4D セキュリティ・ガイド

このガイドについて

セキュリティとはリスクの排除、許可のないアクセスや不法な情報の漏洩をブロックするだけでなく、データ喪失の防御と破損に対する防衛も含まれます。このガイドでは、あなたのデータベースを守る 4D の独自のセキュリティ機能について説明します。また、4D のセキュリティを使って、あなた自身の特別なセキュリティ・ポリシーを強化する方法についても記述しています。

概要

あなたのデータを破壊、喪失、失敗などから保護する 4D の主要な機能には以下のものが含まれています:

- **認証**: 4D はビルトインおよびカスタマイズされた認証をサポートしています。アクティブ・ディレクトリと LDAP による認証も可能です。
- **低レベル認証システムによるアクセス・コントロール**: 4D にはビルトインのユーザー認証システムが含まれており、データベース中の情報あるいはデータベース操作へのアクセス権限を個別に設定したグループの作成が可能です。
- **データ暗号化**: 慎重に扱うべき情報を含むテーブルを、コードやユーザー・インターフェイスを用いて暗号化することによって、データの機密性を確かなものにします。
- **バックアップとログ: あなたのデータやファイル・ストラクチャを検証、メンテナンス、バックアップするツールが、実行に失敗した場合やデータ破損、事故による消去の場合に、データの整合性を確保します。**
- **サーバーのモニタリングと管理**: プロセスやユーザーのリストを簡単に検索し、特定のプロセスを起動したユーザーの特定や、アイドル状態のユーザーの接続を切断することができます。

認証

認証とは、ユーザーの認証情報の照合プロセスで、通常はユーザー名とパスワードをベースにしています。データベースにアクセスするとき、セッションと認証されたユーザーを関連付けるために、ログイン/パスワードの組み合わせを求めることが推奨されます。

4D は3つの異なる方法でアカウントを照合します:

- ***ビルトイン認証システム:ログインとパスワードのチェック。***
- ***外部認証:*4DServer アプリケーションは認証をあなたの** アクティブ・ディレクトリーに委任でき、Windows セッション・ログインを取得し、それを使用して 4D ユーザーを標準のログインメソッドでアプリケーションにログインさせることができます。
- ***カスタマイズ・システム:*4D は、カスタム認証システムを構築するツールとコマンドの組み合わせを提供します。**

ビルトイン認証

初期設定のアカウント

全てのデータベースには、デフォルトでふたつのユーザー・アカウントがあります:

デザイナーは、もっとも強力なアカウントです。データベースのデザインをコントロールし、ユーザーとグループを作成し、アクセス権限を割り振り、デザイン環境を使うこともできます。データーベースはデザイナーに対して制限できる点はありません。あなたのデータベースをいかなる不正なアクセスからも保護するために、確実に強力なパスワードを割り当ててください。

アドミニストレーター(管理者):このアカウントは、通常アクセス・システムを管理する仕事を与えられます。デザイナーに次いで最も強力なユーザーとして考えられます。
これらのふたつのアカウントは消去できませんが、名前の変更はできます。変更した場合でも、デザイナーは赤、管理者は緑というように、アイコンの色で識別することができます。

デフォルト・ユーザーを有効にする

あなたのデータベースにデフォルトのユーザーを設定するためには、セキュリティ・オプションを有効化しなければなりません。一度有効にすれば、『データベース設定』の「セキュリティ」ページのドロップダウン・リストから”デフォルト・ユーザー”を選択できます。このオプションは、データベース

へのアクセスを簡略化する一方で、ユーザー・アクションへの完全なコントロールを維持することができます。

パスワードが必要なデフォルト・ユーザーを有効化する際には、『データベース設定』の「セキュリティ」ページより”ユーザーは自分のパスワードを変更できる”オプションのチェックをはずすことが推奨されます。

アカウント作成

『デザイン』メニューの「ツールボックス」から新しいアカウントを作成することができます。名前と暫定パスワードを設定するだけです。新しいユーザーがログインすると、ユーザーはパスワードを新しく入力することでそれを変更することができます。

もしもパスワードを忘れた場合には、デザイナーか管理者のみが新しいパスワードを設定できます。

外部認証

4D Server は、Windows の SSO (Single Sign On)を利用することで外部認証に対応しています。それによって、会社の Windows ドメインに(Active Directory を使用して)すでにログインしている場合、ユーザーはパスワードを再入力することなく Windows の 4D アプリケーションにアクセスすることができます。

外部認証を有効にするには、『データベース設定』の「C/S(クライアント-サーバー/ネットワーク)」ページにある”ドメインサーバーによるユーザーの認証”を設定する必要があります。

4D は NTLM と Kerberos の両方のプロトコルに対応しています。使用されるプロトコルは、カレントの設定によって 4D が自動的に選択し、常に可能な限り最も安全なプロトコルを選択します。

注意すべきは、SSO はログインの承認を提供しているに過ぎないということです。このログインをあなたの標準 4D ログイン・メソッドに渡すかどうかはあなた次第です。

カスタム認証

もしも 4D ビルトイン認証システムがあなたの要望に合わない場合には、あなたのユーザーとパスワード・ハッシュを専用のテーブルに保存することで、いつでも独自のカスタマイズされたシステムを構築することができます。4D は、パスワード・ハッシュを保管するため、そして保管したハッシュが与えられたパスワードに適合するかを確認するために、ふたつのセキュリティ用のコマンドを用意しています。そのコマンドとは、**Generate password hash** と **Verify password** です。これらのコマンドを使うと、パスワードそのものはデータベースには保存されないため、高レベルのセキュリティを保証します。さらにふたつのコマンドは、遅くなるように設計されている *bCrypt* アルゴリズムをベースにしています。そのため、総当り攻撃にも耐性があり、潜在的な危機を最小限に留めることができます。

データアクセス・システム

アクセス・コントロールを有効にする

データベース・アクセスの管理はどんな場合でも推奨されますが、特に複数名のユーザーがいる場合は推奨されます。4D の接続コントロールシステムは、ユーザーとグループに基づいています。ユーザーを作成した後、グループにユーザーを入れ、各グループに適切なアクセス権限を割り付けます。グループには特定のオペレーション、例えばテーブル・レコードとテーブル定義のオペレーションなどに対してアクセス権限を与えることができます。

接続システムはデフォルトでは有効ではありませんので、有効化するまでは誰でもあなたのデータベースにアクセスできます。このシステムを有効化するためには、『デザイナー』アカウントにパスワードを割り付ける必要があり、あなたがテーブル、フォーム、メニュー、メソッドなどに付与したすべてのアクセス権限がその後で有効になります。データベースを開くために、ユーザーはパスワードの入力が必要になります。

アクセス権限

4D データベースにはすべて、データへのアクセスをコントロールするために、事前に定義された以下のオプションが含まれます:

- ***デザインアクセス権***：全てのユーザーはアプリケーションにアクセスできます。しかし、デザインモードへのアクセスは制限できます。このオプションは、特定のグループのみデータベースのデザイン環境に入れるようにできます。デフォルトで、デザイナーと管理者のアカウントは常にデザイン環境にアクセスできます。

- ***フォーミュラエディタと4D Write Proドキュメントで使用できるコマンドのフィルタリングとプロジェクト・メソッドの制限***：デフォルトで、4Dはアプリケーション・モードでのコマンド、関数、プロジェクト・メソッドへのアクセスと、4D Write Proドキュメント中で使用された式へのアクセスを制限します。

フォーム・コントロールを通して限定アクセス

フォームへのアクセスとオーナー権限をユーザー・グループに対して設定することによって、フォームを使用してユーザーのデータの閲覧と機能へのアクセスをコントロールすることができます。また、完全にカスタマイズしたインターフェイスを使用して、ユーザー権限に従ったツールと機能をユーザーに対して提供するようにすることもできます。

データ暗号化

4D はデータベース・レベルで – 各テーブルベースで – 暗号化が可能です。暗号化でデータを読めないコード化された形式に変換して、その秘匿性を保護します。正しい有効な暗号キーを使って暗号を解除するまでは、データは判読できないままです。

なぜデータを暗号化するのか

秘匿性やインテグリティ（整合性）に加えて、他にも 4D 暗号化に対処する必要がありました。例えば、GDPRにより厳密には必要ではないけれど、暗号化が推奨される場合。コンテンツが暗号化されるので、クラウド内にバックアップを取るようなプロセスも可能にします。そしてまた、デザインした人にだけ暗号化されたデータへのフルアクセス権限を代表してもらえます。データファイルを暗号化することで、データベースはトータル・セキュリティーを持って外部にホストすることができます。

暗号化キー

最初にデータファイルを暗号化するとき、パスフレーズを与えることが必要です。パスフレーズは暗号化キーの作成に使われます。パスフレーズはパスワードのさらに安全なバージョンです。次のアクションで暗号化キーが要求され、パスフレーズを入力、あるいは暗号キーを含む USB デバイスを接続します。セッションの間、4D はメモリー内に与えられた暗号キーのリストを保持します。

暗号化するテーブルの選択

4D 内でデータを暗号化するときに、最初のステップはストラクチャー・エディターで暗号化したいテーブルをデザインすることです。典型的な方法は、秘密の、あるいは個人データを含むテーブルだけを選択することです。これにより、ターゲットとなるテーブルにセンシティブな情報をグループ化することが奨励されます。

データファイルの暗号化

データファイルの暗号化は、MSC あるいは専用のコマンドを使ってモニターできます。すべてが完全に自動的で、透明性があり、速いのです。

MSC から暗号化

MSC（メンテナンス&セキュリティー・センター）内の新しい「ページの暗号化」では、すべての可能な暗号化アクションを提供します：暗号化、再暗号化、暗号化の削除。

ランゲージから暗号化

MSC に加えて、カスタマイズした暗号化を *Encrypt data file ()*コマンドで管理できます。

ハイレベルのビルトイン・プロテクション

4D SERVER

4D Server は統合されたクライアント/サーバー開発システムで、埋め込まれたデータベース・システムを備えた堅牢なビジネス・アプリケーションを構築するために最適化されています。4D がデータを

（HTTP，SOAP，ODBC，OCI などの標準仕様で）送信したり、あるいは外部から（HTTP，SOAP，ODBC/SQL で）接続できる一方で、メインの通信は内部の 4D ランゲージをベースに、内部の独自のネットワーク・プロトコルを使ったクライアント/サーバー間の通信です。開発ランゲージとネットワーク通信プロトコル間が直接リンクしていることで、SQL インジェクションやバッファー・オーバーフローのような典型的な攻撃シナリオを避けるためのハイレベルのビルトイン・プロテクションを実現しています。

4D ランゲージ

4D ランゲージは強力で成熟した言語であり、ビジネス・アプリケーション・システムを構築するために特化して設計されています。1500 以上のコマンドから成り、データベース操作（order by, query, creating , transactions など）、印刷、他のデバイスやコンピューターとの接続、ドキュメント管理、ウィンドウやユーザー・インターフェイスのコマンドなどをカバーしています。詳細は 4D ランゲージ・マニュアルをご覧ください。
ランゲージそのものはインタープリタ（開発もしくはプロトタイプ）・モードであっても、トークン化されています。単純なテキスト評価として実行されることはありません。

データベースのコンパイル

コードをコンパイルする利点のひとつは、アプリケーションの保護です。4D では、一度データベースをコンパイルすれば、アプリケーション・ビルダーを使って、インタプリタ・コードを消去することができます。この場合、デザイン・モード環境（レコードを除く）へのアクセスはブロックされ、開発用コマンドが無効化されます。

WEB サーバー

4D には*自身のビルトイン HTTP サーバー*の機能があります。静的、動的どちらのコンテンツに対しても強力でかつマルチスレッドのサーバーです。この機能は強固に統合されているため、セキュリティの向上に大きく貢献しています。

よりよいコード・セキュリティ以外に、これで典型的な”アップデートを忘れる”問題が無くなります。すべてが統合されるので、アップデートするべきソフトウェア・パッケージはひとつだけになります（*“ソフトウェアアップデート”セクションでさらに詳細を参照できます*）。標準的なソリューションは、

通常アップデートに際して膨大な量のソフトウェアが必要です：PHP、OpenSSL、Apache、NodeJS など。

どんなものも定期的なアップデートは必要で、ある部分が長い間修正されていないことも普通です。特に専門の IT チームがいない、部門向けソリューションとして使われていた場合はなおさらです。

Web リクエストによって、データベース・レベルだけではなく、アプリケーション・レベルで反応する 4D コードをトリガーすることができます。この緊密な統合によって、ビルトイン認証やカスタマイズされた実装を使いながら、あらゆるリクエストをコントロールできます。もちろん TLS 暗号化もされています（“TLS 暗号化” セクション で詳細を参照）。

SOAP/ WEB サービス・サーバー

HTTP サーバーと同様に、SOAP サーバーはビルトインで、（データベース・レベルだけではなく）ビジネス・オブジェクトを基にした詳細なアクセス・コントロールが可能です。

SQL サーバー

4D リモートからデータアクセスは、デフォルトで 4D 独自のプロトコルを経由する一方、SQL アクセス（ネイティブあるいは ODBC 経由）もサポートされます。さらにオープンソース PDO（PHP Data Objects）ドライバーも可能です。データベース・レベルへの SQL アクセスはパスワード・システムや SQL スキーマを使用して管理可能で、SQL views を使って微調整も可能です。

トリガーを使ってセキュリティを拡張する

トリガーは、テーブル・レコード上で操作イベント（追加、削除、修正）が発生したとき必ず 4D データベース・エンジンによって自動的に実行されます。トリガーはとても強力なツールで、不正な操作や過失によるデータ損失/改ざんから保護してくれるだけでなく、テーブル操作の制限もできます。例えば、請求書システムにおいて、請求先が指定されていない場合には請求書を追加できない、などと言ったように使用することができます。

削除コントロール

この技術は、いわゆる*参照整合性*を強化します。これはテーブル間のリレーションが常に整合性が取れていなければならないということを意味します。外部のキー・フィールドは、外部キーが参照するプライマリ・キーと一致しなければなりません。このように、削除コントロールオプションは、*N 対 1* リレーションの *1* 側のテーブルでレコードが削除されたときに、*N* 側のテーブルでのレコード削除を規制します。通常は、ユーザーはカレントテーブルのレコードのみ削除可能です。

ネットワーク・セキュリティー

TLS 暗号化を可能に

TLS は、コンピュータネットワーク上の安全な通信を提供する暗号プロトコルです。その主な目的は、二つのコンピュータ・アプリケーション間の通信のプライバシーを保護することと、データを損なわないようにすることです。

4D Web サーバーは TLS 経由で安全に通信ができます。暗号化されていない通信は避けるよう強く推奨されていて、安全ではない方法で送られてきたリクエストを、拒否あるいはリダイレクトするのはデベロッパーの責任です。必要なものは TLS 認証だけで、これを購入するか、あるいは Let's encrypt のようなオープンの認証認定から入手する必要があります。

4D Server において TLS を使っている場合、クライアントとの接続は、ネットワークをモニタリングしているソフトウェアなどがデータを受け取った場合でも見られなくする手法で安全が確保されています。

交換した情報は、事前に定義したキー（SSL 定義は不要）か、あるいは 4D Web Server 同様（購入あるいは無料の）TLS 認証によって安全を確保できます。

4D クライアント/サーバー通信の暗号化を実行するには、『データベース設定』ページの「C/S」中の「クライアント＝サーバー/ネットワークオプション」で、オプションの「クライアント−サーバー接続の暗号化」をチェックします。同じことを web 接続で行うには, 『データベース設定』ページの「Web/設定」で、「SSL(TLS)を有効にする」にチェックを付けます。

PFS (PERFECT FORWARD SECRECY)を有効に

Perfect Forward Secrecy (PFS)は、キー合意プロトコルの属性で、長い語句のキーから派生したセッションキーの組み合わせで、将来的にこれらの一つが侵入されたとしても、侵入は不可能であることは確実です。

PFS は 4D Web サーバーの初期設定では無効になっています。PFS を有効にするには、TLS プロトコルが使用でき、暗号リストに ECDHE か DHE（初期設定では true）を含むことが必要です。PFS は DH パラメータファイルを使います。このファイルはもし存在しなければ 4D が自動的に作成します。

HSTS(HTTP STRICT TRANSPORT SECURITY)を有効に

HTTP (Strict Transport Security) –**ベータ版**として提供 – は、HTTPS に代わる通信が必要な安全ポリシーで、HTTP との通信を防ぎます。HSTS は、全体の通信チャンネルがデータを送る前に暗号化することを保証します。従って、送信中のデータを読んだり、修正したりする攻撃をブロックします。

バックアップとログ・システム

トランザクション・ベースのログ

4D は、その日から使えるトランザクション・ベースのログ・システムを提供します。全てのデータ修正操作は記録されており、必要な場合にロールバック（元に戻す）ができます。

緊急の場合でも、トランザクションは復元可能で、何も喪失することはありません。中断があった場合、再起動時にデータベースは自動的に検査され、欠けている操作（例：メモリーにあって、ディスクには保存されていない操作など）は復元され、データベースを以前の状態に戻せます。データ全体の破損（例：ディスク不良など）の場合、データファイルは自動的に最後のフル・バックアップとジャーナル（トランザクションのログ・ファイル）から復元され、最新の仕事までが統合されます。

ジャーナルはまた、事故で消去してしまった場合にも有効です。法医学的にも、データの復旧という点でも。

標準的なバックアップは、4D 製品の一部であり、ライセンスの追加は不要で、必要なのは（ディスク不良に備えた）追加のハードディスクだけです。

24 時間無休の環境では、4D は直列のミラーシステム、またはスターミラーシステムの使用をサポートしています。製品、ミラー、そしてそのまたミラーはクラスターのようなシステムを築き、絶え間ないサービスを提供します。追加のミラーシステムは重大な災害があったときでもデータを保護できるように、別の都市(あるいはクラウド上)で稼働させることも可能です。

仮想マシン・スナップショット（VSS WRITER）

トランザクション・ベースのログに加えて、4D は仮想マシンのスナップショットも対応します。専用の *VSS writer* アプリケーションが自動的に、Windows のボリューム・シャドウ・コピーサービス(VSS)を通して送られる仮想アプリケーションに対してスナップショットのリクエストを管理します。

4D は自動的に VSS ライター・サービスをインストールします。仮想マシン・アドミニストレーターが、例えば VMWare を使ってスナップショットを始めたとき、VMWare はゲストへ情報を伝え、次に VSS ライターへ伝え、その後 4D Server に伝わり、4D はキャッシュをフラッシュして1秒間待機します。それから OS に.4DD、.4DIndx と.Journal ファイルがひとつのグループであり、同時にスナップショットされるよう要求します。OS がこれを管理し、4D Server はクライアントからのリクエストの管理を継続できます。スナップショットによって被る”フリーズ”が続くのは1秒ほどなので、接続したユーザーにはほとんど気づかれません。
4D Server のアドミニストレータ・ウィンドウのモニターページ は、*アプリケーション情報エリア*に VSS ライター・サービスの状況を表示します。
*注意：スナップショットはバックアップは**更新しません**。*

ソフトウェア・アップデート

オールインワン・アップデート

今日のソフトウェアは、ソフトウェア製品、データベース・サービス、ミドルウェア、アプリケーション・サーバー、Web サーバーなどの複雑な組み合わせであることがあります。例えば OpenSSL DLL のように、更新することを忘れることがあるのは当たり前です。それ以上に、人々はソフトウェアの

アップデートを忘れて(あるいはリマインダーを無視して)います。時間がかかるので理解できることではあります。

4D はこの問題を多くの方法で削減しています。管理者の日常生活を助けるだけではなく、設計によってリスクを最小限にしています。

4D はオールインワン・ソリューションとして統合されているので、すべてがひとつのフォルダーにインストールされています。単純なドラッグ &ドロップ操作で置き換えられるので、ビジネス・アプリケーションをひとつのステップでアップデート可能です。何かを忘れることはありません。

サーバーのアップデートは完全に自動化できます。 操作は 4D そのものが管理や強制をするわけではなく、完全にソリューション・デベロッパーの手に委ねられています。

クライアント最小バージョン

クライアント/サーバー・アプリケーションの中では、サーバー側で、接続できるクライアントの最低バージョンを指定できます。例えば、サーバーをアップデートしたとき、(安全上の理由から)古いクライアントの接続を許可したくない場合にこれは有用です。この場合、クライアント・アプリケーションの古いバージョンからの接続は拒否され、アップグレードを催促することができます(これも自動的に実行できます。)

他のセキュリティを考える

強力なパスワードを使う

セキュリティ・システムの最初のステップは、強力なパスワードの使用です。あなたのサーバーの周辺にいかに多くの保護システムが置かれようとも、あなたのパスワードを知られてしまえばどこからでもアクセスされてしまいます。一般的な組み合わせを推測したり、あるいはランダムに可能性を作ってすべてを試したり、その両方によって、パスワードを確定しようとするプログラムが数え切れないほどあります。

最良の防衛は強力なパスワード、たとえば数字、大文字、小文字、そしてできれば他の文字と組み合わせたパスワードを使用することです。これで時間をかけてパスワードを推測するのがほぼ不可能となります。

接続システムのテスト

一度あなたのアカウントが設定されたら、接続システムの定義と権限のテストを検討すべきです。継続的にシステムを評価することで、データが保護されていることを確認できます。特に新しい機能が追加された場合に必要です。

ハードドライブの暗号化

完全に暗号化されたハードドライブ も強力な防護となります。暗号化した SSD のようなハードウェア・ソリューションや BitLocker (Windows)や FileFault (Mac)のようなソフトウェアを利用するのはどんな場合でもよいアイデアと言えます。自己暗号化ドライブは SSD のコントローラーにビルトインされた暗号化エンジンを使って、すべてのファイルを暗号化します。SSD の本質的に速い技術によって、ハードドライブよりも相当速く毎日の保存タスクを実行できる一方で、同時にデータのセキュリティを改良することもできます。