As políticas estudadas têm diversas interseções, mas e preciso notar que a ISO 27001 como as outras baseadas nos padrões ISO como a ISO 9001, é centrada em documentação para o aprimoramento dos seus controles do SGSI. A PCI-DSS tem um núcleo de proteção pratica de dados sensíveis durante todo seu ciclo de consumo, especialmente voltada a guarda de informações como números de cartão de crédito. A HIPAA por sua vez tem controles abrangentes em torno da estrutura de dados que ela designa como ePHI, e por essa abrangência e proximidade dela com esse conjunto de dados, ela foi escolhida para ser usada como base dos controles comuns para uma abordagem do ponto de vista do banco de dados.

Detalhamento de implementação de alguns dos controles

Identity Framework [104]

Identity Framework [104] é um sistema de código aberto para identificação de usuários que permite adicionar funcionalidades de identificação e controle de permissões a um aplicativo web. Ela é recomendada para ser usada como padrão na identificação de usuários de aplicativos criados no .NET Framework +4 ou do .NET Core +2.x (que é multi-plataforma). Os recursos relevantes para nosso modelo de controles que essa plataforma plugável adiciona "out-of-the-box":

- Cria automaticamente as tabelas de login no banco para armazenamento das credenciais (7 tabelas mais uma de controle de modificações)
- Armazenamento seguro de senhas no banco:
    - **ASP.NET Identity Version 2**: *PBKDF2 with HMAC-SHA1, 128-bit salt, 256-bit subkey, 1000 iterations*
    - **ASP.NET Core Identity Version 3**: *PBKDF2 with HMAC-SHA256, 128-bit salt, 256-bit subkey, 10000 iterations*
- Requisitos ajustáveis de complexidade de senha
- Requisitos ajustáveis de número máximo de tentativas de logon
- Cookies de sessão protegidos de alteração
- Autenticação de dois fatores implementável com provedores SMS e Email [105]

Full over the wire encryption [105,106]

"Full over the wire encryption" é a ideia de proteger os dados de ponta-a-ponta em todo seu trajeto de consumo isso exige que o canal entre o servidor web e o servidor de dados e o canal entre o servidor web e o cliente sejam encriptados com criptografia forte.

Canal servidor web – browser cliente [105]

Para o canal servidor web para clientes web [105] rodando IIS+7 e Windows +10 respectivamente essa chave é negociada na ordem da seguinte tabela:

Cipher suite string	Allowed by SCH_USE_STRONG_CRYPTO	TLS/SSL Protocol versions
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	Yes	TLS 1.2
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	Yes	TLS 1.2
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	Yes	TLS 1.2
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	Yes	TLS 1.2
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA*	Yes	TLS 1.2, TLS 1.1, TLS 1.0

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA*	Yes	TLS 1.2, TLS 1.1, TLS 1.0

Tabela 501: Prioridade de cifras da versão de lançamento do Windows 10

Note que a menor cifra negociada sem “faal-back” para TLS1.0 nesse cenário é a TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, que deve ser tomada como padrão para os controles da aplicação.

Canal Servidor web – servidor de dados [106]

Para o canal servidor web para servidor de dados [106] rodando IIS +7 e SQL Server +2016 o protocolo TLS1.2 para conexões está disponível nativamente, para o SQL Server versões 2008 a 2014 ele está disponível via atualizações de segurança. Entendendo que nosso ambiente tem controles de segurança específicos que exigem um ambiente atualizado, vamos assumir ele atualizado, portanto pronto para uso exclusivo do TLS1.2.

Ele é configurado no servidor SQL Server atribuindo um certificado a instância emitido por uma CA (ele pode ser auto-emitido pelo próprio SQL Server, mas isso permite ataques man-in-the-middle) e habilitando a flag “ForceEncryption”[108] toda comunicação com o banco que não seja sobre TLS1.2 vai ser rejeitada.

No aplicativo, apartir da versão .NET Framework +4.6 (antigo .NET ambiente windows somente) ele é padrão, no novo .NET Core (multi-plataforma, open-source) também, então o aplicativo sendo construído em .NET Core 2.x vai negociar a cifra forte por padrão uma vez habilitada no banco, e vai conectar de maneira transparente se as anteriores forem desativadas.

Conexão direta administrativa de emergência [107]

Um controle essencial é a conexão direta administrativa [107], mais do que somente uma conexão normal ao servidor, ela roda em espaços de CPU e memória específicos do servidor, portanto está sempre disponível independente da carga no mesmo, é única (só pode haver uma ativa) e permite qualquer tarefa administrativa.

Criptografia e decriptografia de dados [108,109,110]

Existem vários meios de implementá-las, mas vamos tratar aqui de duas: Always Encrypted e Transparent Data Encription

Always Encrypted [108]

Always encrypted é um recurso do banco que permite manter determinadas colunas criptografadas permanentemente, e só as descriptografar no aplicativo, fazendo todo o transito desses dados sob proteção, esse recurso antes do SQL Server 2016 era restrito a versão enterprise, mas depois do SP1 do SQL Server 2016 ela se tornou aberta as demais edições. O aplicativo cliente (um aplicativo web por exemplo) deve ter o driver preparado para processar e desencriptar esses dados e acesso a chave de descriptografia. O banco de dados guarda metadados sobre as colunas e os dados criptografados, nunca a chave de criptografia chega ao mecanismo de dados o que dá segurança a esse recurso mesmo em cenários compartilhados, ou de nuvem.

Dois modos de operação estão disponíveis:

- Criptografia determinística – Sempre gera a mesma saída para dados criptografados desde que a entrada seja a mesma, esse tipo de criptografia permite criação de índices nos dados, assim como join e group by nessas colunas, mas pode permitir "adivinhar" o valor protegido por análise de padrão na tabela
- Criptografia randomizada – Sempre gera valores diferentes para entradas semelhantes, o que torna a análise substancialmente mais difícil, mas como contrapartida esse método impede a busca, agrupamento, indexamento e joins nessas colunas

Como funciona o algoritmo

O recurso Always Encrypted usa o algoritmo AEAD_AES_256_CBC_HMAC_SHA_256 para criptografar os dados.

O algoritmo AEAD_AES_256_CBC_HMAC_SHA_256 é derivado da especificação http://tools.ietf.org/html/draft-mcgrew-aead-aes-cbc-hmac-sha2-05. Ele usa um esquema de Criptografia Autenticada com Dados Associados, seguindo uma abordagem Encrypt-then-MAC. Ou seja, o texto original é primeiro criptografado e o MAC é produzido com base no texto cifrado resultante.

Para atender padrões, o AEAD_AES_256_CBC_HMAC_SHA_256 usa o modo de operação CBC, onde um valor inicial é alimentado no sistema (chamado vetor de inicialização (IV)). A descrição completa do modo CBC pode ser encontrada em http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

Transparent Data Encription [109]

O recurso Transparent Data Encryption fornece proteção a arquivos de dados num modelo chamado "encrypting data at rest". Nesse modelo é levado em consideração o roubo ou subtração física dos discos de dados do servidor, nesse recurso, os arquivos físicos de dados (dados e log) são completamente encriptados, a chave simétrica e guardado num setor especifico de boot do banco e protegidas por um certificado digital guardado no banco de sistema ou num módulo EKM.

O TDE faz a criptografia e a descriptografia dos dados em tempo real. Esse modo de operação permite atender a diversas regulamentações internacionais e boas práticas de vários setores da indústria ao mesmo tempo que permite aos desenvolvedores usar proteção AES ou 3DES nos dados sem ter de alterar de nenhum modo as aplicações existentes, sendo que essa criptografia é transparente para o aplicativo.

Os formatos de operação para esse modo são o AES-128, AES-192 e o AES-256 (a partir do SQL Server 2016 o 3DES foi tornado obsoleto).

Consumo no aplicativo web dos ambientes criptografados [110]

No aplicativo web o Always Encripted depende do driver SQL Server estar disponível (qualquer driver superior ao .NET 4.6 suporta, tão bem como as posteriores .NET Core 1 e 2) e a maneira de habilitá-la na string de conexão:

```
string connectionString = "Data Source=server63; Initial Catalog=Clinic;
Integrated Security=true; Column Encryption Setting=enabled";
```

O aplicativo também precisa ter acesso as chaves de descriptografia para poder usar o recurso. A referência completa está em [108]

No caso do TDE, como esse recurso usa criptografia transparente, o aplicativo não precisa de alteração alguma para utiliá-lo, cabendo ao banco fazer a mudança:

```
-- Create the database encryption key that will be used for TDE.
USE AdventureWorks2012 ;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM  = AES_128
ENCRYPTION BY SERVER ASYMMETRIC KEY ekm_login_key ;
GO

-- Alter the database to enable transparent data encryption.
ALTER DATABASE AdventureWorks2012
SET ENCRYPTION ON ;
GO
```

Nesse exemplo, o banco está sendo habilitado a usar criptografia AES-128, com um provedor de chaves EKM externo utilizado via uma conta de login pré configurada. O exemplo completo de código está na referência [109].

Auditoria [113]

É necessário estabelecer os grupos de auditoria mínimos exigidos pela norma, eles foram organizados nas seguintes tabelas e foram mantidos nomes quando aplicável que permitem sua aplicação em outras plataformas DBMS.

Auditoria de nível de servidor

ALTER CERTIFICATE	ALTER CREDENTIAL	ALTER LOGIN	ALTER SERVER ROLE
CREATE CREDENTIAL	CREATE LOGIN	CREATE SERVER ROLE	DROP CREDENTIAL
DROP LOGIN	DROP SERVER ROLE	AUDIT LOGIN	AUDIT LOGIN FAILED
AUDIT LOGOUT	DENY SERVER PERMISSIONS	GRANT SERVER PERMISSIONS	REVOKE SERVER PERMISSIONS

Auditoria de nível de banco de dados

ALTER ASYMMETRIC KEY	ALTER AUTHORIZATION	ALTER DDL TRIGGER	ALTER DML TRIGGER
ALTER MASTER KEY	ALTER PROCEDURE	ALTER ROLE	ALTER SERVICE MASTER KEY
ALTER SYMMETRIC KEY	ALTER USER	CREATE ASYMMETRIC KEY	CREATE CERTIFICATE
CREATE DML TRIGGER	CREATE MASTER KEY	CREATE PROCEDURE	CREATE ROLE
CREATE SYMMETRIC KEY	CREATE USER	DROP SYMMETRIC KEY	DROP CERTIFICATE
DROP MASTER KEY	DROP ROLE	DROP SYMMETRIC KEY	DROP USER
DML INSERT	DML UPDATE	DML DELETE	SELECT INTO
CLOSE MASTER KEY	CLOSE SYMMETRIC KEY	DENY DATABASE PERMISSIONS	EXECUTE AS DATABASE PRINCIPAL
GRANT DATABASE	OPEN MASTER KEY	OPEN SYMMETRIC KEY	REVOKE DATABASE

PERMISSIONS			PERMISSIONS

A auditoria de tabelas que fazem parte do núcleo de dados sensíveis do ePHI também devem ter auditoria de criação, alteração e exclusão de dados, é possível usar triggers, estabelecer procedures de acesso que guardem essa informação ou usar recursos de versionamento automático de linhas como o change data capture na referência técnica [113].

Procedimentos seguros de instalação [114]

A instalação do ambiente de dados é um fator determinante no sucesso da implementação de controles posteriores, desde a documentação a escolha de contas de serviço, um ambiente instalado que passe na validação para entrar em produção pode se mostrar de difícil correção. Alguns dos pontos principais numa instalação segura são:

- Projeto abrangente – O projeto da instalação deve conter o detalhamento em cada nó (servidor) do meio físico empregado, de preferência não só do servidor de dados (OLTP), mas do servidores web e de relatórios (OLAP)
    - Hardware empregado (CPU, Memória) e sua redundância se houver cluster
        - Para Azure/Nuvem: Mesmo o ambiente de nuvem deve ser descrito em termos de seus núcleos virtuais e memória alocada, ainda que a modalidade seja a elástica
    - Discos utilizados e em que arranjo (por exemplo: RAID10, dividido fisicamente para dados, log e tempdb, sistema operacional e paginamento em RAID 5)
        - Para Azure/Nuvem: É comum pensar que discos virtuais se ajustam a carga e o problema no gargalo físico desaparece em ambientes assim, o que é um erro. Para ambientes sensíveis a desempenho ou de grande capacidade, um ajuste das ligação de discos virtuais e a aquisição de discos virtuais com canais dedicados é essencial
    - Presença de storage ou discos compartilhados e se é empregada em um cluster
    - Tipo de cluster se houver (failover, distribuir carga)
    - Virtualização e modelo de virtualização empregada (hypervisor, container, etc)
    - Descrição detalhada dos meios de interconexão entre os dispositivos, e qual padrão de criptografia será usado em cada um (dados-web, web-cliente, dados-bi)
    - Descrição detalhada dos meios e regime de backup, atentando que algumas das normas pedem backups geograficamente esparsos, o que em termos práticos geralmente se traduz numa redundância física de pelo menos 400Km entre as localidades de contingência
        - Para Azure: O ambiente SQL Server suporta tanto backups no Azure quanto ajuste de redundância geográfica, então é possível por exemplo fazer backups na nuvem e ajustar para que tenha 3 pontos de redundância global. Num cenário de banco rodando somente no Azure (com os controles de criptografia aqui descritos) seria possível ajustar essa redundância e garantir a continuidade da operação, mas ainda assim tanto para fins de simplicidade de conformidade como garantia adicional contra corrupção intencional

- Pensar a instalação do sistema operacional – A instalação de um banco de dados seguro começa na instalação do sistema operacional, desde o arranjo de arquivos de sistema e arquivo de paginação até configurações de política e chaves
    - Estabelecer uma política de auditoria do sistema operacional para logins e uso de privilégios
    - Estabelecer controles de criptografia fortes para comunicação e senhas
        - No Windows: Desabilitar negociação de todos protocolos exceto NTLMv2, Exigir canal seguro com chave forte para comunicação com qualquer cliente, Impedir o armazenamento de senhas de hash fraco no registro, estabelecer políticas de senha complexa
    - Criar e tornar seguros os arranjos de disco onde ficarão os componentes de banco, se forem mais de um
        - Em Windows: Exigir NTFS e estabelecer políticas fortes para as pastas de dados
        - Para SQL Server: A instalação do SQL Server já exige NTFS e restringe ao mínimo as permissões ACL nas mesmas
    - Ajustar os firewalls e detectores de intrusão para as características de acesso do servidor de dados, e somente a elas
    - Não instalar o mecanismo de dados em controladores de domínio, eles têm características de portas e acesso muito distintas, o que torna muito difícil sua convivência com quaisquer outros serviços de rede, exceto os que trivialmente são empregados junto ao AD (exemplos: DHCP, DNS)
- Instalação do mecanismo de dados – É muito comum as instalações padrão de produtos, desde hardware a software incluírem configurações padrão reprováveis do ponto de vista de segurança, de senhas máster padrão a portas abertas e serviços desnecessários a instalação é um passo crítico num ambiente seguro. No caso específico do SQL Server, desde a versão 2005 tem sido feito um esforço para tornar a experiência de instalação livre de falhas de configuração por padrão, algumas das configurações padrão endereçadas
    - Conta sysadmin – No SQL Server o modo padrão de autenticação é o integrado, a menos que explicitamente ativado a autenticação do SQL Server na instalação, isso permite manter a conta master inativa desde a instalação
    - Privilégios de contas de sistema – Um grande problema de serviços e bancos de dados em geral é a conta onde esses serviços rodam, em outras palavras, essa conta é uma conta de sistema operacional e dependendo dos seus privilégios, o comprometimento do mecanismo de dados pode levar ao comprometimento do servidor. Nas instalações do SQL Server apartir do 2008, as contas de serviço são criadas pela instalação com os privilégios mínimos necessários para sua operação, configurações complexas que teria de ser feitas a mão anteriormente:
        - Impedir logon interativo e de console (esse usuário não loga no sistema como usuários padrão)
        - Alocar e liberar páginas de memória (geralmente somente administradores tem esse privilégio, o que fazia muitos administradores de servidor colocar nestas contas o privilégio de administrador, para facilitar o processo)
    - Exigir que as pastas de dados estejam em formato NTFS para permitir controle de acesso

- Aplica os controles de acesso (ACL) recomendados aos seus arquivos binários e aos locais de dados
- Permite instalar os serviços de maneira granular, e não seleciona nenhum serviço por padrão (nem o do mecanismo de dados em si)
- Exige a instalação do .NET Framework 4.6 ou mais novo (o que já permite as chaves mais fortes de proteção de dados e comunicação estarem prontamente ativas)
- Torna obsoletos controles criptográficos a cada nova versão lançada, permitindo determinados controles somente por ativação posterior deliberada
- Protege configurações avançadas sensíveis e não as ativa por padrão
    - Por exemplo a interface de execução de comandos, existem procedures do SQL que permitem direcionar comandos de console ao sistema operacional, o procedimento de ativação dessas procedures não é trivial e nem por meio gráfico intencionalmente, além delas serem marcadas como obsoletas em versões futuras
- Phone home e telemetria [115] – Esse passo tem de ser dado com parcimônia, por um lado, o recurso de enviar crashs do mecanismo para a Microsoft é importante pois permite numa situação de emergência de disponibilidade um atendimento do suporte mais rápido, mas eventualmente estes crashs incluem informações dos procedimentos em trânsito durante o crash, o que pode significar trânsito de informação sensível para fora da empresa para um terceiro (a Microsoft no caso) se as informações estiverem protegidas por criptografia Always Encrypted nem o banco vai poder mandar os dados sigilosos mas com TDE como o banco faz a descriptografia transparente, isso seja possível
- Atualizações de vulnerabilidades – Por padrão a instalação checa versões mais novas não de versão maior (major) mas de atualizações cumulativas e service packs (CU,GDR,SP) e as instala durante a instalação do mecanismo se uma conexão de rede estiver disponível

1. [101] Triggers DML (Baseado em alteração de dados) e DDL (baseado em alteração de schema) https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql
2. [102] Procedimentos de banco compilados que permitem utilizar API's REST e SOAP via código C# e bibliotecas .NET de dentro do mecanismo de banco de dados https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-oltp/creating-natively-compiled-stored-procedures
3. [103] Detalhes de escolha entre autenticação do SQL Server por logins ou Autenticação integrada do windows utilizando serviços de diretório (Active Directory Services) https://docs.microsoft.com/en-us/sql/relational-databases/security/choose-an-authentication-mode
4. [104] Configuração do provedor Identity Framework para autenticação de usuário https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?tabs=visual-studio%2Caspnetcore2x
    a. Dados específicos de hash e salt de senha https://andrewlock.net/exploring-the-asp-net-core-identity-passwordhasher/
    b. Autenticação de dois fatores em .NET Core +2.x https://docs.microsoft.com/en-us/aspnet/core/security/authentication/2fa

    c. Configurar autenticação integrada do Windows no aplicativo .NET Core +2.x https://docs.microsoft.com/en-us/aspnet/core/security/authentication/windowsauth?tabs=aspnetcore2x
2. [105] Referência de cifras fortes negociáveis para uso em clientes Windows https://msdn.microsoft.com/en-gb/library/windows/desktop/aa374757(v=vs.85).aspx
    1. Tabela específica do cenário para IIS +7 e Windows +10 (v1057 de lançamento) https://msdn.microsoft.com/library/windows/desktop/mt767769.aspx
    2. Requerer SSL no nível de servidor https://support.microsoft.com/en-us/help/298805/how-to-enable-ssl-for-all-customers-who-interact-with-your-web-site-in
    3. Referência adicional para forçar uso de HTTPS no .NET Core +2.x https://docs.microsoft.com/en-us/aspnet/core/security/enforcing-ssl
3. [106] Habilitar conexões criptografadas com TLS1.2 no canal cliente – servidor de dados https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine
    1. Referência de versões para uso de TLS1.2 no canal servidor de dados – servidor web https://technet.microsoft.com/en-us/library/ms189067(v=sql.105).aspx
    2. Desabilitar protocolos adicionais para conexão (a criptografia mais forte será negociada, esse passo é para restringir a conexão e impedir fall-back para cifras mais fracas) https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/enable-or-disable-a-server-network-protocol
4. [107] Como conectar ao SQL Server usando DAC https://technet.microsoft.com/pt-br/library/ms178068(v=sql.105).aspx
5. [108] Always Encripted https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine
    1. Detalhes da criptografia e métodos https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-cryptography
6. [109] Transparent Data Encription https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption
    1. Habilitando o TDE com um módulo provedor de chaves (EKM) externo https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/enable-tde-on-sql-server-using-ekm
7. [110] Como usar o recurso Always Encrypted no aplicativo cliente https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/develop-using-always-encrypted-with-net-framework-data-provider
8. [111] Visão geral do padrão MVC https://msdn.microsoft.com/pt-br/library/dd381412(v=vs.108).aspx
9. [112] Visão geral do uso de LINQ para consultas ao banco de dados https://msdn.microsoft.com/pt-br/library/dd381412(v=vs.108).aspx
10. [113] Visão geral da auditoria do SQL Server https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine
    1. Grupos de auditoria do SQL Server https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-action-groups-and-actions

2. Recurso de versionamento de linhas change data capture https://docs.microsoft.com/en-us/sql/relational-databases/track-changes/about-change-data-capture-sql-server
11. [114] Instalação segura do SQL Server https://docs.microsoft.com/en-us/sql/sql-server/install/security-considerations-for-a-sql-server-installation
12. [115] Como ativar e desativar os recursos de phone home e telemetria https://support.microsoft.com/en-us/help/3153756/how-to-configure-sql-server-2016-to-send-feedback-to-microsoft