ホワイトペーパー

PostgreSQL セキュリティ ベストプラクティス 2020アップデート

EDB

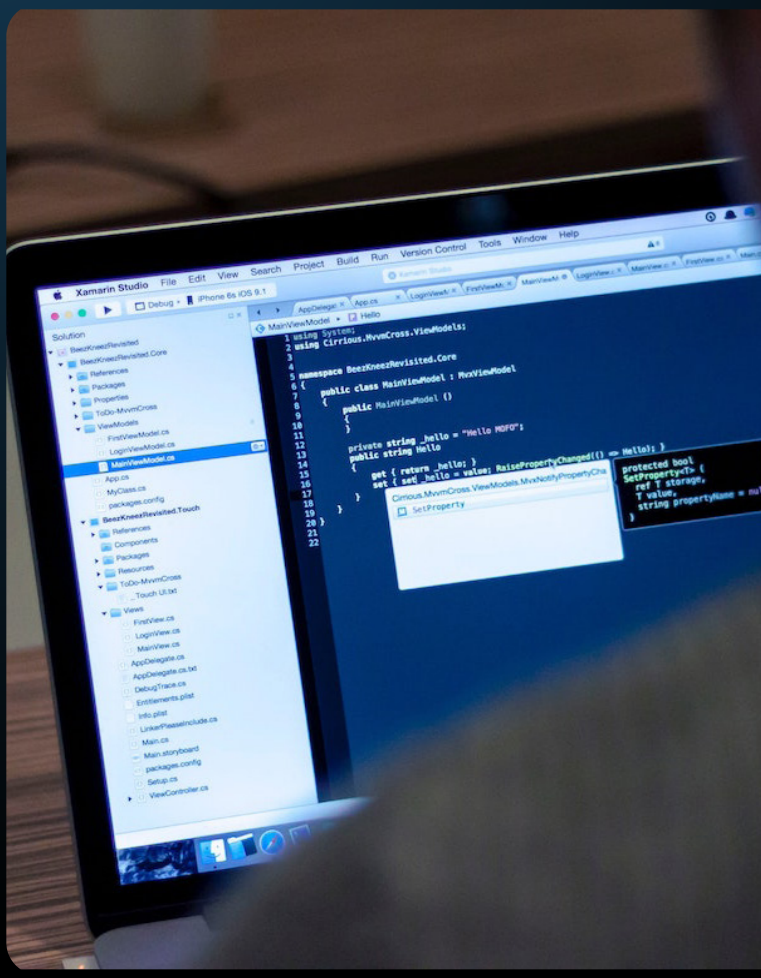
EDBPOSTGRES.COM

目次

1

エグゼクティブサマリー

このホワイトペーパーでは、PostgreSQLデータベースを保護および保護するためのフレームワークと一連の推奨事項について説明します。物理的セキュリティ、ネットワークセキュリティ、ホストアクセス制御、データベースアクセス管理、およびデータセキュリティに対応する階層型セキュリティモデルについて説明します。これらの側面はすべて等しく重要ですが、このペーパーでは、データベースとデータを保護するPostgreSQL固有の側面に焦点を当てています。データベースおよびデータベースで管理されるデータに関連する特定のセキュリティの側面について説明するために、コンピュータとネットワークのセキュリティに共通のAAA（認証、承認、および監査）アプローチを使用します。



このホワイトペーパーの推奨事項のほとんどは、PostgreSQL（コミュニティエディション）と、EnterpriseDB®（EDB™）からのPostgreSQLのエンタープライズクラスの機能豊富な商用ディストリビューションであるEDBPostgres™Advanced Server（Advanced Server）に適用できます。Advanced Serverは、パスワードプロファイル、監査、データ編集、SQL Serverインジェクション保護など、PostgreSQLでは同じ形式では利用できない追加の関連するセキュリティ拡張機能を提供します。

このドキュメントは、PostgreSQL12およびEDBPostgres Advanced Server12用に更新されています。

2

はじめに

レイヤーのセキュリティについて考えることができ、ジョブや役割に必要な最小限のアクセスを許可し、できるだけ早い機会に不要なアクセスをブロックする戦略をアドバイスできます。

1つは、ホストへの物理アクセスを保護することです。

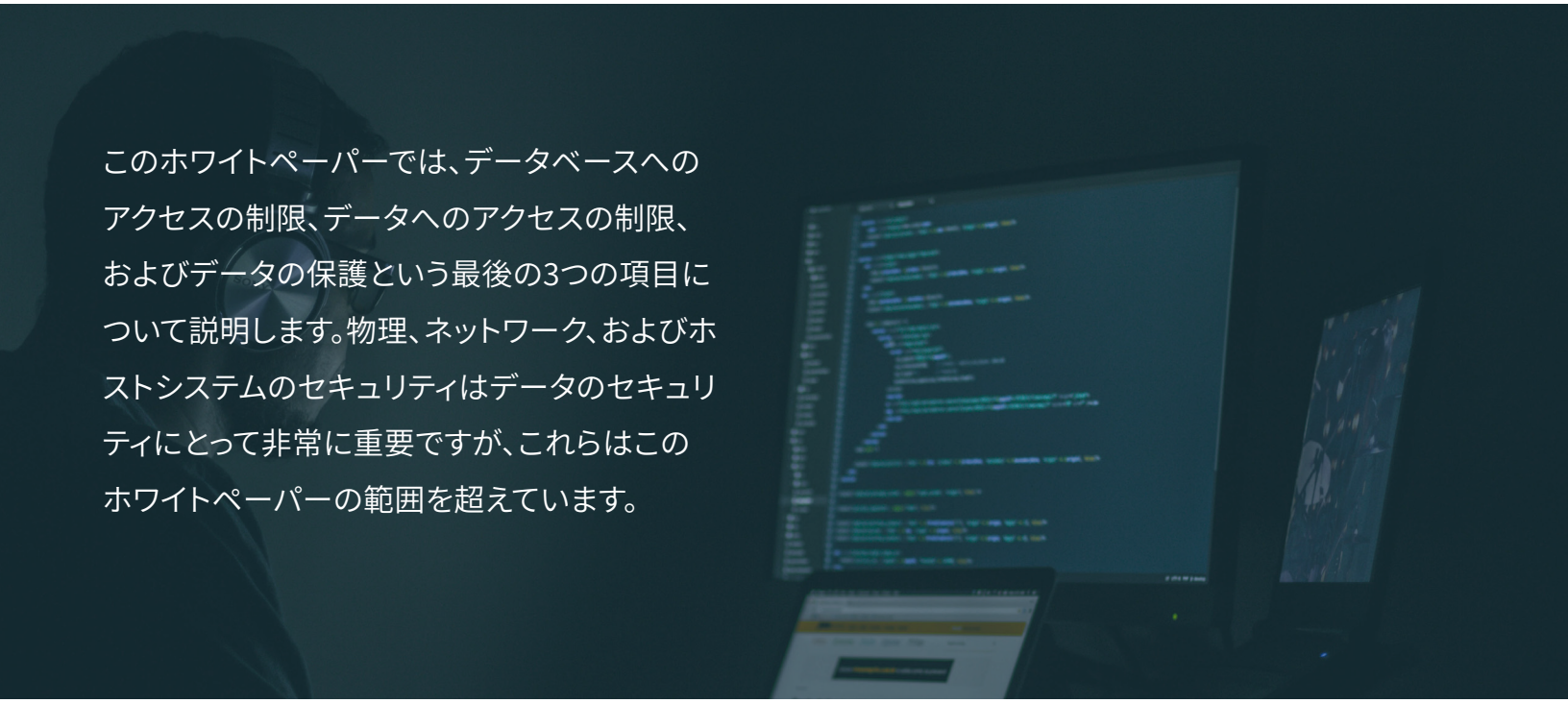
次は、データベースアプリケーションへのアクセスを制限することです。

次は、一般的に企業ネットワークへのアクセスを制限することです。

次は、そこに含まれるデータへのアクセスを制限することです。

次は、データベースホストへのアクセスを制限することです。

次は、内部に保存されているデータを保護することです。



このホワイトペーパーでは、データベースへのアクセスの制限、データへのアクセスの制限、およびデータの保護という最後の3つの項目について説明します。物理、ネットワーク、およびホストシステムのセキュリティはデータのセキュリティにとって非常に重要ですが、これらはこのホワイトペーパーの範囲を超えています。

一般的な推奨事項

- オペレーティングシステムとデータベースにパッチを適用したままにします。EDBのサポートサブスクリプションは、Postgresのセキュリティアップデートと適切なパッチのタイムリーな通知を提供します。yum / dnfやaptなどのパッケージ管理システムと統合できるオペレーティングシステムのアップグレードを監視するために利用できるさまざまなツールがあります。
- ビジネスにとって本当に重要な場合を除いて、ポストマスターポートをインターネットに配置しないでください。このポートを適切にファイアウォールで保護します。それが不可能な場合は、読み取り/書き込みマスターではなく、読み取り専用のスタンバイデータベースをポートで使用できるようにします。すべての接続を監査するネットワークポート転送は、有効な代替手段です。
- サブネット化または他の手法を使用して、データベースポートを他のネットワークトラフィックから分離します。
- ユーザーに、作業を行うために必要な最小限のアクセス権を付与します。それ以上は許可しません。絶対に必要なタスクまたはロールには、スーパーユーザーアカウントの使用を予約してください。
- 構成ファイル（postgresql.confおよびpg_hba.conf）およびログファイル（pg_log）へのアクセスを管理者に制限します。
- データベーススーパーユーザーロール（PostgreSQLではpostgres 、EDB Postgres Advanced Serverではenterprisedb ）によるホストシステムのログインを禁止します。例外的な状況では、必要な場合にのみスーパーユーザーアクセスを有効にします。
- 各ユーザーに独自のログインを提供します。共有資格情報は推奨される方法ではなく、監査がより複雑になります。または、edb_audit_tag機能（EDB Postgres Advanced Serverでのみ使用可能）を使用して、アプリケーションがアプリケーションレベルの接続から生じるセッションに監査情報を追加できるようにします。
- データベースへの不正アクセスを防ぐために、フロントエンドアプリケーションだけに依存しないでください。データベースセキュリティを、LDAP / ADやKerberosなどのエンタープライズレベルの認証および承認モデルと統合します。
- バックアップを保持し、テスト済みの復旧計画を立てます。システムをどれだけ安全に保護しても、侵入者が侵入してデータを削除または変更する可能性があります。不正アクセスを防ぐために、バックアップも安全に保管してください。

ネットワークとコンピュータのセキュリティ用に開発されたAAAモデルの観点からセキュリティを考えると役立つ場合があります。AAAは、Authentication、Authorization、andAuditingの略です。

- 認証：ユーザーが本人であることを確認します。
- 承認：ユーザーにアクセスが許可されていることを確認します。
- 監査（またはアカウンティング）：ユーザー名と時刻を含むすべてのデータベースアクティビティをログファイルに記録します。

すべての機能がこれらのカテゴリに適切に適合するわけではありませんが、AAAモデルはこの議論に役立つフレームワークを提供します。

3

PostgreSQLセキュリティ機能をAAAフレームワークに適用する

3.1 認証

pg_hba.conf（PostgreSQLホストベースアクセス）ファイルは、ユーザー名、データベース、およびソースIP（ユーザーがTCP / IP経由で接続している場合）に基づいてアクセスを制限します。このファイルでも認証方法が割り当てられています。選択する認証方法は、ユースケースによって異なります。

Kerberos / GSSAPI — PostgreSQLは、RFC 1964に準拠したKerberos認証でGSSAPIをサポートします。GSSAPIは、GSSAPIをサポートするシステムに自動認証（シングルサインオン）を提供します。認証自体は安全ですが、GSSまたはSSL暗号化が使用されていない限り、データベース接続を介して送信されるデータは暗号化されません。

SSPI — Windowsシステムを使用していて、シングルサインオン（SSO）認証を実装する場合に使用します。

LDAPは、Kerberos（SSPIとGSSAPIの両方を含む）が問題外の場合にのみ使用する必要があります。パスワードはLDAPサーバーに転送されるため、LDAPの安全性は低く、安全でない方法で簡単に設定できます。

RADIUSは暗号化が弱く、資格情報にmd5ハッシュを使用しているため、使用しないでください。

証明書— TLS証明書認証（SSLと呼ばれることもあります）は、ネットワーク上のトラフィックの暗号化と認証に使用できます。証明書は、マシン間通信でよく使用されます。

LDAPとRADIUS— LDAPとRADIUSは、多数のユーザーがいて、中央の場所からパスワードを管理する必要がある状況で役立ちます。この一元化には、pg_hba.confファイルを小さく、管理しやすくするという利点があり、インフラストラクチャ全体でユーザーに「統一されたパスワードエクスペリエンス」を提供します。データベースにアクセスするには、サービスとそのサービスへの接続に依存しているため、LDAPとRADIUSの両方に強固なインフラストラクチャが必要です。

md5 — md5は、ユーザー名とパスワードの情報をデータベースに保存します。これは、ユーザー数が非常に少ない場合に適した代替手段となる可能性があります。パスワードは安全にハッシュされるため、Scramはmd5よりも非常に優先されます。

スクラム—信頼できるユーザーの数が非常に少ない場合は、scram-sha-256認証を使用することをお勧めします。パスワードは安全にハッシュされるため、Scramはmd5よりも非常に優先されます。

拒否—この方法を使用して、特定のユーザー、特定のデータベースへの接続、および/または特定のソースIPを拒否します。

信頼—信頼認証は、一致するクライアントがそれ以上の認証なしでサーバーに接続できるようにするため、例外的な状況でのみ使用する必要があります。

各認証方法の影響を完全に理解していることが不可欠です。これらおよびその他の認証方法の詳細については、PostgreSQLのドキュメントを参照してください。

はじめにで述べたように、pg_hba.confファイルへのアクセスは管理者に制限する必要があります。このファイルを適切に整理しておくようにしてください。より大きく、より複雑なファイルは、保守が難しく、誤ったエントリや古いエントリが含まれている可能性が高くなります。このファイルを定期的に確認して、不要なエントリがないか確認してください。

3.2 パスワードプロファイル

バージョン9.5以降、Advanced Serverは、MD5またはSCRAM認証を使用する場合にOracle互換のパスワードプロファイルをサポートします。パスワードプロファイルは、DBAが同等の認証要件を共有するロールのグループを簡単に管理できるようにするパスワード属性の名前付きセットです。各プロファイルは、1人以上のユーザーに関連付けることができます。ユーザーがサーバーに接続すると、サーバーはログインロールに関連付けられているプロファイルを適用します。

2.3が利用可能に、より多くの情報については、ORACLE®開発者ガイドのためにEDBのデータベースの互換性の「プロファイル管理」を参照してください。

プロファイルは次の目的で使用できます。

- 失敗したログイン試行の許容回数を指定します。
- ログイン試行の失敗が多すぎるため、アカウントをロックします。
- 有効期限のパスワードをマークします。
- パスワードの有効期限が切れた後の猶予期間を定義します。
- パスワードの複雑さに関するルールを定義します。
- パスワードの再利用を制限するルールを定義します。

3.3 承認

ユーザーが適切に認証されたら、データを表示してデータベースで作業を実行するためのアクセス許可を付与する必要があります。以前にアドバイスしたように、ユーザーがジョブを実行するために必要な特権のみを付与し、共有（グループ）ログイン資格情報を禁止します。役割の割り当てを介してPostgreSQLのユーザーとグループを管理します。ロールは、個々のユーザーまたはユーザーのグループを指す場合があります。Postgresでは、ロールはクラスター（データベースサーバー）レベルで作成されます。これは、クラスター/データベースサーバー用に定義されたすべてのデータベースにロールが適用されることを意味します。ロールの権限を適切に制限することは非常に重要です。権限は、データベースオブジェクト（テーブル、ビュー、関数など）、テーブル内の行、および編集ポリシーに適用できます。

3.3.1 – データベースオブジェクトへのアクセス

割り当てられた特権と警告は、PostgreSQL CREATEROLEのドキュメントに概説されています。

- すべてのユーザーからCREATE特権を取り消し、信頼できるユーザーにのみ付与します。
- 信頼できない手続き型言語で記述された関数またはトリガーの使用を許可しないでください。
- SECURITYDEFINER関数を使用すると、ユーザーは制御された方法で昇格された特権レベルで関数を実行できますが、不注意に記述された関数は不注意にセキュリティを低下させる可能性があります。詳細については、ドキュメント「CREATEFUNCTIONのセキュリティ定義関数を安全に作成する」を確認してください。
- データベースオブジェクトは、アプリケーションユーザーが接続できるロールではなく、安全なロール、理想的にはデータベースへのアクセスが非常に制限されているロール（たとえば、Unixドメインソケットからのみ）が所有する必要があります。これにより、攻撃者がオブジェクトを変更または削除できる可能性が最小限に抑えられます。これはセキュリティの観点からは好ましいことですが、スキーマ自体を管理するアプリケーションフレームワークでは問題が発生する可能性があります。このような機能は注意して実装する必要があります。

log_statementが「ddl」以上に設定されている場合、ALTER ROLEコマンドを使用してロールのパスワードを変更すると、edb_filter_log.redact_password_commandがサーバーに指示するEDB Postgres Advanced Server 11以降を除き、ログにパスワードが公開されることに注意してください。保存されたパスワードをログファイルから編集します。

認証情報（ユーザー名やパスワードなど）がテーブルに格納されている場合、テーブルが名目上安全であっても、ステートメントロギングを使用するとその情報が公開される可能性があります。同様に、機密情報がクエリで使用される場合（たとえば、キーとしてのあらゆる種類の個人情報）。これらのパラメーターは、ステートメントのロギングによって公開できます。

3.3.2 – ビュー

ビューへのアクセスは上記のように制御でき（データベースオブジェクトです）、ビューを使用して、テーブルのVIEWを作成し、そのVIEWのアクセス許可を制限することで、特定のユーザーグループに対するデータの表示を制限できます。PostgreSQLバージョン9.2以降では、Robert Haasによって説明されているようなセキュリティの問題を回避するために特別な予防措置が必要と思われる場合に、CREATE VIEW WITH（security_barrier）のオプションが提供されます。

3.3.3 – 行レベルのセキュリティ

PostgreSQLはバージョン9.5で行レベルセキュリティ（RLS）を導入しました。RLSを使用すると、現在のユーザーロールに基づいてテーブル行にきめ細かくアクセスできます。これには、SELECT、UPDATE、DELETE、およびINSERT操作が含まれます。詳細については、こちらをご覧ください。

EDB Postgres Advanced ServerのDBMS_RLSパッケージには、このメカニズムのOracle互換の実装が含まれています。これには、ADD_POLICY、DROP_POLICY、およびUPDATE_POLICYのOracle互換の実装が含まれています。

3.3.4 – データ編集

データの編集-一部のデータ要素を非表示にしたり、特定のユーザーグループのデータを選択的に難読化したりする機能は、データへのアクセスを管理するためのもう1つの手法です。EDB Postgres Advanced Serverは、バージョン11でデータ編集を導入しました。

データ編集は、PostgreSQLの役割と連携して、特定のデータ要素への読み取りアクセスを許可または取り消すポリシーベースのツールです。たとえば、あるユーザーグループには社会保障番号がXXX-XX-1235と表示されますが、データ管理者の役割のメンバーには詳細が表示されます。データ編集に関する追加情報は次のとおりです。

Constant	Type	Value	Description
NONE	INTEGER	0	No redaction, zero effect on the result of a query against table.
FULL	INTEGER	1	Full redaction, redacts full values of the column data.
PARTIAL	INTEGER	2	Partial redaction, redacts a portion of the column data.
RANDOM	INTEGER	4	Random redaction, each query results in a different random value depending on the datatype of the column.
REGEXP	INTEGER	5	Regular Expression based redaction, searches for the pattern of data to redact.
CUSTOM	INTEGER	99	Custom redaction type.

3.13.1 Using DBMS_REDACT Constants and Function Parameters

3.4 監査

Advanced Serverは、監査レポートを作成する機能を提供します。データベース監査により、データベース管理者、監査人、およびオペレーターは、複雑な監査要件をサポートするためにデータベースアクティビティを追跡および分析できます。これらの監査対象のアクティビティには、データベースへのアクセスと使用、およびデータの作成、変更、または削除が含まれます。監査システムは、構成ファイルで定義された構成パラメーターに基づいています。

監査することをお勧めします（精査のレベルを上げることによってリストされています）：

- ユーザー接続
- DDLの変更
- データの変更
- データビュー

非常に詳細なレベルの精査により、多くのログメッセージが発生する可能性があります。必要なレベルでのみログに記録します。Postgresを使用すると、ユーザーごとおよびデータベースごとにログレベルを調整できます。異常な動作がないか、監査ログを頻繁に確認してください。ログの管理過程を確立します。

高いログレベルとデータベースへのパスワードの保存を組み合わせると、パスワードがログに表示される可能性があることに注意してください。EDB Postgres Advanced Serverは、バージョン11でedb_filter_log.redact_password_commands拡張機能を導入して、監査ログファイルから保存されたパスワードを編集するようサーバーに指示します。

ここでは、AdvancedServerの監査ログ機能の詳細を参照してください。

> Advanced Serverを使用すると、データベースとセキュリティの管理者、監査人、およびオペレーターは、EDB監査ログ機能を使用してデータベースアクティビティを追跡および分析できます。

3.5 データ暗号化

PostgreSQLはいくつかのレベルで暗号化を提供し、データベースサーバーの盗難、悪意のある管理者、および安全でないネットワークによる開示からデータを保護する柔軟性を提供します。

- パスワードストレージの暗号化
- 特定の列の暗号化
- データパーティションの暗号化
- ネットワーク全体でのパスワードの暗号化
- ネットワーク全体でのデータの暗号化
- SSLホスト認証
- クライアント側の暗号化

これらのオプションの詳細については、PostgreSQLのドキュメントをご覧ください。

クライアントとデータベース間の転送中にデータがスニッフィングされることが懸念される場合は、データスニッフィングがリスクではないことが確実でない限り、postgresql.confファイルでSSLを有効にしてください。SSL暗号化はオーバーヘッドを追加する可能性があり、証明書の管理には注意が必要ですが、一般的にはこれが推奨される方法です。

データベース内、またはファイルシステムレベル（いずれか）でデータを暗号化することもできます。EDBのブログで透過的データ暗号化の詳細を参照してください。この暗号化オプションを使用すると、データはファイルシステムから読み取られるときに復号化されるため、DBAはデータを表示できます。役割と特権をロックダウンすることが不可欠です。その他のオプションには、Thales Vormetric Transparent Encryption（VTE）の使用が含まれます。

pgcrypto contribモジュールを使用して、列ごとにデータを暗号化します。この方法にはいくつかの欠点があります。

- テーブルのサイズによっては、パフォーマンスが低下する可能性があります。
- 暗号化されたフィールドは検索またはインデックス付けできません。
- 暗号化はテーブルの作成時に適用する必要があり、高度な計画が必要です。
- 暗号化キーの管理も複雑になる可能性があります。

さらに、アプリケーションは暗号化/復号化を処理して、データベースとの各交換が暗号化されたままになり、悪意のあるDBAがデータを表示しないようにする必要があります。

3.6 SQLインジェクション攻撃

SQLインジェクション攻撃は、データベースのコンテンツ、構造、またはセキュリティに関する手がかりを攻撃者に提供するSQLステートメントを実行することによってデータベースを侵害しようとする試みです。SQLインジェクション攻撃を防ぐことは、通常、アプリケーション開発者の責任です。データベース管理者は通常、潜在的な脅威をほとんどまたはまったく制御できません。

PostgreSQLでSQLインジェクション攻撃を防ぐための標準的な方法は、パラメータ化されたクエリを使用することです。EDB Postgres Advanced Serverを使用している場合は、SQL / Protectモジュールを使用してSQLインジェクション攻撃から保護することをお勧めします。SQL / Protectは、一般的なSQLプロファイルの受信クエリを調べることにより、通常のデータベースセキュリティポリシーに加えてセキュリティの層を提供します。SQL / Protectは、潜在的に危険なクエリについて管理者に警告し、これらのクエリをブロックすることにより、データベース管理者に制御を戻します。詳細については、ここをクリックしてください。

```
shared_preload_libraries = '$libdir/dbms_pipe,$libdir/edb_gen,$libdir/sqlprotect'
                                        # (change requires restart)
        .
        .
        .
edb_sql_protect.enabled = off
edb_sql_protect.level = learn
edb_sql_protect.max_protected_roles = 64
edb_sql_protect.max_protected_relations = 1024
edb_sql_protect.max_queries_to_save = 5000
```

4.1.2 Configuring SQL/Protect

4

参考文献

- WindowsおよびLinux上のPostgreSQLのEDBセキュリティ技術実装ガイドライン（STIG）
- ブログ：PostgreSQLを保護する方法：セキュリティ強化のベストプラクティスとヒント
- ブログ：パスワードプロファイルを使用した役割の管理：パート1
- ブログ：パスワードプロファイルを使用した役割の管理：パート2
- ブログ：パスワードプロファイルを使用した役割の管理：パート3

EDBについて

PostgreSQLは、イノベーションを促進し、ビジネスを加速させようとしている組織にとって、ますます選択されるデータベースになっています。EDBのエンタープライズクラスのソフトウェアはPostgreSQLを拡張し、お客様がオンプレミスとクラウドの両方でPostgreSQLを最大限に活用できるように支援します。また、24時間年中無休のグローバルサポート、プロフェッショナルサービス、およびトレーニングは、お客様がリスクを管理し、コストを管理し、効率的に拡張するのに役立ちます。

EDBは、世界中に16のオフィスを持ち、主要な金融サービス、政府、メディアと通信、情報技術組織など、4,000を超える顧客にサービスを提供しています。人、チーム、企業向けのPostgreSQLについては、EDBpostgres.comにアクセスしてください。

ホワイトペーパー

PostgreSQL セキュリティ ベストプラクティス 2020アップデート

EDB™

EDBPOSTGRES.COM