=SQL.105).aspx

Demo instructions:

To configure SSL:

- Install a certificate in the Windows certificate store of the server computer.
- Click Start, in the Microsoft SQL Server program group, point to Configuration Tools, and then click SQL Server Configuration Manager.
- Expand SQL Server Network Configuration, right-click the protocols for the server you want, and then click Properties.
  On the **Certificate** tab, configure the Database Engine to use the certificate.
- On the **Flags** tab, view or specify the protocol encryption option. The login packet will always be encrypted.
    - When the **ForceEncryption** option for the Database Engine is set to **Yes**, all

client/server communication is encrypted and clients that cannot support encryption are denied access.

- When the **ForceEncryption** option for the Database Engine is set to **No**, encryption can be requested by the client application but is not required.

- SQL Server must be restarted after you change the **ForceEncryption** setting.

Credentials (in the login packet) that are transmitted when a client application connects to SQL Server are always encrypted. SQL Server will use a certificate from a trusted certification authority if available. If a trusted certificate is not installed, SQL Server will generate a self-signed certificate when the instance is started, and use the self-signed certificate to encrypt the credentials. This self-signed certificate helps increase security but it does not provide protection against identity spoofing by the server. If the self-signed certificate is used, and the value of the **ForceEncryption** option is set to **Yes**, all data transmitted across a network between SQL Server and the client application will be encrypted using the self-signed certificate

Certificate Requirements
For SQL Server to load a SSL certificate, the certificate must meet the following conditions:

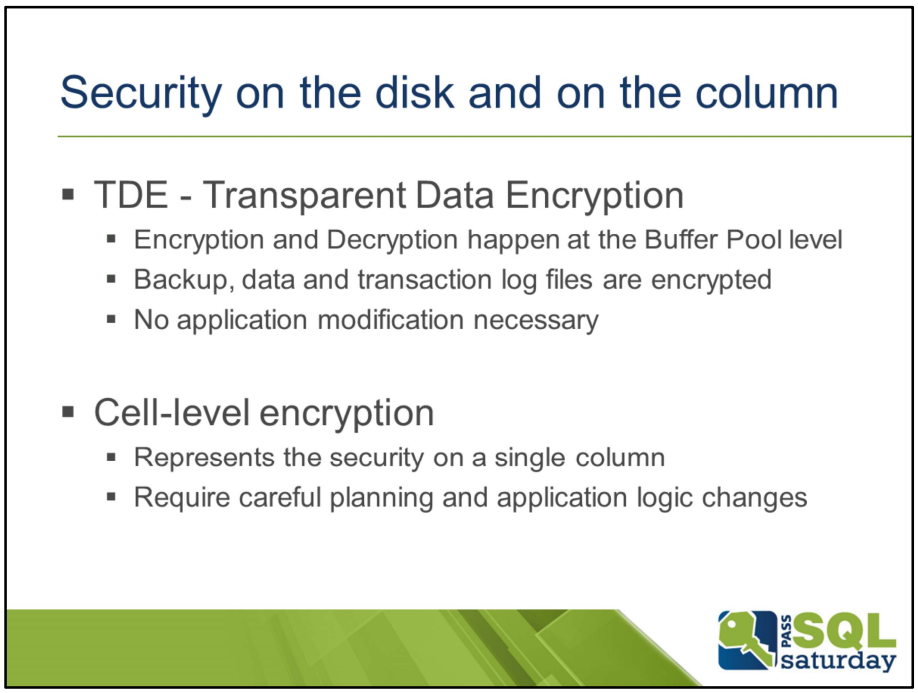
The certificate must be in either the local computer certificate store or the current user certificate store.

The current system time must be after the Valid from property of the certificate and before the Valid to property of the certificate.

The certificate must be meant for server authentication. This requires the Enhanced Key Usage property of the certificate to specify Server Authentication (1.3.6.1.5.5.7.3.1).

The certificate must be created by using the KeySpec option of AT_KEYEXCHANGE. Usually, the certificate's key usage property (KEY_USAGE) will also include key encipherment (CERT_KEY_ENCIPHERMENT_KEY_USAGE).

The Subject property of the certificate must indicate that the common name (CN) is the same as the host name or fully qualified domain name (FQDN) of the server computer. If SQL Server is running on a failover cluster, the common name must match the host name or FQDN of the virtual server and the certificates must be provisioned on all nodes in the failover cluster.



Transparent Data Encryption (TDE)
http://msdn.microsoft.com/en-us/library/bb934049.aspx

Move a TDE Protected Database to Another SQL Server
http://msdn.microsoft.com/en-us/library/ff773063(v=sql.110).aspx

Implementing Row- and Cell-Level Security in Classified Databases Using SQL Server 2005
http://technet.microsoft.com/en-us/library/cc966395.aspx#XSLTsection132121120120

CREATE VIEW UserTable AS SELECT ID, DecryptByKey(SensitiveData), CONVERT(money, CONVERT(varchar(50), DecryptByKey(SensitiveMoneyData))), NonSensitiveData
FROM tblBaseTable (READCOMMITTED), vwVisibleLabel
WHERE tblBaseTable.RLSMappingID = vwVisibleLabel.ID GO

Demo instructions:

Step 1:

Open SQL Server Management Studio and click new query

Create the Master Encryption Key and Certificate in the master database.

```
USE master
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Mke21%&@;/'
GO
CREATE CERTIFICATE Adventureworks2014Cert
WITH SUBJECT = 'DEK Certificate to encrypt Adventureworks2014'
GO
```

Step 2:

Open a new query window and execute this to create the Database Encryption Key in Adventureworks2014 database with the AES_128 algorithm

```
USE [Adventureworks2014]
GO
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256 ENCRYPTION BY
SERVER CERTIFICATE Adventureworks2014Cert
GO

--backup log [Adventureworks2014] to disk = 'c:\temp\adv.trn'

ALTER DATABASE [Adventureworks2014] SET ENCRYPTION ON
GO
```

Step 3:

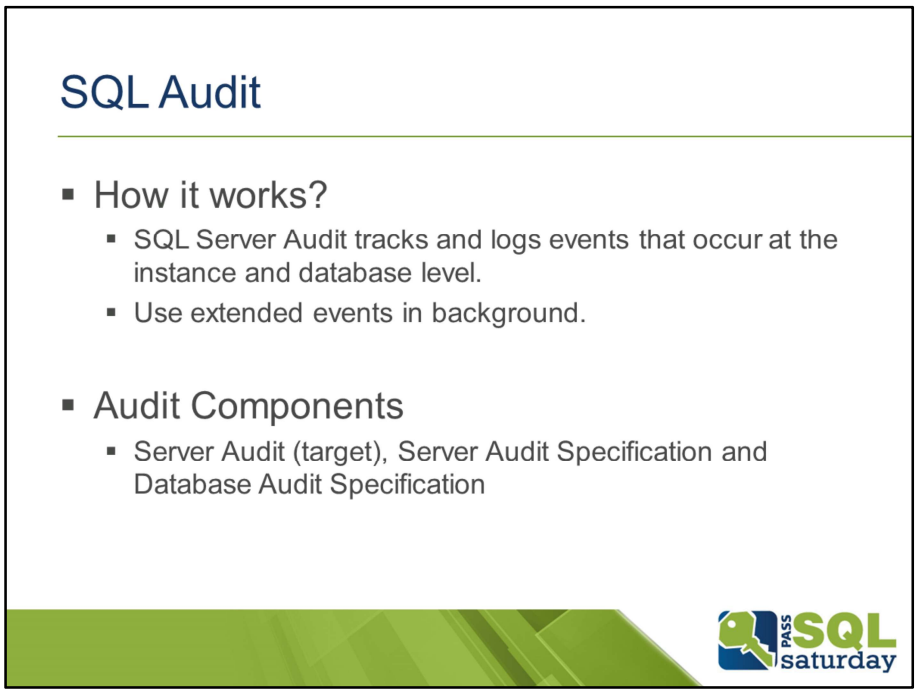
Execute this query to create a backup of the Certificate [Adventureworks2014Cert] we created to a file with a private key.

```
USE master
GO
BACKUP CERTIFICATE Adventureworks2014Cert
TO FILE = 'c:\temp\Adventureworks2014_cert.cer'
WITH PRIVATE KEY (FILE = 'c:\temp\Adventureworks2014_pk.pvk' ,
ENCRYPTION BY PASSWORD = 'Mke21%&@;/' )
GO
```

Step 4:

Type this into a New Query window and execute to create a backup of the encrypted database.

```
USE master
GO
BACKUP DATABASE [Adventureworks2014]
TO DISK = N'c:\Adventureworks2014.bak'
GO


/* The value 3 represents an encrypted state
   on the database and transaction logs. */
SELECT *
FROM sys.dm_database_encryption_keys
WHERE encryption_state = 3;
GO


ALTER DATABASE [Adventureworks2014] SET ENCRYPTION OFF
GO

USE Adventureworks2014
GO
DROP DATABASE ENCRYPTION KEY;
GO

USE [master]
GO
DROP CERTIFICATE [Adventureworks2014Cert]
GO

USE [master]
DROP MASTER KEY;
GO
```



SQL Server Audit (Database Engine)
http://msdn.microsoft.com/en-us/library/cc280386.aspx

SQL Server Audit Action Groups and Actions
http://msdn.microsoft.com/en-us/library/cc280663(v=sql.110).aspx

Demo instructions:

```
/* cleanup*/
USE [AdventureWorks2014]
GO
ALTER DATABASE AUDIT SPECIFICATION [Audit-Database-Lab]
WITH (STATE = OFF)
GO
USE [AdventureWorks2014]
GO
DROP DATABASE AUDIT SPECIFICATION [Audit-Database-Lab]
GO
```

```
USE [master]
GO
ALTER SERVER AUDIT SPECIFICATION [ServerAuditSpecificationlab]
WITH (STATE = OFF)
GO
USE [master]
GO
DROP SERVER AUDIT SPECIFICATION [ServerAuditSpecificationlab]
GO


USE [master]
GO
ALTER SERVER AUDIT [Audit-Server-Lab]
WITH (STATE = OFF)
GO
USE [master]
GO
DROP SERVER AUDIT [Audit-Server-Lab]
GO
```

Demo instructions:

Setup:

```
-------------------------------------------------------------------------------------------------------
--Server Audit
-------------------------------------------------------------------------------------------------------
GO
USE [master]
GO
------------------------------------------------------------
--ALTER SERVER AUDIT [Audit-Server-Lab] WITH (STATE = OFF)
--GO
--DROP SERVER AUDIT [Audit-Server-Lab]
--GO
------------------------------------------------------------

 IF NOT EXISTS
```

```
(SELECT 1 FROM sys.dm_server_audit_status
WHERE NAME = 'Audit-Server-Lab'  )

CREATE SERVER AUDIT [Audit-Server-Lab]
TO FILE
(FILEPATH = N'C:\temp\' ---> cria arquivo de auditoria neste caminho
,MAXSIZE = 15 MB --> tamanho maximo do arquivo
,MAX_ROLLOVER_FILES = 2 --. numero maximo de arquivos a serem retidos no
sistema
,RESERVE_DISK_SPACE = OFF --> pre aloca um tamanho de arquivo no disco.
)
WITH
(QUEUE_DELAY = 1000 --> valor minimo de atraso da auditoria... Sincrono = 0. No
caso espera até 1000 milesegundos para começar.
,ON_FAILURE = CONTINUE ---> indica se instancia deve parar ou continuar se houver
falha na auditoria.
)
ALTER SERVER AUDIT [Audit-Server-Lab] WITH (STATE = on)
GO

USE [master]
GO

CREATE SERVER AUDIT SPECIFICATION [ServerAuditSpecificationlab]
FOR SERVER AUDIT [Audit-Server-Lab]
ADD (SERVER_ROLE_MEMBER_CHANGE_GROUP),
ADD (SERVER_PRINCIPAL_CHANGE_GROUP)
GO

ALTER SERVER AUDIT SPECIFICATION [ServerAuditSpecificationlab] WITH (STATE=on)
go

------------------------------------------------------------------------------------------------------------
--Cria Tabela e auditoria no banco de dados
------------------------------------------------------------------------------------------------------------
USE [AdventureWorks2014] --> banco de dados o qual vai ser auditado.
GO
-----------------------------------------------------------
IF  NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[LAB_TABELA]') AND type in (N'U'))
DROP TABLE [dbo].[LAB_TABELA]
```

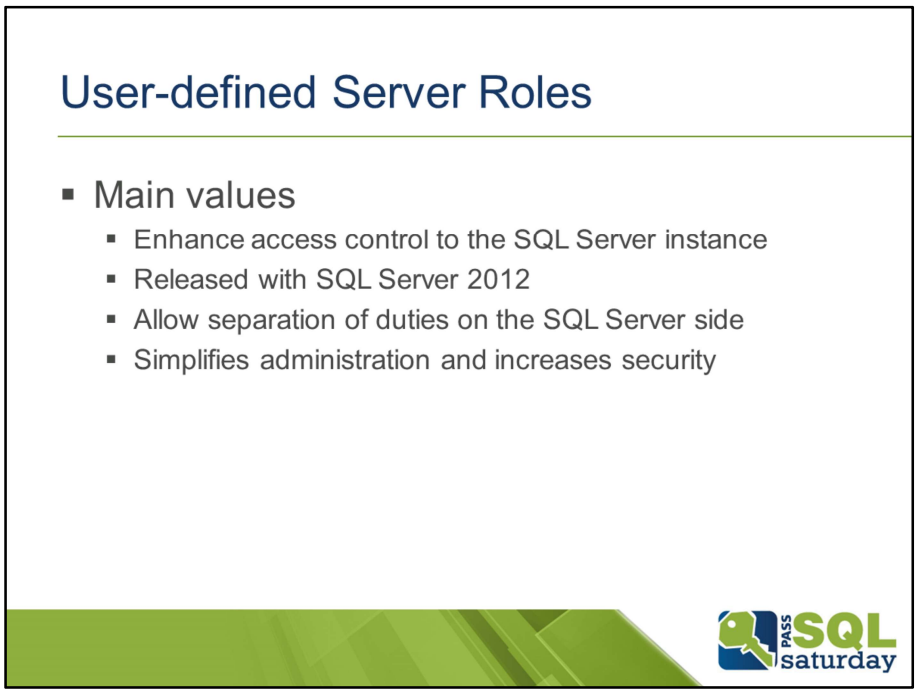
```
------------------------------------------------------------
CREATE TABLE [dbo].[LAB_TABELA]
(
[COLUNA] [varchar](250) NULL,
--[COLUNA_TESTE] [varbinary](256) NULL,
[COLUNA_INTEIRO] [int] identity (1,1)
) ON [PRIMARY]

-------------------------------------------------------------------------------------------------------
--Enable Audit
-------------------------------------------------------------------------------------------------------

 IF NOT EXISTS
(SELECT 1 FROM  sys.database_audit_specifications
 WHERE NAME = 'Audit-Database-Lab'  )

CREATE DATABASE AUDIT SPECIFICATION [Audit-Database-Lab]
FOR SERVER AUDIT [Audit-Server-Lab]
ADD (DELETE ON OBJECT::dbo.[LAB_TABELA] BY public),
ADD (INSERT ON OBJECT::dbo.[LAB_TABELA] BY public ),
ADD (UPDATE ON OBJECT::dbo.[LAB_TABELA] BY public ),
ADD (SCHEMA_OBJECT_CHANGE_GROUP)
WITH (STATE = ON)
```

Audit execution:

```
-------------------------------------------------------------------------------------------------------
USE [AdventureWorks2014]
GO
-------------------------------------------------------------------------------------------------------
-

DECLARE @CONT INT

SET @CONT = 0

WHILE @CONT <> 50000
BEGIN

INSERT INTO [dbo].[LAB_TABELA]
VALUES ('COLUNA TESTE')
```

```
IF @CONT =100 OR @CONT =1000 OR @CONT =2000
BEGIN
UPDATE [dbo].[LAB_TABELA]
SET [COLUNA] = 'COLUNA TESTE 100'
WHERE [COLUNA_INTEIRO] = @CONT -1
END

SET @CONT = @CONT + 1
END
GO

SELECT TOP 1000 * FROM [LAB_TABELA]

UPDATE [dbo].[LAB_TABELA]
SET [COLUNA] = 'COLUNA TESTE 100'
WHERE [COLUNA_INTEIRO] = 1

alter table LAB_TABELA drop column COLUNA_INTEIRO


alter table LAB_TABELA add COLUNA_INTEIRO [int] identity (1,1)


-----------------------------------------------------------------------------------------------------------
--verifica os dados inseridos no arquivo de auditoria

SELECT
DATEADD(hh, DATEDIFF(hh, GETUTCDATE(), CURRENT_TIMESTAMP),    event_time  )
as corrected_time
, event_time
,action_id
,session_server_principal_name AS UserName
,server_instance_name
,database_name
,schema_name
,object_name
,statement
, server_principal_name
FROM sys.fn_get_audit_file('C:\temp\*.sqlaudit', DEFAULT, DEFAULT)
WHERE action_id IN ( 'AL', 'SL', 'IN', 'DR', 'LGIF' , '%AU%' )
```

```
/* Lista a sigla e a descrição dos action_id de cada operação no SQL Audit */
Select DISTINCT action_id,name,class_desc,parent_class_desc from
sys.dm_audit_actions
```



New User-defined Server Roles in SQL Server 2012
http://www.databasejournal.com/features/mssql/new-user-defined-server-roles-in-sql-server-2012.html

Credits for Adreas Wolter

CONTROL SERVER vs. sysadmin
http://www.insidesql.org/blogs/andreaswolter/2013/08/control-server-vs-sysadmin-sa-permissions-privilege-escalation-caveats

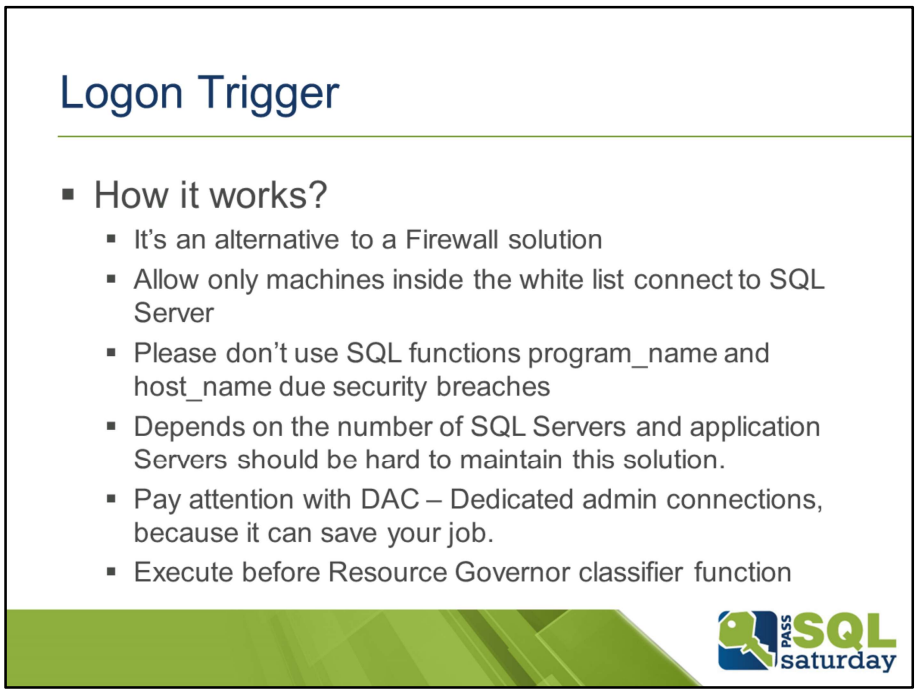
New Permissions in SQL Server 2014
http://www.insidesql.org/blogs/andreaswolter/2014/04/new-permissions-sql-server-2014

--- Privilege escalation demo
--- cleanup 1

```
ALTER SERVER ROLE [Role_DBA_Level2] DROP MEMBER [DBA_Level2]

DROP SERVER ROLE [Role_DBA_Level2]

--- cleanup 2

grant IMPERSONATE ANY LOGIN TO [Role_DBA_Level2];

grant SELECT ALL USER SECURABLES to [Role_DBA_Level2];

DROP LOGIN DBA_Level2;

USE [master]
GO

CREATE LOGIN DBA_Level2 WITH PASSWORD=N'teste123/',
DEFAULT_DATABASE=[master], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO

CREATE SERVER ROLE [Role_DBA_Level2];

ALTER SERVER ROLE [Role_DBA_Level2]ADD MEMBER DBA_Level2;

GRANT CONTROL SERVER TO [Role_DBA_Level2];

--- Log onto the SQL Server using SSMS as DBA_Level2 and check all permissions

EXECUTE AS LOGIN = 'DBA_Level2'
SELECT entity_name, permission_name
FROM sys.fn_my_permissions(NULL, NULL)entity;

-- Let’s compare it to the list of all permissions which exist on server-scope:

SELECT class_desc COLLATE Latin1_General_CI_AI, permission_name COLLATE
Latin1_General_CI_AI
FROM sys.fn_builtin_permissions('SERVER')EXCEPT SELECT entity_name,
permission_name FROM sys.fn_my_permissions(NULL, NULL)

--- Checking DBCC and DBO permissions.

DBCC CHECKDB;
```

```
DBCC LOGINFO;
DBCC DROPCLEANBUFFERS;

DBCC FREEPROCCACHE;

exec sp_readerrorlog
exec xp_fixeddrives

--- Let's compare with sysadmin. Log onto the SQL Server using SSMS with sa account

DBCC CHECKDB;
DBCC LOGINFO;
DBCC DROPCLEANBUFFERS;

DBCC FREEPROCCACHE;

exec sp_readerrorlog
exec xp_fixeddrives

select * from AdventureWorks2014.dbo.DatabaseLog

use master
go
deny SELECT ALL USER SECURABLES to [Role_DBA_Level2];

--- Log onto the SQL Server using SSMS with DBA_level2 account

select * from AdventureWorks2014.dbo.DatabaseLog

--- Privilege Escalation risk - Log onto the SQL Server using SSMS with DBA_Level2

ALTER SERVER ROLE sysadmin ADD MEMBER DBA_Level2;

EXECUTE AS LOGIN = 'sa';
ALTER SERVER ROLE sysadmin ADD MEMBER DBA_Level2;
ALTER SERVER ROLE sysadmin DROP MEMBER DBA_Level2;

--VIEW SERVER ROLES MEMBERS
SELECT sRole.name AS [Server Role Name] , sPrinc.name AS [Members]
FROM sys.server_role_members AS sRo
JOIN sys.server_principals AS sPrinc
```

```
ON sRo.member_principal_id = sPrinc.principal_id
JOIN sys.server_principals AS sRole
ON sRo.role_principal_id = sRole.principal_id;

--- Why?
--- Logins with CONTROL SERVER permission can impersonate EVERY Login
--- Log onto the SQL Server using SSMS with sa account

DENY IMPERSONATE ANY LOGIN TO [Role_DBA_Level2];

DENY ALTER ANY LOGIN TO [Role_DBA_Level2];

DENY SHUTDOWN TO [Role_DBA_Level2];
```



Logon Triggers
https://msdn.microsoft.com/en-us/library/bb326598(v=sql.120).aspx

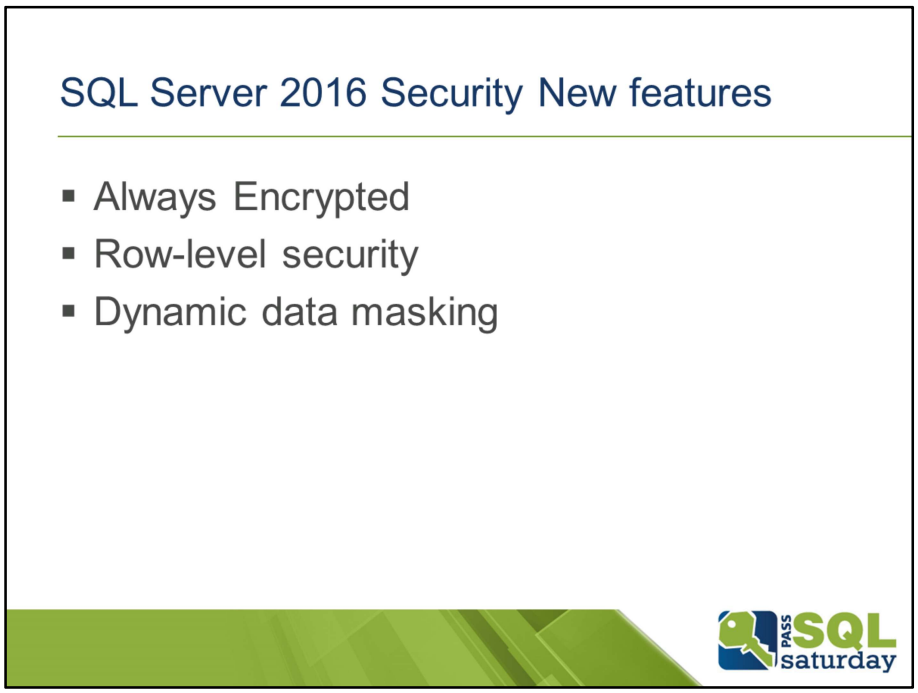
CREATE TRIGGER (Transact-SQL)
https://msdn.microsoft.com/en-us/library/ms189799(v=sql.120).aspx

DDL Triggers and LOGON Triggers
http://blogs.technet.com/b/vipulshah/archive/2007/12/04/ddl-triggers-and-logon-triggers.aspx

```
-- Demo script:
-- Execute using a sysadmin login

USE [master]
GO

CREATE DATABASE DBADMIN;
```

```
create table dbo.audit_connections ([session_id] varchar(50),[login_name]
varchar(100),[program_name] varchar (200),[host_name] varchar (200),[login_time]
datetime,[client_net_address] varchar(100),[client_tcp_port] varchar(50));

Create table WhiteListBD ([IpAddress] Varchar(100) null, Data_Liberacao DateTime
null Default GetDate());

-- Insert the ip addresses from the application server, dba machines, etc
Insert into WhiteListBD values ('insert ip address here', Getdate());

CREATE LOGIN [SecurityLogon] WITH PASSWORD=N'xxx',
DEFAULT_DATABASE=[DBADMIN], DEFAULT_LANGUAGE=[us_english],
CHECK_EXPIRATION=OFF, CHECK_POLICY=ON
GO

ALTER SERVER ROLE [sysadmin] ADD MEMBER [SecurityLogon]
GO

create trigger tr_audit_logon on all server
with execute as 'SecurityLogon'
for logon as

begin

Declare @IpAddress xml
Set @IpAddress = eventdata() ;


If Not Exists (select [IpAddress] from DBADmin.dbo.WhiteListBD
where @IpAddress.value('(/EVENT_INSTANCE/ClientHost)[1]','varchar(100)') =
IpAddress )
begin


        rollback

        insert into DBADmin.dbo.audit_connections
        select s.session_id,
        ORIGINAL_LOGIN() --s.login_name
        ,s.program_name,s.host_name,s.login_time,c.client_net_address,
```

```
        c.client_tcp_port
        from sys.dm_exec_sessions s join sys.dm_exec_connections c on s.session_id
= c.session_id
        where s.session_id = @@spid
        raiserror('Conexoes externas, que não sejam provenientes dos servidores de
aplicação e do time de banco de dados, não são permitidas.',10,15) with Log
end
end

go

disable trigger all on all server;

-- Demo part 2:

-- Inform this values during connection using SSMS (Click Options; Additional Connection
Parameters): Workstation ID=Fakemachine; Application Name=Spyapp

-- Execute this query and check the fields program_name and host_name
select ses.session_id,ses.host_name, ses.program_name, ses.host_process_id,
con.client_net_address
from sys.dm_exec_sessions ses inner join sys.dm_exec_connections con
  on ses.session_id = con.session_id
```

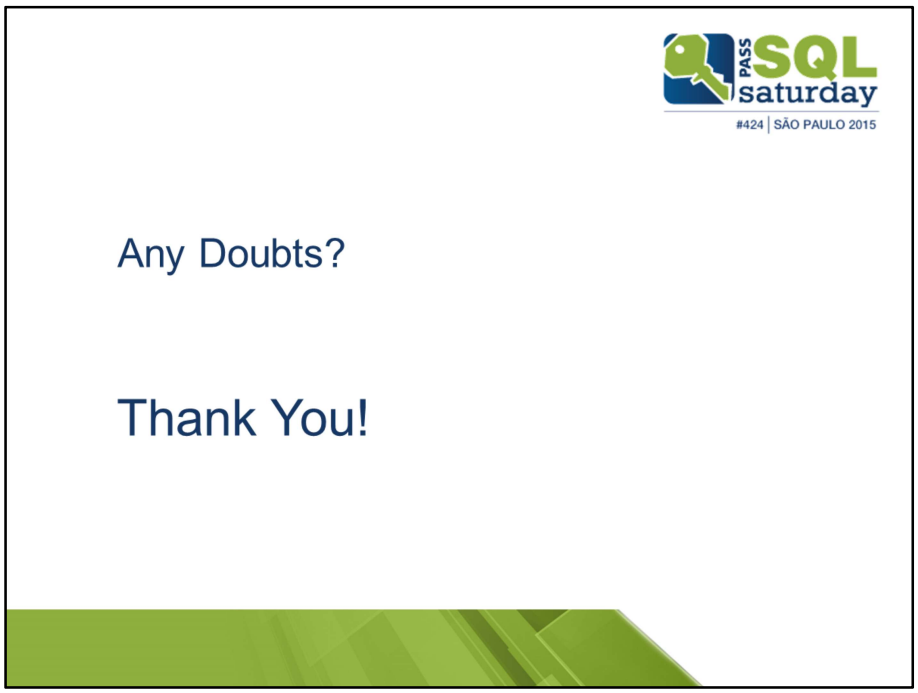


Overview and Roadmap for Microsoft SQL Server Security
https://channel9.msdn.com/Events/Ignite/2015/BRK2570

Always Encrypted
https://msdn.microsoft.com/en-us/library/mt163865.aspx

Row-Level Security
https://msdn.microsoft.com/en-us/library/dn765131.aspx

Dynamic Data Masking
https://msdn.microsoft.com/en-us/library/mt130841.aspx


PASS SQL saturday
#424 | SÃO PAULO 2015
Any Doubts?
Thank You!