CHAPTER 2

データベース アクセスのセットアップ

データベースへの読み取り専用アクセスを取得するには、最初にアクセスを許可するようにアプライアンスを設定する必要があります。次に、JDBC ドライバをダウンロードし、アクセスするアプライアンスからの SSL 証明書を受け入れることで、アプライアンスに接続するクライアント システムを設定する必要があります。最後に、アプライアンスに接続するようにレポート アプリケーションを設定します。

コメント　データベース アクセスをセットアップする前に、前提条件（1-12 ページ）で説明されている前提条件を満たす必要があります。

詳細については、次の項を参照してください。

データベース ユーザ アカウントの作成

ライセンス：すべて

Firepower システム データベースへのアクセスを設定するには、最初にユーザ アカウントを作成し、Firepower システム データベースにアクセスするための権限をそのアカウントに割り当てる必要があります。この権限を付与するには、システム提供の External Database User ユーザ ロール、または組織が作成し、External Database User 権限が含まれているカスタム ユーザ ロールを、このアカウントに割り当てます。ユーザ アカウントの作成と、特定のユーザ ロールの権限の確認については、『*Firepower Management Center Configuration Guide*』を参照してください。

警告　**External Database Access はグローバル権限です。External Database Access が付与されているユーザは、すべてのドメインに対して情報を照会できます。**

ヒント　システム提供の Administrator ロールが割り当てられているユーザには、デフォルトで External Database User 権限が付与されています。

マルチドメイン型展開では、Admin アクセス権限があるドメインでユーザ アカウントを作成できます。ただし、External Database User ロールはグローバル ドメイン レベルでのみ使用可能です。External Database User は、ドメインに関係なくすべてのイベントにアクセスできます。

Firepower Management Center でのデータベースアクセスの有効化

ライセンス: すべて

外部データベース ユーザの作成後に、アプライアンスのデータベースへのアクセスを許可するように Firepower Management Center を設定する必要があります。また、アプライアンスでデータベース アクセス リストを設定し、外部データベースを照会するすべてのホスト IP アドレスを追加する必要があります。

データベース アクセスを有効にする方法:

アクセス: 管理

ステップ 1　Firepower Management Center で [System] > [Configuration] を選択します。

ステップ 2　左側の [External Database Access] をクリックします。

[Database Settings] メニューが表示されます。

ステップ 3　[Allow External Database Access] チェックボックスをオンにします。

[アクセスリスト（Access List）] フィールドが表示されます。

ステップ 4　サードパーティ アプリケーションの要件に応じて、[Server Hostname] フィールドに Firepower Management Center の完全修飾ドメイン名（FQDN）または IPv4 アドレスを入力します。証明書をインストールするときに IPv6 アドレスを使用できないため、IPv6 アドレスは使用できません。

FQDN を入力する場合は、クライアントが Firepower Management Center の FQDN を解決できることを確認する必要があります。IP アドレスを入力する場合は、クライアントがその IP アドレスを使用して Firepower Management Center に接続できることを確認する必要があります。

ステップ 5　1 つ以上の IP アドレスからのデータベース アクセスを追加するため、[Add Hosts] をクリックします。

[アクセスリスト（Access List）] フィールドに [IP アドレス（IP Address）] フィールドが表示されます。

ステップ 6　[IP Address] フィールドでは、追加する IP アドレスに応じて次のいずれかを入力できます。

- 正確な IPv4 アドレス（`192.168.1.101` など）
- 正確な IPv6 アドレス（`2001:DB8::4` など）

- IP アドレス範囲。
  - Firepower システムで IP アドレス範囲を使用する方法については、『*Firepower Management Center Configuration Guide*』の IP アドレス規則を参照してください。
- **any**（任意の IP アドレスを指定）

ステップ 7　[追加（Add）] をクリックします。

IP アドレスがデータベース アクセス リストに追加されます。

ステップ 8　必要に応じてデータベース アクセス リストのエントリを削除するには、削除アイコン（ ）をクリックします。

ステップ 9　[Save（保存）] をクリックします。

データベース アクセス設定が保存されます。

ステップ 10　次の項（JDBC ドライバのダウンロード）の手順に進みます。

JDBC ドライバのダウンロード

ライセンス：すべて

外部データベース ユーザを作成し、Firepower Management Center でデータベース アクセスを許可するように設定したら、JDBC ドライバをクライアント システムにダウンロードします。データベースに接続するときに、この JDBC ドライバを使用する必要があります。

JDBC ドライバをダウンロードするには、次の手順を実行します。

アクセス：Admin

ステップ 1　Firepower Management Center で [System] > [Configuration] を選択します。

ステップ 2　左側の [External Database Access] をクリックします。

[Database Settings] メニューが表示されます。

ステップ 3　[Client JDBC Driver] の横にある [Download] をクリックし、ブラウザのプロンプトに従って **client.zip** パッケージをダウンロードします。

ステップ 4　ZIP パッケージを解凍します。場所を書きとめておきます。
パッケージのファイル構造を保持してください。
ドライバはその他のファイルとと共に ZIP ファイル（**client.zip**）に含まれています。パッケージのディレクトリ構造は次のとおりです。

- **bin**：RunQuery と呼ばれるサンプル クライアントと、クライアントと Firepower Management Center の間での暗号化通信用証明書をインストールするときに使用する実行ファイルが含まれています。
- **lib**：JDBC ドライバ JAR ファイルが含まれています。
- **src**：**bin** ディレクトリ内の実行ファイルのソース コードが含まれています。

ステップ 5　次の項（クライアント SSL 証明書のインストール）の手順に進みます。

クライアント SSL 証明書のインストール

JDBC ドライバのダウンロード後に、システム提供プログラム InstallCert を使用して Firepower Management Center からの SSL 証明書を受け入れ、インストールします。クライアント システムと Firepower Management Center は SSL 証明書認証を使用して安全に通信します。証明書を受け入れる場合、現在実行されている JRE の **security** ディレクトリ内のキーストア（**jssecacerts**）に証明書が追加されます。

```
$JAVA_HOME/jre[version]/lib/security
```

次に、Microsoft Windows と UNIX を実行しているコンピュータでのキーストアの一般的な場所を示します。

- `C:\Program Files\Java\jre[version]\lib\security\jssecacerts`
- `/var/jre[version]/lib/security/jssecacerts`

コメント　データベース アクセス機能にアクセスするために使用する予定の Java クエリ アプリケーションが異なる JRE を使用している場合は、その別の JRE の **security** ディレクトリにキーストアをコピーする必要があります。

InstallCert を使用して SSL をインストールするには、次の手順を実行します。

ステップ 1　コンピュータでコマンドライン インターフェイスを開きます。

ステップ 2　コマンド プロンプトから、ZIP パッケージの解凍時に作成された **bin** にディレクトリを変更します。

ステップ 3　Firepower Management Center の SSL 証明書をインストールするには、次のように入力して Enter キーを押します。

```
java InstallCert defense_center
```

defense_center は Firepower Management Center の FQDN または IP アドレスです。InstallCert は IPv6 アドレスをサポートしていません。IPv6 ネットワークでは、解決可能なホスト名を使用する必要があります。

Microsoft Windows を実行しているコンピュータでは次の例に示すような出力が表示されます。

```
Loading KeyStore C:\Program Files\Java\jre6\lib\security...
Opening connection to defensecenter.example.com:2000...
Starting SSL handshake...
Subject GENERATION=server, T=vjdbc, O="シスコ, Inc.",
...
...
```

証明書を表示するように求められます。

ステップ 4　オプションで、証明書を表示します。

証明書を受け入れるように求められます。

ステップ 5　証明書を受け入れます。

証明書が受け入れられると、Microsoft Windows を実行しているコンピュータから次の例のような出力が表示されます。

```
Added certificate to keystore 'C:\Program Files\Java\jre6\lib\security\jssecacerts'
using alias 'defensecenter.example.com-1'
```

Crystal Reports を使用する予定の場合は、キーストア（jssecacerts）の場所を書きとめておきます。この情報は、後で必要になります。

ステップ 6　次の選択肢があります。

- サードパーティ アプリケーションを使用する場合は、次の項（サードパーティ アプリケーションを使用したデータベースへの接続（2-5 ページ））の手順に進みます。
- カスタム アプリケーションを使用する場合は、カスタム プログラムを使用したデータベースへの接続（2-7 ページ）の手順に進みます。

サードパーティ アプリケーションを使用したデータベースへの接続

証明書のインストール後、JDBC SSL 接続をサポートするサードパーティ クライアントを使用して、Firepower Management Center でデータベースを照会できます。次の表に、クライアントと Firepower Management Center 間の接続を設定するために必要な情報を示します。

表 2-1　データベース アクセス クライアントの接続情報

情報	説明
JDBC URL	次に示す JDBC URL によって Firepower システム データベースが識別されるため、クライアントの JDBC ドライバがそのデータベースとの接続を確立できます。 `jdbc:vjdbc:rmi://defense_center:2000/VJdbc,eqe` *`defense_center`* は Firepower Management Center の FQDN または IP アドレスです。
JDBC ドライバ JAR ファイル	Firepower システム データベースへの接続を設定するときには次に示す JAR ファイルを使用する必要があります。 • `vjdbc.jar` • `commons-logging-1.1.jar` これらのファイルは、JDBC ドライバのダウンロード（2-3 ページ）の説明に従い **`client.zip`** ファイルをダウンロードして解凍した **`lib`** サブディレクトリにあります。
JDBC ドライバ クラス	Firepower システム データベースへの接続を設定するときには次に示すドライバ クラスを使用する必要があります。 `com.sourcefire.vjdbc.VirtualDriver`
ユーザ名とパスワード	アプライアンスのデータベースに接続するには、External Database User 権限が付与されているユーザ アカウントを使用します。詳細については、データベース ユーザ アカウントの作成（2-1 ページ）を参照してください。

以降のセクションでは、広く使用されている 3 つの業界標準レポート ツールを使用して、Firepower システム データベースに接続する際のヒントを説明します。これらのツールまたは他の Java ベースのアプリケーションのいずれを使用するかに関係なく、ご使用のレポート ツールのドキュメントで、JDBC SSL 接続の詳しい作成手順を参照してください。

Crystal Reports

以下の情報は 32 ビット Windows 環境への Crystal Reports 2011 のインストールに適用されます。64 ビット Windows 環境を実行している場合はファイルパスが異なる可能性があります。

Crystal Reports 2011 が Firepower システム データベースに接続できるようにするには、次の操作が必要です。

- Firepower Management Center からダウンロードした JDBC ドライバ JAR ファイルを Crystal Reports クラスパスに追加します。Crystal Reports のデフォルト インストールの場合、次に示すファイルでクラスパス セクションを編集できます。
  ```
  C:\Program Files\SAP BusinessObjects\SAP Business Objects
  Enterprise XI 4.0\Java\CRConfig.xml
  ```
- クライアント SSL 証明書のインストール時に作成したキーストアを、適切な Crystal Reports セキュリティ ディレクトリにコピーします。Crystal Reports のデフォルト インストールの場合、ディレクトリは次のとおりです。
  ```
  C:\Program Files\SAP BusinessObjects\SAP Business Objects
  Enterprise XI 4.0\win32_x86\jdk\jre\lib\security
  ```
- シスコ をデータベース名として使用し、Database Expert との新しい JDBC（JNDI）の接続を作成します。

JasperSoft iReport

iReport が Firepower システム データベースに接続できるようにするには、次の操作が必要です。

- Firepower Management Center からダウンロードした JDBC ドライバ JAR ファイルを iReport クラスパスに追加します。
- Firepower Management Center からダウンロードした JDBC ドライバ JAR ファイルを使用して 新しい JDBC ドライバを追加します。ドライバ ファイルの追加後は、iReport は正しいドライバ クラスを検出するはずです。
- 作成したドライバを使用して、新しいデータベース接続を作成します。

Actuate BIRT

BIRT が Firepower システム データベースに接続できるようにするには、次の操作が必要です。

- [Generic JDBC Driver] テンプレートを使用してドライバ定義を追加します。
- [Generic JDBC] プロファイル タイプを使用して新しいデータベース接続を作成します。
- [JDBC Data Source] データ ソース タイプを使用してレポートのデータ ソースを作成します。

ヒント　新しい JDBC データ ソース プロファイルの作成時に シスコ ドライバ クラスを選択できない場合は、Firepower Management Center からダウンロードした JDBC ドライバ JAR ファイルを使用してドライバを追加します。

カスタム プログラムを使用したデータベースへの接続

証明書をインストールしたら、カスタム Java レポート ツールが Firepower システム データベースを照会できるように設定できます。シスコ には、RunQuery という名前のサンプル Java コマンド ライン アプリケーションがあります。このアプリケーションは、Firepower Management Center に含まれている JDBC ドライバを使用して必要な SSL 接続を確立します。RunQuery は、テーブル レコードとテーブル メタデータの両方を取得します。ソース コードは、Firepower Management Center からダウンロードした ZIP パッケージの `src` ディレクトリに含まれています。JDBC ドライバのダウンロード（2-3 ページ）を参照してください。

コメント　RunQuery はサンプル クライアントであり、フル機能のレポート ツール**ではありません**。シスコでは、データベース照会の主な手段として、このツールを使用しないことを**強く**推奨します。RunQuery の使用法については、ZIP パッケージに含まれている README ファイルを参照してください。

カスタム プログラムを使用したデータベースへの接続に関する詳細については、次の項を参照してください。

- カスタム Java プログラムのサンプル コード（2-7 ページ）：RunQuery アプリケーションがデータベース接続のセットアップとクエリの送信に使用する Java クラスとメソッドについて説明します。
- アプリケーションの実行（2-8 ページ）：Java アプリケーションを実行するための環境要件について説明します。

カスタム Java プログラムのサンプル コード

RunQuery ソース コードでは、後述する関数が使用されます。次のコード例に、可能な実装方法の 1 つを示します。

SSL プロバイダー接続の動的な設定

クライアントに SSL セキュリティ証明書をインストールしたら（クライアント SSL 証明書のインストール（2-4 ページ）を参照）、プログラムで次の行を使用して JSSE プロバイダーを動的に登録できます。

```
Security.addProvider(new com.sun.net.ssl.internal.ssl.(Provider());
```

プログラムの JDBC ドライバの初期設定

`Class.forName()` メソッドを使用して Java アプリケーションに JDBC ドライバ クラスを次のようにロードできます。

```
Class.forName("com.sourcefire.vjdbc.VirtualDriver").newInstance();
```

コマンド ラインから起動するプログラムの場合は、ユーザが次のように JDBC クラスを指定できます。

```
java -Djdbc.drivers="com.sourcefire.vjdbc.VirtualDriver" program_name ...
```

program_name はプログラムの名前です。

データベースへのプログラムの接続

プログラムがクエリを送信する前に、プログラムで JDBC 接続オブジェクトを取得する必要があります。接続を確立して接続オブジェクトを取得するには、次のように DriverManager.getConnection メソッドを使用します。

```
Connection conn = DriverManager.getConnection("jdbc:vjdbc:rmi://my_dc:2000/VJdbc,eqe",
"user", "password");
```

my_dc は Firepower Management Center の FQDN または IP アドレス、user はデータベース アクセス ユーザ アカウント名、*password* はアカウント パスワードです。

シスコ テーブルのデータの照会

クエリを送信し、取得したレコードを結果セットに割り当てるため、SQL クエリ オブジェクトを次のように作成します。

```
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("sql");
```

sql は SQL クエリです。サポートされている SQL 関数については、データベースの照会（2-9 ページ）を参照してください。

テーブル クエリの結果の作成

上記のクエリにより生成される結果セット（rs）から、次のようにフィールドを出力できます。

```
while(rs.next())
{
    for(int i=1; i<= md.getColumnCount(); i++)
    {
        System.out.print(rs.getString(i) + " ");
    }
    System.out.print("\n");
}
```

スキーマ情報の取得

プログラムは、データベースのテーブルを次のようにリストできます。

```
DatabaseMetaData metaData = conn.getMetaData();
      ResultSet tables = meta.getTables(null, null, null, null);
      while (tables.next())
      {
System.out.println(tables.getString("TABLE_NAME"));
      }
```

プログラムは、テーブルの列を次のようにリストできます。

```
ResultSet columns = metaData.getColumns(null, null, "table_name", null);
```

table_name は、データベース テーブルの名前です。

アプリケーションの実行

アプリケーションを実行する前に、クライアント コンピュータの CLASSPATH を設定し、アプリケーションの JAR ファイルの場所と現在のディレクトリを組み込む必要があります。

JDBC ドライバのダウンロード（2-3 ページ）の説明に従ってデータベース アクセスの ZIP パッケージをダウンロードして解凍した場合は、CLASSPATH を次のように更新します。

UNIX 環境でアプリケーションを実行するには:

ステップ 1　次のコマンドを使用します。

```
export CLASSPATH=$CLASSPATH:.:path/lib/vjdbc.jar:path/lib/commons-logging-1.1.jar
```

path は、Firepower Management Center からダウンロードした ZIP パッケージを解凍したディレクトリ パスです。

Windows 7 環境でアプリケーションを実行するには:

ステップ 1　[Computer] アイコンを右クリックして [Properties] を選択します。

[System] ウィンドウが表示されます。

ステップ 2　[Advanced System Settings] をクリックします。

[System Properties] ウィンドウが表示されます。

ステップ 3　[Advanced] タブを選択します。

ステップ 4　[Environment Variables...] をクリックします。

[Environment Variables] ウィンドウが表示されます。

ステップ 5　**CLASSPATH** システム変数を選択して [Edit...] をクリックします。

[Edit System Variable] ウィンドウが表示されます。

ステップ 6　[Variable value:] フィールドに次を追加します。

```
.;path\bin;.;path\lib\vjdbc.jar;.;path\lib\commons-logging-1.1.jar;.;path\lib
```

path は、Firepower Management Center からダウンロードした ZIP パッケージを解凍したディレクトリ パスです。

ステップ 7　[OK] をクリックして、値を保存します。

[Environment Variables] ウィンドウが表示されます。

ステップ 8　[OK] をクリックして、値を保存します。これでアプリケーションを実行できます。

データベースの照会

External Database Access はグローバル権限です。クエリで制限していない限り、結果はデータベース内のすべてのドメインを対象としています。

以降のセクションでは、サポートされているクエリの構文と、クエリに関連するその他の重要な要件および制約事項について説明します。

- サポートされている SHOW ステートメント構文（2-10 ページ）では、Firepower システム データベース照会のためにサポートされている MySQL **SHOW** ステートメント構文について説明します。
- サポートされている DESCRIBE または DESC ステートメント構文（2-10 ページ）では、Firepower システム データベース照会のためにサポートされている MySQL **DESCRIBE** ステートメント構文について説明します。
- サポートされている SELECT ステートメント構文（2-11 ページ）では、Firepower システム データベース照会のためにサポートされている MySQL **SELECT** ステートメント構文について説明します。

- 結合の制約（2-12 ページ）では、Firepower システム データベース照会のためにサポートされている結合と、任意のテーブルに対して許可されている結合に関する情報の確認方法を説明します。
- 使い慣れていない形式で保存されているデータの照会（2-13 ページ）では、使い慣れていない形式で保存されているデータ（UNIX タイムスタンプおよび IP アドレスを含む）に対してクエリを実行し、クエリが正常に完了し、予期しているとおりの結果が表示されるようにする方法について説明します。
- パフォーマンス上の理由によるクエリの制限（2-14 ページ）では、クエリに対する制約を適用する場合に、Firepower システム のパフォーマンスを低下させないための推奨事項について説明します。
- クエリのヒント（2-15 ページ）では、さまざまなアプライアンスで侵入イベントを照会する場合のヒントについて説明します。

スキーマの情報と許可されている結合については、以降の章を参照してください。

- スキーマ：システム レベル テーブル（3-1 ページ）
- スキーマ：侵入テーブル（4-1 ページ）
- スキーマ：統計情報追跡テーブル（5-1 ページ）
- スキーマ：検出イベントおよびネットワーク マップのテーブル（6-1 ページ）
- スキーマ：接続ログ テーブル（7-1 ページ）
- スキーマ：ユーザ アクティビティ テーブル（8-1 ページ）
- スキーマ：相関テーブル（9-1 ページ）

サポートされている SHOW ステートメント構文

SHOW ステートメントは、Firepower システム データベースのすべてのテーブルをリストします。Firepower システム データベースの照会に使用できる、サポートされている MySQL **SHOW** ステートメント構文を次に示します。

```
SHOW TABLES;
```

上記にない **SHOW** ステートメント構文はサポートされていません。

サポートされている DESCRIBE または DESC ステートメント構文

Firepower システム データベースでは、**DESCRIBE** ステートメントを限定的に使用できます。Firepower システム データベースでは、DESCRIBE ステートメントの出力には列の名前と各列のデータのタイプだけがリストされます。Firepower システム データベースの照会に使用できる、サポートされている MySQL **DESCRIBE** ステートメント構文を次に示します。

```
DESCRIBE table_name;
```

Firepower システム データベースでは、これと同一のコマンド DESC もサポートされています。

```
DESC table_name;
```

表 2-2　サポートされている DESCRIBE ステートメントの構文

項目	説明
table_name	照会するテーブルの名前

上記にない **DESCRIBE** ステートメント構文はサポートされていません。特に、Firepower システム データベース アクセス機能では次の要素はサポートされていません。

- INDEX FOR 句
- TABLE 句
- PROCEDURE 句

サポートされている SELECT ステートメント構文

Firepower システム データベースの照会に使用できる、サポートされている MySQL SELECT ステートメント構文を次に示します。

```
SELECT
[ALL | DISTINCT]
[COUNT ( field ) | COUNT (*) ]

select_expr [, select_expr ...]
FROM table_references
[WHERE where_condition]
[GROUP BY { column_name | position } [ ASC | DESC ], …]
[HAVING where_condition]
[ORDER BY { column_name | position } [ ASC | DESC ], …]
[LIMIT { [offset,] row_count | row_count OFFSET offset}]
```

次の表に、上記の SELECT ステートメントの句と引数の必須構文について詳しく説明します。

表 2-3　サポートされている SELECT ステートメント構文

項目	説明
select_expr	{*column_name* [[AS] *alias*] | *function*(...) [[AS] *alias*] | *aggregate_function*(...) [[AS] *alias*]}
column_name	照会するフィールドの名前
機能	{ABS | CAST | CEILING | CHAR_LENGTH | COALESCE | CONV | CHARACTER_LENGTH | CONCAT | CONVERT | COUNT | CURRENT_DATE | CURRENT_TIME | CURRENT_TIMESTAMP | EXTRACT | FLOOR | HEX | INET_ATON | INET_NTOA | INET6_ATON | INET6_NTOA | LEFT | LOWER | LPAD | MID | MOD | NULLIF | OCTET_LENGTH | POSITION | RIGHT | ROUND | SUBSTRING | SYSDATE | TIME | TIMESTAMP | TRIM | UPPER}
aggregate_function	{AVG | COUNT | COUNT(DISTINCT) | MAX | MIN | SUM}
field	関数の実行対象フィールドの名前
table_references	次のいずれかを入力します。 • *table_reference* INNER JOIN *table_reference* join_condition • *table_reference* LEFT [OUTER] JOIN *table_reference* join_condition
table_reference	*table_name* [[AS] alias]
table_name	照会するテーブルの名前
join_condition	ON *conditional_expr*

表 2-3　サポートされている SELECT ステートメント構文(続き)

項目	説明
conditional_expr	結合に対応したフィールド値の等価比較。詳細については、結合の制約(2-12 ページ)を参照してください。
where_condition	次のいずれかを入力します。 • IS NULL または IS NOT NULL • NOT, ! • BETWEEN ... AND ... • LIKE • =, !=, <>, >, >=, <, <=

サポートされている MySQL 構文の記述方法に慣れていない場合は、次の表でヒントを参照してください。

表 2-4　MySQL 構文の形式

記号	表記	意味
角カッコ	[]	オプションの句または引数
波カッコ	{}	必要な句または引数
パイプ	|	句または引数の選択

上記にない SELECT ステートメント構文はサポートされていません。特に、Firepower システム データベース アクセス機能では次の要素はサポートされていません。

- SELECT *:フィールドを明示的に指定する必要があります。
- ユニオン
- サブクエリ
- GROUP BY 句の WITH ROLLUP 修飾子
- INTO 句
- FOR UPDATE 句

結合の制約

パフォーマンスおよびその他の実際的な理由から、Firepower システム データベースのテーブルに対して実行できる結合は限定されています。シスコ では、その結果がイベント分析に役立つ可能性のない結合は実行できません。

内部結合または左(外部)結合だけを実行できます。入れ子になった結合、クロス結合、自然結合、右(外部)結合、フル(外部)結合、および USING 句を使用した結合はサポート**されていません**。

スキーマに関するドキュメントでは、各テーブルでサポートされている結合が示されています。リストされていない結合はサポートされていません。たとえば、**compliance_event** テーブルと **intrusion_event** テーブルを IP アドレス フィールドで結合することはできません。これは、両方のテーブルに IP アドレス情報が含まれている場合でも同様です。また、廃止されたテーブルと廃止されたフィールドでの結合はリストされていません。

使い慣れていない形式で保存されているデータの照会

Firepower システム データベースでは、表示に適していない形式で一部のデータが保存されています。以降の項では、さまざまなフィールドに対してクエリを実行し、クエリが正常に完了し、予期しているとおりの結果が表示されるようにする方法について詳しく説明します。

IPv6 形式のアドレス

Firepower システム データベースには、IPv6 アドレスがバイナリ形式で保存されます。結果を 16 進数形式で表示するには、**HEX()** 関数を使用します。特定の IPv6 アドレスのデータベースを照会するには、**UNHEX()** 関数を使用します。

たとえば、モニタ対象セッションに関する情報を含む **connection_log** テーブルを照会し、その照会を特定の IPv6 アドレスで制限するステートメントを次に示します。

```
SELECT HEX(initiator_ip), HEX(responder_ip), packets_sent, bytes_sent

FROM connection_log

WHERE initiator_ip = UNHEX('20010db8000000000000000000004321');
```

IPv4 アドレス

Firepower システム データベースでは、IPv6 アドレスと同じフィールドに、IPv4 アドレスがバイナリ形式で保存されます。IPv6 アドレスと同様に、16 進数形式で表示する場合は `HEX()` 関数を使用します。データベースは RFC に準拠するためにビット 80 ～ 95 に 1 を取り込みますが、これによって無効な IPv6 アドレスが生成されます。たとえば IPv4 アドレス 10.5.15.1 は `00000000000000000000FFFF0A050F01` として保存されます。

MAC アドレス

Firepower システム データベースには、MAC アドレスがバイナリ形式で保存されます。結果を 16 進数形式で表示するには、`HEX()` 関数を使用します。

たとえば、次のステートメントは **rna_host_mac_map** テーブルを照会します。このテーブルには、IP アドレスでは識別されなかったホストと その MAC アドレスに関する情報が含まれており、照会で取得できるホストが最初の 5 つのホストに限定されます。

```
SELECT HEX(host_id), HEX(mac_address)

FROM rna_host_mac_map

LIMIT 5;
```

パケット データ

Firepower システム データベースは、侵入イベントのパケット データをバイナリ形式で保存します。結果を 16 進数形式で表示するには、HEX() 関数を使用します。

たとえば、次のステートメントは **intrusion_event_packet** テーブルを照会し、特定イベントのパケット データを取得します。

```
SELECT HEX(packet_data)
FROM intrusion_event_packet
WHERE event_id = 1234;
```

UNIX タイムスタンプ

Firepower システム データベースではほとんどのタイムスタンプが UNIX タイムスタンプとして保存されます。UNIX タイムスタンプは、1970 年 1 月 1 日 00:00:00（UTC）以降の経過秒数を表します。現地時間で結果を表示するには、**FROM_UNIXTIME()** 関数を使用します。

たとえば、次のステートメントは **audit_log** テーブルを照会し、最大 25 件の結果を返します。このテーブルには、アプライアンスの Web インターフェイスでのユーザ アクションがすべて記録されています。

```
SELECT FROM_UNIXTIME(action_time_sec), user, message
FROM audit_log
LIMIT 0, 25;
```

データベースのすべての時刻は UTC の時刻であることに注意してください。**CONVERT_TZ()** 関数を使用できますが、この関数は結果を UTC で返します。

一部のイベントには、マイクロ秒の精度が関連付けられている点に注意してください。UNIX タイムスタンプとマイクロ秒の増分を連結するには、**CONCAT()** 関数と **LPAD()** 関数を使用します。たとえば、次のステートメントは **intrusion_event** テーブルを照会します。

```
SELECT CONCAT(FROM_UNIXTIME(event_time_sec)), '.', LPAD (event_time_usec, 6, '0')),
HEX(host_id),
rule_message
FROM intrusion_event
LIMIT 0, 25;
```

データベースに対し、特定の UNIX タイムスタンプのイベントを照会するには、**UNIX_TIMESTAMP()** 関数を使用します。

パフォーマンス上の理由によるクエリの制限

Firepower システム データベース テーブルに対して実行できる結合はシステムにより制限されていますが、一部のコストがかかるクエリ（Firepower Management Center のパフォーマンスに悪影響を及ぼす可能性のあるクエリ）が許可されます。

したがって、大規模なテーブルの場合は結果セットを制限するようにします。戦略としては次のものがあります。

- 特定のリーフ ドメインにクエリを制限する
- 特定の時間範囲にクエリを制限する
- IP アドレスによりクエリを制限する
- **LIMIT** 句を使用する

展開によっては、多数のテーブルを照会する際に結果セットを制限する必要があることがあります。特に次のテーブルには、DC3000 での最大 1 億件のイベントを格納できます。

- **fireamp_event**
- **intrusion_event**
- **intrusion_event_packet**
- **connection_log**（5.0 より古いバージョンでの名前：**rna_flow**）
- **connection_summary**（5.0 より古いバージョンでの名前：**rna_flow_summary**）

モニタ対象ネットワークでシステムが検出したホストの数によっては、ネットワーク マップ テーブルに対するクエリはコストが高くなることがあります。

クエリのヒント

以降の項では、検出エンジンまたは侵入イベントを含むクエリを作成する場合に、固有の結果が得られるようにするためのヒントを示します。

デバイス名

デバイス名は複数の Firepower Management Center 間で必ずしも一意である必要はありません。一意であるようにするには、特定のデバイス UUID をクエリに含めます。

侵入イベント

複数の管理対象デバイス上の侵入イベントに一意に一致するためには、**intrusion_event** テーブルの次のフィールドをクエリに指定できます。

- `intrusion_event.event_id`
- `intrusion_event.event_time_sec`
- `intrusion_event.sensor_uuid`

クエリのトラブルシューティング

1 つのクライアントへのアクセスを許可するように複数の Firepower Management Center を設定できますが、各システムを個別に設定する必要があります。各システムで使用可能な情報は複数の要因によって決まります。照会するデータがない場合、クエリから予期している結果が返されません。

以下に、クエリから結果が返されない原因のいくつかを示します。

- クエリの条件が細かすぎます。たとえば、クエリの時間範囲または IP アドレス範囲を調整する必要があります。
- イベント発生の原因となったネットワーク トラフィックによっては、イベントの一部のフィールドにデータが取り込まれないことがあります。たとえば、すべての接続イベントにペイロード情報が含まれているわけではありません。
- 照会するイベント タイプのログ記録を設定していませんでした。

- イベント保存を無効にしていました。
- ユーザがアクセスできないドメインを照会しようとしています。

イベントの生成方法とログ記録方法の詳細については、『*Firepower Management Center Configuration Guide*』を参照してください。

サンプル クエリ

以降の項に収録されているサンプル クエリでは、データベース アクセス機能の使用方法を示します。

注意　展開によっては、一部のサンプル クエリの実行には非常にコストがかかることがあります。詳細については、「パフォーマンス上の理由によるクエリの制限（2-14 ページ）」を参照してください。

ユーザの監査レコード

次のクエリは、特定ユーザの監査ログのレコードをすべて返します。タイムスタンプは UTC で表示されます。

```
SELECT FROM_UNIXTIME(action_time_sec), user, message
FROM audit_log
WHERE user = 'eventanalyst';
```

プライオリティおよび分類別の侵入イベント

次のクエリは、[Events By Priority and Classification] ワークフローの [Drilldown of Event]、[Priority]、[Classification] ビューを複製します。ユーザ設定でデフォルトの [Intrusion Events] ワークフローを変更していない場合、これが Firepower Management Center Web インターフェイスで [Analysis] > [Intrusion Events] を選択すると最初に表示されるページです。

```
SELECT rule_message, priority, rule_classification, count(*) as Count
FROM intrusion_event
WHERE reviewed="0" GROUP BY rule_message, priority, rule_classification
ORDER BY Count
DESCLIMIT 0, 25;
```

侵入イベントおよびその関連ポリシー

次のクエリは、指定した週の侵入イベントをリストします。イベントごとに、関連する侵害ポリシー違反とルール分類が表示されます。

```
SELECT FROM_UNIXTIME(event_time_sec) AS event_time, event_id AS intrusion_event,
intrusion_event_policy_name AS policy, rule_classification AS classification

FROM intrusion_event

WHERE event_time_sec BETWEEN UNIX_TIMESTAMP('2011-10-01 00:00:00') AND

UNIX_TIMESTAMP('2011-10-07 23:59:59')

ORDER BY policy ASC;
```

検出されたホストのリスト

次のクエリは、ネットワークで検出されたすべての MAC ホスト（IP アドレスのないホスト）のホスト ネットワーク マップでの基本情報と、各 NIC のハードウェア ベンダーを返します。

```
SELECT HEX(mac_address), mac_vendor, host_type, FROM_UNIXTIME(last_seen_sec)

FROM rna_mac_host;
```

次のクエリは、IP アドレスを MAC アドレスにマップします。

```
SELECT HEX(ipaddr), HEX(mac_address), HEX(host_id)

FROM rna_host_ip_map LEFT JOIN rna_host_mac_map on

rna_host_ip_map.host_id=rna_host_mac_map.host_id;
```

検出されたサーバのリスト

次のクエリは、2 つの関連するテーブルを結合し、ネットワークで検出されたサーバを、その属性の多くとともにリストとして返します。これは、Firepower Management Center の Web インターフェイスでサーバのテーブル ビューに表示される内容に似ています。

```
SELECT FROM_UNIXTIME(s.last_used_sec), HEX(s.host_id), s.port, s.protocol, s.hits,

i.service_name, i.vendor, i.version, i.source_type, s.confidence

FROM AS s

LEFT JOIN rna_ip_host_service_info AS i ON (s.host_id = i.host_id AND s.port = i.port AND
s.protocol =

i.protocol);
```

このクエリは、データベース アクセスで必要なため、host_id、port、および protocol のセットでテーブルを左結合する点に注意してください。rna_host_service の結合（6-37 ページ）および rna_host_service_info の結合（6-41 ページ）を参照してください。

ネットワークのサーバの脆弱性

次のクエリは、2 つの脆弱性関連テーブルを結合し、特定のホストで検出された有効なサーバ関連の脆弱性と、各脆弱性がネットワーク上で悪用可能であるかどうかのリストを示します。

```
SELECT h.rna_vuln_id, v.title, v.remote

FROM rna_host_service_vulns AS h

LEFT JOIN rna_vuln AS v ON (h.rna_vuln_id = v.rna_vuln_id)

WHERE h.ip_address = INET_ATON('10.10.10.4')

AND h.invalid = 0;
```

このクエリは、rna_host_service_vulns（6-47 ページ）と rna_vuln の結合（6-58 ページ）で必要なため、**rna_vuln_id** でテーブルを左結合する点に注意してください。

オペレーティング システムの概要

次のクエリは、[Operating System Summary] ワークフローで [Summary of OS Names] ページを複製します。ユーザ設定でデフォルトのワークフローを変更していない場合、これが Firepower Management Center Web インターフェイスで [Analysis] > [Hosts] を選択し、次に [Hosts] を選択すると最初に表示されるページです。

```
SELECT vendor, product, count(*) AS total

FROM rna_host_os

GROUP BY vendor, product

ORDER BY total DESC;
```

ホストのオペレーティング システムの脆弱性

次のクエリは、2 つの脆弱性関連テーブルを結合し、特定のホストで検出された有効なオペレーティング システム関連の脆弱性と、各脆弱性がネットワーク上で悪用可能であるかどうかのリストを示します。

```
SELECT h.rna_vuln_id, v.title, v.remote

FROM rna_host_os_vulns AS h

LEFT JOIN rna_vuln AS v ON (h.rna_vuln_id = v.rna_vuln_id)

WHERE h.host_id = UNHEX('9610B6E6F1784DA4B39BEA7A210AAD68')

AND h.invalid = 0;
```

このクエリは、データベース アクセスで必要なため、**rna_vuln_id** でテーブルを左結合する点に注意してください。rna_host_os_vulns（6-31 ページ）および rna_vuln の結合（6-58 ページ）を参照してください。

ホストの違反カウント

次のクエリは、[Host Violation Count] ワークフローで [Host Violation Count] ページを複製します。ユーザ設定でデフォルトの [Compliance White List Violations] ワークフローを変更していない場合、これが Firepower Management Center Web インターフェイスで [Analysis] > [Correlation] > [White List Violations] を選択すると最初に表示されるページです。

```
SELECT host_id, HEX(host_id), white_list_name, count(*) AS total

FROM white_list_violation

GROUP BY host_id, white_list_name

ORDER BY total DESC;
```