Conformidade de Sistemas Gerenciadores de Bancos de Dados com aplicações

Lenin Cristi, Lucas Zanoni de Oliveira, Otávio Sanchez, Carlos Frederico Teixeira, Helton Abrantes de Souza, Vanessa Marques Correa

CMCC – Universidade Federal do ABC (UFABC)
Santo André – SP – Brasil

{frederico.teixeira,lenin.cristi,lucas.zanoni,otavio.sanchez,
helton.abrantes,vanessa.marques}@aluno.ufabc.edu.br

Resumo. *O presente artigo trata dos controles de segurança aplicáveis a depósitos de dados e ao seu consumo seguro do ponto de vista de normas internacionais governamentais e de mercado com foco nas regulações PCI-DSS, ISO 27002 e HIPAA. Seu objetivo principal é consolidar esses dispositivos num grupo de controles comuns, oferecer uma visão histórica do fim de cada uma e finalmente mostrar meios de aplicação dos controles consolidados em um produto de mercado multi plataforma.*

1. Apresentação geral das normas utilizadas

O objetivo geral é mapear os controles de segurança aplicáveis a banco de dados nas normas ISO 27001, PCI-DSS 3.2 e HIPAA e estabelecer um conjunto comum de controles que permita atingir conformidade com as três.

1.1 PCI-DSS

O Payment Card Industry Security Standards Council (PCI-SSC) foi fundado pelas principais organizações que atuam com cartões de crédito, como a American Express, Discover Financial Services, JCB International, MasterCard Worldwide e Visa Inc., como um fórum global para a disseminação de padrões de segurança na proteção de dados de pagamento, nele é definido o PCI Data Secutity Standart (PCI-DSS), padrão criado para aumentar o controle sobre os dados do portador do cartão para que assim o número de fraudes com cartões de crédito, sejam reduzidas.

O principal objetivo do PCI-DSS é construir e manter uma rede segura através da qual irá ser conduzida as transações, protegendo sempre as informações do titular do cartão de fraudes ou possíveis ataques. Além disso, ele deve garantir sempre um acesso seguro à rede mantendo padrões de segurança.

Para que uma empresa obtenha a certificação do PCI-DSS, é necessário que ela possua uma auditoria e auxílio de um profissional ou empresa com certificação QSA (Qualified Security Assessor). A empresa que está em busca da certificação PCI-DSS deverá atingir todos os requisitos e processos estabelecidos pela norma dentro de um ciclo de 12 meses. Dentro destes requisitos estão instalação de antivírus, testes de segurança e vulnerabilidade, constantes scans para detecção de possíveis ameaças à segurança, além de restringir o acesso de informações contidas em um banco de dados.

Outras ações cabíveis e requeridas pelo PCI-DSS, para a proteção de informações sigilosas contidas em um banco de dados são:

- Todos as consultas, acessos e ações do usuário no banco de dados devem ocorrer através de métodos programáticos.
- O banco de dados só deve ser acessado para consulta, por um administrador do mesmo.
- Os IDs dos aplicativos do banco de dados, não podem ser usados por usuários individuais ou outros processos. Mas sim, por um aplicativo.

1.2 ISO IEC 27002

No ano de 1995, tivemos a criação de um grupo de normas que consolidam as diretrizes de segurança da informação, pelas organizações internacionais ISO (The International Organization for Standardization) e IEC (International Electrotechnical Commission), sendo representada pela série 27000. Nesse grupo de normas, temos a ISO/IEC 27002, norma internacional que estabelece um código de boas práticas para auxiliar a implementação do Sistema de Gestão de Segurança da Informação (SGSI). Essas normas, são importantes para a implementação do SGSI, em qualquer tipo de organização.

O objetivo da ISO 27002, é estabelecer diretrizes, ajudar na implementação, inicialização, gestão, além de melhorar a segurança da informação de uma organização.

Seus benefícios e fins práticos são:

- Melhoria na conscientização da segurança da informação;
- Maior controle de ativos e informações sensíveis da organização, além de ofercer uma abordagem para implementação de políticas de controle;
- Identificação e correção de pontos fracos e sensíveis;
- Maior redução de riscos de responsabilidade pela não implementação de um
- SGSI ou determinação de políticas e procedimentos;
- Melhor organização com processos e mecanismos bem desenhados e geridos;
- Redução de custos relacionados a incidentes de segurança de informação.
- Conformidade com a legislação e outras regulamentações e certificações;

1.3 HIPAA

A Lei de Portabilidade e Responsabilidade de Seguros de Saúde foi criada nos EUA no ano de 1996 (HIPAA). A partir desta lei, o departamento de Saúde e Serviços Humanos dos EUA (HHS) desenvolveu regulamentações para proteger a privacidade e segurança de certas informações de saúde.

Para cumprir esse requisito, o HHS publicou os documentos conhecidos como a Regra de Privacidade HIPAA e a Regra de Segurança HIPAA.

A Regra de Privacidade, ou Padrões para Privacidade de Informações de Saúde Individualmente Identificáveis, estabelece padrões nacionais para a proteção de certas informações de saúde.

Os Padrões de Segurança para a Proteção de Informações de Saúde Protegidas Eletrônicas (a Regra de Segurança) estabelecem um conjunto de padrões de segurança para proteger certas informações de saúde que são mantidas ou transferidas em formato eletrônico.

A Regra de Segurança operacionaliza as proteções contidas na Regra de Privacidade, abordando as salvaguardas técnicas e não técnicas que as organizações chamadas "entidades cobertas" devem colocar em prática para proteger as "informações de saúde protegidas eletrônicas" (e-PHI) dos indivíduos.

A regra de segurança exige que as entidades mantenham salvaguardas administrativas, técnicas e físicas razoáveis e apropriadas para proteger o e-PHI.

Especificamente, as entidades cobertas devem:

- Garantir a confidencialidade, integridade e disponibilidade de todos os e-PHI que eles criam, recebem, mantêm ou transmitem;
- Controlar e manter backup dos dados, seja no armazenado e / ou processado em sistemas de computador e bancos de dados.
- Criptografar os dados, enquanto em trânsito em redes não seguras.
- Garantir acessibilidade somente por pessoal com responsabilidades de trabalho para acessar o PHI
- Monitorar e controlar o acesso autorizado e não autorizado.
- Identificar e proteger contra ameaças razoavelmente antecipadas à segurança ou integridade das informações;
- Proteger contra usos ou divulgações razoavelmente antecipados e inadmissíveis;
- Garantir a conformidade de sua força de trabalho.

A regra de segurança define "confidencialidade" no sentido que o e-PHI não deve estar disponível ou ser divulgado a pessoas não autorizadas. Os requisitos de confidencialidade da regra de segurança apóiam as proibições da regra de privacidade contra usos impróprios e divulgações de PHI. A regra de segurança também promove os dois objetivos adicionais de manter a integridade e a disponibilidade do e-PHI.

Sob a regra de segurança, "integridade" significa que o e-PHI não é alterado ou destruído de maneira não autorizada. "Disponibilidade" significa que o e-PHI é acessível e utilizável sob demanda por uma pessoa autorizada.

O HHS reconhece que as entidades cobertas possuem características próprias, portanto, a regra de segurança é flexível para permitir que entidades cobertas analisem suas próprias necessidades e implementem soluções apropriadas para seus ambientes específicos. O que é apropriado para uma entidade dependerá da natureza dos negócios, bem como do tamanho e dos recursos.

Portanto, quando uma entidade coberta está decidindo quais medidas de segurança usar, a regra não dita essas medidas, mas exige que a entidade coberta considere:

- Seu tamanho, complexidade e capacidades,

- Sua infra-estrutura técnica, de hardware e de software,
- Os custos das medidas de segurança
- Probabilidade e possível impacto de riscos potenciais para o e-PHI.

2. Requerimentos consolidados para conformidade

As políticas estudadas têm diversas interseções, mas é preciso notar que a ISO 27002 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.

Tabela de controles comuns de SGSI para as normas ISO 2001, PCI-DSS 3.2 e HIPAA usando a HIPAA como base. Os item sombreados em verde são parcialmente ou totalmente atendidos pela implementação de outros controles, que estão indicados no campo “coberto”.

ISO 27001	PCI-DSS 3.2	HIPAA	Descrição	Coberto (total ou parcialmente) pela implementação do controle	Detalhes de implementação na camada do banco de dados	Detalhes de implementação na camada do aplicativo web	MS SQL Server 2016	.NET Core +2.0 (ASP.NET MVC C#)
		Controles basicos						
		§ 164.308 (a)(1)(i)	Exige implementação de politicas e procedimentos relacionados a detecção, prevenção e correção de violações de segurança	Audit Controls Standard – § 164 .312(b)				
		§164.308(a)(1)(ii)(D)	Exige que a solução implementada permita inspeção regular dos logs de auditoria, relatórios de acesso e	Audit Controls Standard – § 164 .312(b)				

			registro de incidentes					
A.7.1.3 Uso aceitável dos ativos	Implement Strong Access Control Measures Requirement 7: Restrict access to cardholder data by business need to know	§164.308(a)(3)(ii)(B)	Exige que os procedimentos implementados garantam que o acesso do funcionário ao ePHI armazenado é apropriado	Audit Controls Standard – § 164 .312(b)				
A.7.1.3 Uso aceitável dos ativos A.8.3.3 Retirada de direitos de acesso A.11.2.1 Registro de usuário	Implement Strong Access Control Measures Requirement 7: Restrict access to cardholder data by business need to know	§164.308(a)(3)(ii)(C)	Exige que a solução implementada garanta que todo acesso do funcionário ao ePHI, que está definido no parágrafo (a)(3)(II)(B), quando este deixa a organização seja revogado		Implementar triggers de banco de dados na tabela de funcionários do banco ERP para baixa automática dos logins relacionados a funcionários desligados **OU** Utilizar a API do ERP para baixa automática dos logins relacionados a funcionários desligados	n/a	Table triggers (quando o acesso direto ao DB do ERP é possível) [101] **OU** Compiled Stored Procedures (para acesso a API do ERP) [102]	

		§164.308 (a)(4)(i)	Todo acesso ao ePHI deve ser autorizado e todas as politicas e procedimentos devem ser consistentes com os requerimentos aplicaveis definidos na CFR 45 parte 164, subsessão E	Access Control Standard – §164 .312(a)(1)				
		§164.308 (a)(4)(ii)(A)	Uma empresa que é subordinada de um conglomerado maior deve implementar políticas e procedimentos para proteger o acesso aos ePHI da sua organização controladora	Access Control Standard – §164 .312(a)(1)				
		§164.308(a)(4)(ii)(C)	Implementar políticas e procedimentos criados de acordo com políticas de autorização de acesso, e permite controle completo dos direitos de usuário ao ePHI	Access Control Standard – §164 .312(a)(1)				

		§164.308(a)(5)(ii)(C)	Todas as tentativas de login devem sem registradas e procedimentos de relatório para detectar discrepâncias devem ser estabelecidos	Access Control Standard – §164 .312(a)(1)				
		§164.308(a)(5)(ii)(D)	Todos ao eventos de criação e troca de senhas devem ser auditados e revisados	Audit Controls Standard – § 164 .312(b)				
		§164.308(a)(6)(i)	Todos os incidentes de segurança devem ser registrados, revisados e endereçados	Audit Controls Standard – § 164 .312(b)				
A.5.1.1 Documento da política de segurança da informação A.5.1.2 Análise crítica da política de segurança da informação A.10.1.1 Documentação dos procedimentos de operação A.10.7.4		§164.308(a)(6)(ii)	Estabelecer procedimentos que permitam identificar e apropriadamente responder a suspeitos/conhecidos incidentes de segurança incluindo documentar/ revisar todos os incidentes e suas consequências	Audit Controls Standard – § 164 .312(b) Padrões de documentação física e eletrônica	Implementar documentação completa em meio fisico com acesso restrito **E** em formato digital em repósitório específico para políticas de segurança **REQUISITO** O meio físico deve ter acesso auditado e não conter senhas ou chaves E O		(Externo ao SQL Server) Guarda de documentação em depósito com controles específicos ou datacenter em cofre separado do cofre do meio de backup magnético se houver **E** File Server com permíssões ACL estritas	

Segurança da documentação dos sistemas					meio digital deve ser auditado por política de acesso específica		ao pessoal responsável pela operação do banco e aplicativo somente, com auditoria de acesso para leitura, criação, alteração e exclusão dos arquivos	
		§164.308(a)(7)(ii)(B)	Estabelecer e implementar procedimentos que permitam recuperar qualquer perda de informação do ePHI armazenado	Integrity Standard – § 164 .312(c)(1)				
		Controles principais						
		Controle de acesso						

A.10.4.1 Controle contra códigos maliciosos (ver implementação da camada do aplicativo) A.11.1.1 Política de controle de acesso A.11.2.3 Gerenciamento de senha do usuário A.11.2.4 Análise crítica dos direitos de acesso de usuário	Implement Strong Access Control Measures Requirement 8: Identify and authenticate access to system components Maintain a Vulnerability Management Program Requirement 6: Develop and maintain secure systems and applications (ver implementação da camada do aplicativo)	Access Control Standard – §164 .312(a)(1)	Política de identificação e autorização		Autorização do SQL Server: Papéis únicos por funcionário identificado unicamente com a tabela de funcionários do ERP **OU** Autorização integrada do Windows: Logins do AD em grupo auditado por GPO específico de acesso ao ePHI **REQUISITO** §164.308(a)(3)(ii)(C)	Implementar a autenticação de usuário baseada em forms ou integrada com AD **REQUISITO** Implementar salva de senhas de usuário em criptografia de mão única (SHA-512 mais SALT ou superior) **E** Implementar criptografia de dados de sessão (AES-256 ou superior) **E** Implementar dois fatores de segurança (certificado, sms ou token de aplicativo)	Windows Active Directory Authentication **OU** SQL Server Authentication and Authorization [103]	.NET Core Identity Framework (Este padrão de autenticação cobre todos os requisitos especificados em "plataforma") [104] **OU** Personificar o usuário de domínio do Windows na aplicação (AD sobre LAN ou VPN) [103]
A.11.2.2 Gerenciamento de privilégios	Implement Strong Access Control Measures Requirement 7: Restrict access to cardholder data by business need to know	Unique User Identification Specification – Required §164 .312(a)(2)(i)	Exige que todo login de acesso seja único para identificação e registro de identidade		Exigir a personificação do papel de usuário especifico do funcionário após a conexão com usuário padrão do aplicativo web com privilégio mínimo **OU** Implementar	Exigir a personificação do papel de usuário especifico do funcionário após a conexão com usuário padrão do aplicativo web com privilégio mínimo **OU** Implementar	Windows Active Directory Authentication **OU** SQL Server Authentication and Authorization **REQUISITO** implementar permissão mínima no usuário de	.NET Core Identity Framework **OU** Personificar o usuário de domínio do Windows na aplicação (AD sobre LAN ou VPN) **REQUISITO** O aplicativo deve

					autenticação integrada (AD)	autenticação integrada (AD)	conexão e permissões granulares nos usuários de funcionários	conectar ao banco com privilégio mínimo e personificar o usuário específico do funcionário para operar o ePHI
		Emergency Access Procedure Specification – Required §164 .312(a)(2)(ii)	Política de acesso de emergência		Implementar meio de acesso de emergência ao mecanismo de dados	n/a	Emergency Access Policy (Pode ser usado o DAC ou Console de acesso direto administrativo) **E** SQL Server Audit **REQUISITO** As contas com acesso de emergência devem ser minimas para permitir redundância (ex: 2) e estarem sujeitas a política de auditoria padrão	n/a
		Automatic Logoff Specification – Addressable § 164 .312(a)(2)(iii)	Exige logoff automático de usuário		Implementar expiração de sessão	Implementar expiração de sessão no aplicativo	Windows Active Directory Group Policy Enforcement	Configure Session Timeout **REQUISITO** Essa política deve ser implementada

								independente do meio de autenticação
A.6.2.3 Identificando segurança da informação nos acordos com terceiros* A.12.5.4 Vazamento de informações * (*Always Encripted e o TDE funcionam em ambientes e nuvem) A.9.1.1 Perímetro de segurança física** (**TDE fornece data-at-rest encryption)	Protect Cardholder Data Requirement 3: Protect stored cardholder data	Encryption and Decryption Specification – Addressable § 164 .312(a)(2)(iv)	Política de criptografia e decriptografia		Implementar criptografia de colunas com chave externa ao mecanismo de banco de dados **OPCIONAL** Usar criptografia transparente no banco inteiro	Configurar aplicativo para utilizar chave externa na conexão para possibilitar leitura das colunas criptografadas	Always Encrypted[108] para criptografia de ponta-a-ponta em registros sensíveis; TDE[109] Para criptografia transparente	Configurar Connection String do aplicativo para usar chave externa de decriptografia[108]
	Protect Cardholder Data Requirement 4: Encrypt transmission of cardholder data across open, public networks	Person or Entity Authentication Standard – § 164 .312(d)	Política de identificação	Access Control Standard – §164 .312(a)(1)	Usar somente TLS1.2	Usar somente TLS1.2	SQL Server Password Policy **OU** Domain Password Policy Enforcement **REQUISITO** Transport Layer Security	Configurar o servidor de aplicativo para recusar conexões que não satisfaçam requisitos mínimos de segurança [105]

							(TLS\SSL) Enforcement [106]	**REQUISITO** Transport Layer Security (TLS\SSL) Enforcement [105]
		Auditoria e conformidade						
A.10.10.1 Registros de auditoria A.10.10.2 Monitoramento do uso do sistema A.10.10.4 Registros (log) de administrador e operador A.10.10.5 Registros (logs) de falhas	Regularly Monitor and Test Networks Requirement 10: Track and monitor all access to network resources and cardholder data	Audit Controls Standard – § 164 .312(b)	Exige implementação de auditoria contínua do armazenamento de dados com mecanismos que registrem e reportem atividades nos armazens que contém ou usam dados do ePHI		Implementar auditoria e logins, alteração de privilégios e uso de dados **OPCIONAL** Habilitar a auditoria de alteração de registros	Implementar auditoria de uso de recursos e comportamentos não relacionados a dados (não cobertos pela auditoria de banco)	SQL Server Audit [112] **E** Policy-Based Management **OPCIONAL** Change Data Capture **OU** Table DML Triggers [101]	Entity Frramework Functions mapeando SQL Server Procedures criadas especificamente para auditoria
		Integridade de dados						

A.10.5.1 Cópias de segurança das informações A.10.4.1 Controle contra códigos maliciosos A.12.2.1 Validação dos dados de entrada A.12.2.2 Controle do processamento interno A.14.1.3 Desenvolvimento e implementação de planos de continuidade relativos à segurança da informação	Maintain a Vulnerability Management Program Requirement 6: Develop and maintain secure systems and applications	Integrity Standard – § 164 .312(c)(1)	Implementar auditoria contínua do armazenamento em todas as atividadesrelativas ao ePHI, para protegê-lo de tentativas impróprias de alteração ou deleção	Audit Controls Standard – § 164 .312(b)	Implementar auditoria de alteração de objetos e banco **E** Estabalecer designs consistentes de banco de dados aderentes a modelagem formal de R-DB's **E** Estabelecer baselines de performance e procedimentos de tunning **E** Estabelecer meios de backup automático encriptado e distribuído geograficamente	Isolar completamente a camada de interação do usuário da camada de acesso a dados por meio de modelos ou procedimentos de banco fortemente tipados	Database DDL Triggers **E** Database and Application Development Standards and Guidelines; Constraints, Triggers and Referential Integrity **E** Database Performance Collection **E** Database Backups **OU** Azure Encrypted Backup	Usar modelo Entity Framework **OU** Modelo Database-first/Code-First **REQUISITO** Usar modelos (classes) numa abordagem estrita MVC (Modelo-View-Controle) [111] **E** Usar somente LINQ [112] para todas as consultas (joins, searchs, order bys, etc) **OU** Encapsular as regras de negócio em procedimentos fortemente tipados e mapear em funções dentro do Entity Framework
A.9.2.3 Segurança do cabeamento A.12.2.3 Integridade de mensagens	Protect Cardholder Data Requirement 4: Encrypt transmission of cardholder data across open, public networks	Mechanism to authenticate ePHI Specification – Addressable § 164 .312(c)(2)	Mecanismo de autenticação		Criptografia de ponta-a-ponta para pacotes (Full over the wire encryption)	Criptografia de ponta-a-ponta para pacotes (Full over the wire encryption)	Enforce Minimun Transport Layer Security (TLS\SSL) Standard to TLS1.2 [106]	Enforce Minimun Transport Layer Security (TLS\SSL) Standard to TLS1.2 to SQL Server [106] & Enforce Minimun

								Transport Layer Security (TLS\SSL) Standard to TLS1.0 to Clients [105]
		Comunicaçõ es seguras						
A.10.8.4 Mensagens eletrônicas A.10.9.1 Comércio eletrônico A.10.9.2 Transações on-line	Protect Cardholder Data Requirement 4: Encrypt transmission of cardholder data across open, public networks	Transmissio n Security Standard – § 164 .312(e)(1)	Padrão de transmissão segura	Mechanism to authenticate ePHI Specification – Addressable § 164 .312(c)(2)	Desabilitar todos os protocolos (TLS/SSL) exceto TLS1.2 para conexão	Desabilitar todos os protocolos (TLS/SSL) exceto TLS1.2 para acesso a dados E Requerer criptografia de chave forte para acesso de cliente web	Use only TLS1.2 [106]	Use only TLS1.2 on data acess [106] **E** Enforce minimun SSL requiriments on IIS [105]
A.10.8.4 Mensagens eletrônicas A.10.9.1 Comércio eletrônico A.10.9.2 Transações on-line	Protect Cardholder Data Requirement 4: Encrypt transmission of cardholder data across open, public networks	Integrity Controls Specification – Addressable § 164 .312(e)(2)(i)	Padrão de integridade de controles	Mechanism to authenticate ePHI Specification – Addressable § 164 .312(c)(2)	Desabilitar todos os protocolos (TLS/SSL) exceto TLS1.2 para conexão	Desabilitar todos os protocolos (TLS/SSL) exceto TLS1.2 para acesso a dados E Requerer criptografia de chave forte para acesso de cliente web	Use only TLS1.2 [106]	Use only TLS1.2 on data acess [106] **E** Enforce minimun SSL requiriments on IIS [105]
A.12.3.1 Política para o uso de controles criptográfico s	Protect Cardholder Data Requirement 4: Encrypt transmission of cardholder data across open, public	Encryption Specification – Addressable § 164 .312(e)(2)(ii	Padrão de criptografia	Mechanism to authenticate ePHI Specification – Addressable § 164 .312(c)(2)	Desabilitar todos os protocolos (TLS/SSL) exceto TLS1.2 para conexão	Desabilitar todos os protocolos (TLS/SSL) exceto TLS1.2 para acesso a dados E Requerer criptografia	Use only TLS1.2 [106]	Use only TLS1.2 on data acess [106] **E** Enforce minimun SSL requiriments on IIS [105]

	networks					de chave forte para acesso de cliente web		
		Controles adicionais						
		§164.316(b)(1)(ii)	Manter um registro físico ou digital de quem, como e de onde audita o ePHI		Implementar grupo auditado de banco de dados para usuários com acesso a tabelas de auditoria	n/a	SQL Server Audit [113]	n/a
A.10.10.3 Proteção das informações dos registros (logs) A.15.3.2 Proteção de ferramentas de auditoria de sistemas de informação		§164.316(b)(2)(i), §164.316(b)(2)(ii), §164.528(a)	Garantir o arquivamento seguro dos registros de segurança por 6 anos apartir da sua data de criação e permitir que esses registros sejam disponíveis de maneira imediata		Provisionar hardware fisico OU Serviço de nuvem elástico para 6 anos de crescimento das tabelas de auditoria	n/a	Provisionar recursos para crescimento dos arquivos de dados específicos de auditoria no servidor local ou serviço do Azure por 6 anos **OPCIONAL** Habilitar recurso auto-growth para estes arquivos se não houverem métricas seguras para 6 anos	n/a
		Não enquadrados, mas potenciais controles baseados em						

		§164 .312(a)(1), §164 .312(a)(2)(i)			Desabilitar recurso "phone home" e telemetria do mecanismo de dados	Desabilitar telemetria do servidor	Configuração de registro [115]	Configuração de aplicativo
		§164 .312(a)(1), §164 .312(a)(2)(i)			Implementar superfície mínima de ataque na instalação		Procedimento de instalação da instância [114]	Procedimento de publicação
		§164 .312(a)(1), §164 .312(a)(2)(i)			Implementar permisões mínimas de usuário para as contas de serviço usadas na instalação		Procedimento de instalação da instância [114]	Procedimento de publicação

Detalhamento de implementação de alguns dos controles da tabela [301]

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 "fall-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 descriptografia de dados [108,109,110]

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

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 PERMISSIONS	OPEN MASTER KEY	OPEN SYMMETRIC KEY	REVOKE DATABASE 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

2.1 Instalação e planejamento

2.1.1 Instalação

Boas práticas na instalação da ferramenta utilizada para criação de banco de dados.

Pode-se utilizar como exemplo o Microsoft SQL Server:

Documentação: Sempre opte em instalar toda a documentação do Microsoft SQL Server nas instâncias que estão sendo configuradas para que seja facilmente acessada em casos de emergência.

Padronização: Sempre que possível, instale e configure todas as suas instâncias do Microsoft SQL Server de forma consistente, seguindo um padrão dentro da organização.

Features Desnecessárias: Não instale serviços desnecessários do Microsoft SQL Server. Sempre avalie a necessidade quando da instalação de serviços como Microsoft Full-Text Search, Notification Services ou o Analysis Services.

Uma prática utilizada nessa seção é a Surface Area Reduction (Redução da área de superfície). de segurança que envolve a interrupção ou a desativação de componentes não utilizados.Alguns recursos, serviços e conexões são desabilitados ou interrompidos para reduzir esta área. Depois de instalar o SQL Server ou atualizar para o SQL Server, você deve executar a configuração da área de Superfície do SQL Server para verificar quais recursos e serviços estão habilitados e em execução e para verificar quais tipos de conexões o SQL Server aceitará. Após a configuração inicial, você pode usar a Configuração da Área de Superfície do SQL Server para verificar ou alterar o estado de recursos, serviços e conexões.

Desempenho: Para um bom desempenho do Microsoft SQL Server rodando sobre sistema operacional Windows, desative todos os serviços do sistema operacional que não forem necessários. Considere sempre instalar seu servidores nas versões do Windows Server.

Otimização: Para um bom desempenho do Microsoft SQL Server, dedique o servidor físico para rodar uma única instância do banco de dados, sem outras aplicações que consumam recursos que seriam destinados ao gerenciador de banco de dados. Por padrão, aplicações rodam somente em servidores de aplicações.

I/O: Para um bom desempenho de escrita e leitura dos discos rígidos, separe os arquivos de banco de dados (.MDF) dos arquivos de log (.LDF)

em unidades de discos diferentes. Considere utilizar sempre um conjunto de discos em raid, uma boa padronização é utilizar um array de discos em raid 5 para o arquivo de dados (.MDF) e um array de discos em raid 10 para o arquivo de log (.LDF). Essa separação é importante, pois gera um

potencial isolamento entre leitura e escrita de discos.

TEMPDB: Se o uso do TEMPDB for alto, considere alocar o arquivos de dados e de log deste banco em um array de discos separado.

Domain Controller: Nunca (digo com bastante ênfase nesta parte) instale o Microsoft SQL Server em um controlador de domínio.

Sempre considere isolar a segurança do seu domínio do(s) seu(s) repositório(s) de dados.

Partição: Certifique-se de que o Microsoft SQL Server está/será instalado em uma partição NTFS.

Encriptação/Compressão: Não utilize na partição onde estão os arquivos de dados do Microsoft SQL Server (.MDF e .LDF) a encriptação (EFS) ou a compressão de dados. Isso afeta e muito o desempenho global de seu servidor de dados.

Formatação da Partição: Considere formatar a partição de dados para o Microsoft SQL Server em modo de alocação 64 KB.

2.1.2 Planejamento

O planejamento de segurança envolve o desenvolvimento de políticas de segurança e a implementação de controles para evitar que os riscos de computador se tornem realidade.
Cada organização é diferente e precisará planejar e criar políticas com base em suas metas e necessidades individuais de segurança.
Depois de identificar os ativos, é necessário determinar todos os riscos que podem afetar cada ativo. Uma maneira de fazer isso é identificar todas as maneiras diferentes pelas quais um ativo pode ser danificado, alterado, roubado ou destruído.

Por exemplo:
Informações financeiras armazenadas em um sistema de banco de dados.

A fim de desenvolver uma política de segurança da informação eficaz, as informações produzidas ou processadas durante a análise de risco devem ser categorizadas de acordo com sua sensibilidade à perda ou divulgação. A maioria das organizações usa algumas categorias de informações, como Proprietary, For Internal Use Only ou Organization Sensitive. As categorias usadas na política de segurança devem ser consistentes com quaisquer categorias existentes. Os dados devem ser divididos em quatro classificações de sensibilidade com requisitos de tratamento separados: confidencial, confidencial, privado e público. Esse sistema de classificação de sensibilidade de dados padrão deve ser usado em toda a organização.

Depois de identificar os riscos e a sensibilidade dos dados, estime a probabilidade de ocorrência de cada risco.

Tipos de Políticas de Segurança utilizadas.

Políticas podem ser definidas para qualquer área de segurança. Cabe ao administrador de segurança e ao gerente de TI classificar quais políticas precisam ser definidas e quem irá defini-las. Os vários tipos de políticas que podem ser incluídos são:

Políticas de senha

Responsabilidades Administrativas

Responsabilidades do usuário

Políticas de email

Políticas da Internet

Políticas de backup e restauração.

Assim, para alcançar um bom planejamento de segurança, é necessário seguir alguns passos indispensáveis:

1. Manter uma política de backup regular;
2. Ter um servidor dedicado ao banco de dados;
3. Dispor de uma opção de redundância;
4. Restringir o acesso ao servidor somente a pessoas autorizadas;
5. Definir restrições de acesso na política de segurança do banco de dados;
6. Atualizar seu ambiente com os últimos patches de segurança.

2.2 Autenticação e autorização

Autenticação é processo relacionado ao momento de login na INSTÂNCIA de banco de dados que pode ser feito através da checagem de credenciais oferecidas através do sistema operacional (Windows Authentication) ou inserção de usuário e senha através do (SQL Authentication) no caso do SQL Server. Outras implementações de banco de dados seguem uma linha parecida.

A escolha entre os dois tipos de autenticação dependem do cenário em que o administrador do banco de dados se encontra. O que deve ser levado em consideração é para tomar essa decisão são os seguintes pontos: Se existe um Domain Controller, se a aplicação e o banco estão na mesma máquina, se há um workgroup, se existem conexões de non-trusted domains e etc...

Autorização é processo de permitir um usuário autenticado usufruir do serviço de banco de dados a depender do seu tipo de acesso. Nesse sentido é possível garantir e/ou revogar direitos do usuário em questão através de comandos DCL (Data Control Language) emitidos pelo administrador do banco de dados.

Os comandos DCL podem ser categorizados basicamente entre GRANT e REVOKE. A depender da versão do Database Management Software é possível que a implementação varie, por exemplo no SQL Server além destes é possível também utilizar o comando DENY. Esses tipos de comandos são portanto responsáveis por gerenciar atribuição de privilégios DML (Data Manipulation Language) e DDL (Data Definition Language) para um usuário ou role (grupo de usuários). Ao atribuir uma role a um usuário você automaticamente já passa todos os privilégios necessários em ‘one go”. Além disto, também existe a possibilidade de passar um privilégio para uma role que se estenderá para todos os usuários dentro dela.

Outros detalhes relacionados a autorização no banco de dados também fazem referência ao controle de locks. De que maneira? Cada banco de dados tem uma implementação padrão de bloqueio de linhas e/ou tabelas para garantir que o princípio ACID seja garantido. A depender da aplicação que roda sobre o banco é necessário que determinadas atividades de “read consistency” e execução de DML requeiram locks mais abrangentes do que as configurações padrão e daí cabe ao administrador do banco de dados avaliar se essas mudanças podem de alguma forma impactar a disponibilidade dos dados, principalmente quando falamos de banco de dados com uma característica muito forte de OLTP (Online Transaction Processing) que muitas vezes pode resultar em degradação de performance devido o fato do desenvolvedor estar solicitando um

lock além do que realmente é necessário impactando todos que utilizam o serviço do database.

Os bancos e dados também são capazes gerar arquivos em diretórios do sistema operacional. Logo é de suma importância que a política de segurança esteja devidamente aplicada para garantir que somente usuário autorizados possam devidamente ler, escrever ou ambos privilégios num arquivo como um XML, CSV ou dumps específicos associados à tecnologia de banco de dados por exemplo.

Os administradores também devem se certificar se os profiles estão devidamente configurados para os usuários que se conectam ao banco de dados. Os profiles são responsáveis determinar parametrização de recurso um usuário pode usar do banco de dados (e. g. quantidade de memória, quantidade de disco, uso de CPU, tempo de inatividade de sessão e etc.) e parâmetro de senha (e. g. quantidade de tentativas de login, número de caracteres, quais caracteres, data de expiração e etc.)

2.3 Sigilo e canal seguro

Além da autenticação e autorização que são conceitos chaves da segurança da informação, compondo o A.A.A. que, é base para qualquer aplicação/serviço seguro,também temos o sigilo que se procurarmos o significado em um dicionário, teremos que é "o que permanece escondido da vista ou do conhecimento", e essa definição é exatamente o que tem de haver em um banco de dados.

Sabendo disso, a melhor maneira de provarmos esse conceito aos bancos de dados é aplicando criptografia nos arquivos e logs armazenados, a fim de fazer com que os arquivos ali presentes permaneçam intactos e os registros de atividades (logs) inalterados. Como podemos ver no próprio artigo da Microsoft[1] a respeito do banco de dados SQL, temos a opção de aplicar diferentes algoritmos de criptografia, sendo estes: DES, Triple DES, TRIPLE_DES_3KEY, RC2, RC4, RC4 de 128 bits,DESX, AES de 128 bits, AES de 192 bits e AES de 256 bits.

Outro ponto importante quanto ao sigilo dos dados presentes em um banco de dados, é quanto à mídia física da aplicação. O TDE ou criptografia de dado transparente, trabalho com os dados em repouso, ou seja, aqueles que não estão sujeitos ao tráfego. Sendo assim, esse método, consiste em encriptar o conteúdo presente na mídia física, a fim de impossibilitar que um atacante consiga acessar os arquivos ali presentes, através do roubo do mesmo.

O canal seguro por sua vez, se consiste em prover uma conexão confiável entre o banco de dados e o usuário. Para que isso seja possível utilizamos o TLS (transport layer security) que se consiste em um protocolo de segurança que provém comunicações seguras, como por exemplo o e-mail, navegação web (https) e outros meios de transferência de dados (como é o caso dos bancos de dados). O mesmo trata-se de uma criptografia assimétrica que garante integridade e privacidade do canal que será criptografado, impedindo a ação bem sucedida de sniffers (interceptadores de tráfego), uma vez que estes não conhecem a chave.

---

[1] https://docs.microsoft.com/pt-br/sql/relational-databases/security/encryption/choose-an-encryption-algorithm, acessado em 21/04/2018, às 16:50.

Outro conceito bastante importante quando tratamos de canal seguro, é a proteção estendida. Ela tem como função impedir ataques de retransmissão de autenticação, seja por atração ou falsificação. Podemos encontrar nesse tipo de proteção a associação de serviço que, exige o envio de um SPN (nome da entidade) pelo cliente, mitigando ataques de engenharia social (atração), e também a associação de canal que, estabelece um canal seguro entre o banco de dados e o cliente, verificando um token, e impedindo ataques tanto de atração, quanto de falsificação.

2.4 Ataques a banco de dados

Quando tratamos de banco de dados que não possuem controles de segurança adequados como os descritos na seção anterior, os atacantes conseguem facilmente explorar essas vulnerabilidades através de ataques como o XSS e o SQL Injection.

2.4.1 XSS (Cross-Site Scripting)

O Cross-site scripting é baseado no conceito de injeção e execução de scripts/códigos maliciosos em uma página web, ocorrendo, geralmente, na forma de um script ao lado do browser do usuário. Basicamente, o XSS busca se aproveitar do princípio de confiança que o alvo tem em relação ao site usado como isca para um outro não confiável, fazendo com que a execução ocorra sem suspeitas e consequentemente a máquina seja infectada.

Os motivos que levam a esse ataque ter sucesso são, principalmente, os seguintes:

1. Fonte desconfiável
2. Ausência de validação de uma entrada

Uma vez que o script estiver inserido na página, será possível a extração de diversas informações sobre o usuário, como por exemplo cookies e dados de sessão, tendo até a possibilidade de alterar elementos da página. Esse tipo de ataque permite que os atacantes consigam obter a sessão da vítima, redirecionando-o para um site malicioso ou até mesmo “pichando” o site.

Por outro lado, temos dois tipos de cross-site scripting, sendo eles:

1. Reflected: ocorre quando o servidor retorne parâmetros antigos, ou seja, já enviados anteriormente, sem que haja uma validação ou filtragem do conteúdo.
2. Stored: ocorre quando o servidor armazena a entrada maliciosa em seu banco de dados, fazendo com que a saída não seja validada, retornando de maneira explícita.

Para prevenir-se de ataques de XSS, é importante com que a validação e filtragem de entradas como inputs, caixas de texto e parâmetros de URL, sejam sempre realizadas e nunca enviadas diretamente. Outro ponto importante para prevenção, é o uso de headers específicos, fazendo com que o browser possa identificar e não validar a injeção de script/código.

2.4.2 SQL Injection

A injeção de SQL é baseada na manipulação do código da linguagem de pesquisa declarativa de banco de dados relacionais. Esse ataque é feito por meio da exploração de uma vulnerabilidade/falha na codificação de uma aplicação, que possua um input para que a manipulação possa ser executada por meio deste.

É uma técnica onde o atacante introduz comandos maliciosos no banco de dados, por meio dos campos de escritas ou até mesmo de uma URL, a fim de extrair informações importantes do DB, como por exemplo usuários e senhas.

Conforme informado pela equipe da TecMundo[2], em janeiro de 2017, a página de busca do Google, sofre um defacement, ação esta que foi confirmada pela própria equipe da Google. Esse ataque, como informado pelo próprio "hacker", utilizou técnicas de injeção de SQL.

3. Conclusão

Em 2017, ataques digitais tomaram conta da mídia mundial. O Brasil sofreu 30 ataques de negação de serviço (DDoS) por hora, a HBO sofreu com o roubo de seus episódios de Game of Thrones por hackers, milhões em criptomoedas foram roubadas em todo o planeta, 57 milhões de usuários da Uber[3] tiveram seus dados roubados e o famoso WannaCry assombrou todo o mundo. Já em 2018, não começou diferente, pois os Jogos de Inverno tiveram sua abertura hackeada, a Netshoes teve os dados de seus clientes vazados, a Intel veio a público com duas vulnerabilidades graves em seus chips, entre outros.

Quando falamos de banco de dados, os riscos se tornam ainda mais elevados, visto que a importância que eles possuem em cada aplicação e serviço hoje existentes, é indiscutível. Sendo assim, com a intenção de blindar o máximo possível os DB de atacantes, diversas ferramentas e técnicas de segurança, são aplicadas nos mais diversos tipos de banco de dados, com o intuito de diminuir a probabilidade de um ataque ser bem-sucedido e seu consequente impacto.

Levando em consideração, a frequência e complexidades que esses ataques ocorrem, é inevitável que empresas, instituições e governos, contratem cada vez mais profissionais do ramo da segurança da informação. Os ativos de todas as grandes empresas da atualidade estão no meio digital, portanto, todo cuidado é pouco, visto que se caso forem roubados ou simplesmente atacados, severas perdas ocorrerão, sejam diretamente financeiras ou até mesmo de reputação.

---

[2] https://www.tecmundo.com.br/tecmundo-explica/113195-sql-injection-saiba-tudo-ataque-simples-devastador.htm, acessado em 7 de abril de 2018, às 16:14.

[3] https://exame.abril.com.br/tecnologia/57-milhoes-de-usuarios-do-uber-tiveram-dados-roubados-por-hackers/ acessado em 06/04/18, às 09:30.

Referências

- Referência geral
    - Shivnandan Singh; Rakesh Kumar Rai. A Review Report on Security Threats on Database/ (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3215 - 3219
- Normas
    - PCI DSS
        - Norma
            - Official PCI Security Standards Council Site - Verify PCI Compliance, Download Data Security and Credit Card Security Standards
            - Indústria de cartões de pagamento (PCI) Padrão de Segurança de Dados - Requisitos e procedimentos da avaliação de segurança
        - Artigos
            - PCI DSS 3.2 and SQL Server – By Grant Carter INTRODUCTION – Mark Weber's blog

- ISO IEC 27002
  - Norma
    - ABNT Catalogo (ABNT NBR ISO/IEC 27002)
  - Artigos
    - 13 effective security controls for ISO 27001 compliance | Blog | Microsoft Azure
    - Download SQL Server White Paper: SQL Server 2008 Compliance Guide from Official Microsoft Download Center
    - Now Available: Guide for enhancing privacy and addressing GDPR requirements with the Microsoft SQL platform | SQL Server Security Blog
- HIPPA
  - Norma
    - https://www.hhs.gov/hipaa/index.html
  - Artigos
    - FIN_HIPAA-Complaince-with-SQL_050211.pdf
    - HIPAA Compliance for SQL Server DBAs - Solution center
    - Summary of the HIPAA Security Rule | HHS.gov
- Descrições gerais e apresentação das normas
  - ACID - Wikipedia
  - Microsoft SQL Server - Wikipedia
  - Payment Card Industry Data Security Standard - Wikipedia
  - ISO/IEC 27002 - Wikipedia
  - Health Insurance Portability and Accountability Act - Wikipedia
  - Sarbanes–Oxley Act - Wikipedia

- Referencias técnicas específicas de plataforma SQLServer
  - Gerais
    - Segurança de dados — Criptografia do SQL Server | Microsoft
    - Centro de segurança do Mecanismo de Banco de Dados do SQL Server e do Banco de Dados SQL do Azure | Microsoft Docs
    - Leia o white paper SQL Server Data Security GDPR
  - Instalação e Planejamento
    - https://msdn.microsoft.com/en-us/library/hh567632(v=cs.95).aspx
    - https://technet.microsoft.com/en-us/library/ms173748(v=sql.90).aspx
    - https://www.devmedia.com.br/boas-praticas-de-seguranca-para-sql-server-2008-2008-r2-revista-sql-magazine-90/22006
  - Sigilo e Canal Seguro
    - https://technet.microsoft.com/pt-br/library/mt634205(v=vs.85).aspx
    - https://msdn.microsoft.com/pt-br/library/dn948096(v=sql.120).aspx
  - Consumo e Entrega
    - https://www.tecmundo.com.br/tecmundo-explica/113195-sql-injection-saiba-tudo-ataque-simples-devastador.htm
    - https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)
- Referências técnicas específicas da tabela de cruzamento de normas e aplicações

- [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
- [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
- [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
- [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

- [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
    - Tabela específica do cenário para IIS +7 e Windows +10 (v1057 de lançamento) https://msdn.microsoft.com/library/windows/desktop/mt767769.aspx
    - 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
    - Referência adicional para forçar uso de HTTPS no .NET Core +2.x https://docs.microsoft.com/en-us/aspnet/core/security/enforcing-ssl
- [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
    - 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
    - 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

- [107] Como conectar ao SQL Server usando DAC https://technet.microsoft.com/pt-br/library/ms178068(v=sql.105).aspx
- [108] Always Encripted https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine
    - Detalhes da criptografia e métodos https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-cryptography
- [109] Transparent Data Encription https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption
    - 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
- [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
- [111] Visão geral do padrão MVC https://msdn.microsoft.com/pt-br/library/dd381412(v=vs.108).aspx
- [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
- [113] Visão geral da auditoria do SQL Server https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine
    - Grupos de auditoria do SQL Server https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-action-groups-and-actions
    - 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
- [114] Instalação segura do SQL Server https://docs.microsoft.com/en-us/sql/sql-server/install/security-considerations-for-a-sql-server-installation
- [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