Enterprise Security Configuration and Compliance System Data Dictionary

[System Overview 3](#_Toc90729843)

[ESC2 Database 3](#_Toc90729844)

[Operational System ER Diagrams 3](#_Toc90729845)

[Asset Tracking Subsystem 5](#_Toc90729846)

[Implementation Guide Subsystem 6](#_Toc90729847)

[Implementation Subsystem 7](#_Toc90729848)

[Audit Subsystem 8](#_Toc90729849)

[Logs 9](#_Toc90729850)

[Data Mart Diagrams 10](#_Toc90729851)

[Implementation Fact 10](#_Toc90729852)

[Audit Fact 11](#_Toc90729853)

[Resource Cost Fact 12](#_Toc90729854)

[Log Fact 12](#_Toc90729855)

[Data Table Descriptions 13](#_Toc90729856)

[ADJUSTMENT 13](#_Toc90729857)

[ADJUSTMENT\_SET 13](#_Toc90729858)

[ASSET 13](#_Toc90729859)

[ASSET\_GROUP 14](#_Toc90729860)

[ASSET\_HISTORY 14](#_Toc90729861)

[ASSET\_TYPE 15](#_Toc90729862)

[AUDIT 15](#_Toc90729863)

[AUDIT FACT 16](#_Toc90729864)

[AUDIT\_RULE 17](#_Toc90729865)

[DEPARTMENT 17](#_Toc90729866)

[EMPLOYEE 18](#_Toc90729867)

[EVIDENCE 18](#_Toc90729868)

[EVIDENCE\_SET 19](#_Toc90729869)

[IMPLEMENTATION 19](#_Toc90729870)

[IMPLEMENTATION FACT 20](#_Toc90729871)

[IMPLEMENTATION\_GUIDE 20](#_Toc90729872)

[IMPLEMENTATION\_RULE 21](#_Toc90729873)

[LOG DAILY FACT 22](#_Toc90729874)

[LOG SEVERITY 22](#_Toc90729875)

[LOG SOURCE 23](#_Toc90729876)

[RESOURCE COST FACT 23](#_Toc90729877)

[RULE 24](#_Toc90729878)

[SYSTEM EVENT 25](#_Toc90729879)

[SYSTEM EVENT HISTORY 25](#_Toc90729880)

[SYSTEM EVENT PROPERTY 26](#_Toc90729881)

[SYSTEM EVENT PROPERTY HISTORY 26](#_Toc90729882)

[USER 27](#_Toc90729883)

[USER ROLE 27](#_Toc90729884)

[VERSION 28](#_Toc90729885)

# System Overview

The Enterprise Security Configuration and Compliance (ESC2) database is designed to allow companies that work with the Federal government to manage their overall security stance when working with Controlled Unclassified Information (CUI).

This database will assist companies with compliance with the National Institute of Standards and Technology (NIST) document 800-171 Section 3.4.2. This section requires that companies "Establish and enforce security configuration settings for information technology products employed in organizational systems."

This system utilizes the existing Department of Defense (DOD) Security Technical Implementation Guides (STIGs) as the foundation for securely configuring technology.

The database will support compliance by tracking the implementation of the STIG specifications on their existing and new technologies. It also allows for the tracking of audits to verify that the hardware and software is correctly implemented.

# ESC2 Database

The ESC2 database is designed as a unified operational and analytic database. The ESC2 Architectural Overview goes into the reasoning behind this decision.

**Database Schemas**

Though unified, the database will be broken down into schemas to organize the tables by purpose. This will assist if it becomes necessary to break it up for performance or storage reasons.

The three schemas identified are:

* Operational – All tables intended for use by the operational system
* Warehouse – All fact and dimension tables
* Logging – Logging and Logging history tables

**Diagram Color Coding**

**In the diagrams tables are colored to indicate how they are used.**

* Blue – Operational Tables. These are sometimes used as dimension tables. (Warehouse or Logging)
* Green – Fact Tables (Warehouse Schema)
* Yellow – Pure Dimension tables. (Warehouse Schema)

# Operational System ER Diagrams

Due to the number of tables, the diagrams are broken down into subject areas. Some tables exist in multiple diagrams as they support multiple processes. The tables that exist in many diagrams may not have all of their related tables shown. These relationships will only show in the primary diagram.

**Standardized User Tracking**

In the operational tables user activity tracking is handled by four columns

* created\_by\_id – User Id that created the record
* created\_on – date/time (UTC) that the record was created
* last\_modified\_by\_id – User Id that last modified the record
* last\_modified\_on – date/time (UTC) that the record was last modified.

When a record is created the created\_by\_id and last\_modified\_by\_id columns will match. This is also true for the created\_on and last\_modified\_on columns.

There are a few tables that only have the two created properties. These tables are not intended to be updated.

Due to the number of tables that have these User Id columns, explicit RI is not created for these columns. The application will have to ensure that the properties are correctly set.

Fact tables and other system generated records will not track these audit columns.

## Asset Tracking Subsystem

Diagram

Description automatically generated

The Asset Subsystem represents the core of the overall ESC2 system.

Assets can be grouped into a three-level hierarchy.

The highest level is the Asset Group. Asset Groups will include items like Routers, Switches, Servers, Laptops, Cell Phones, Tablets, etc.

The second level of the hierarchy is Asset Type. Asset Types exist under an Asset Group and allow assets to be grouped by a specific type of item. Asset Types will include items like Fortigate 60E Firewall, Dell PowerEdge 2080, Apple iPhone 10, or Apple iPad 11”.

The final level is the assets themselves. These will be specific instances of hardware or software that is to be tracked and secured. This would include items like a specific individual iPhone or the Dell PowerEdge 2080 File Server.

The Employee and Department tables are used for grouping and identifying who is responsible for the asset.

The User and User Role tables are for system access and allowed user functionality.

## Implementation Guide Subsystem

Diagram

Description automatically generated

Implementation Guides are sets of instructions that allow technology to be securely implemented. The initial source for the Implementation Guides are STIG files supplied by the military (<https://public.cyber.mil/stigs/>).

In the future guides from other sources will be integrated.

Each Implementation Guide can have revisions published at any time as new information becomes available in covered technologies. When this happens a new version record will be created with a new set of rules. Even if most rules remain unchanged a new set will be created. The Implementations (described in the following section) will tie back to the set of rules that was in place at the time the implementation was completed. Similarly, an audit of the Implementation will tie back to the same sets of rules from the Implementation Guide’s version associated with the Implementation.

The system will not allow an audit to be completed with a newer Implementation Guide version. If a new Implementation Guide is published the implementation must be repeated to ensure that all vulnerabilities described are mitigated.

The adjustment\_set and adjustment tables allow these guides to be tweaked by a company to meet their specific requirements.

## Implementation Subsystem

Diagram

Description automatically generated

The Implementation Subsystem allow a company to track users activities as they implement specific Implementation Guides on specific hardware or software.

The implementation table tracks the high level information on an implementation. The implementation\_rule table tracks the resolution of a specific vulnerability (rule).

The evidence\_set and evidence tables allow users to supply evidence that a vulnerability was address using images or other documents.

## Audit Subsystem

Diagram

Description automatically generated

The Audit Subystem allow companies to track a process of independent verification that a system was correctly secured.

The primary tables are the audit and audit rule tables. The audit table tracks the overall audit process, while the audit\_rule table tracks the verification of a specific vulnerability (rule).

Evidence of successful verification or failure to verify that a vulnerability has been address is tracked in the evidence\_set and evidence (not shown) tables.

## Logs

Diagram

Description automatically generated with medium confidence

The Logging Subsystem is used to track near real-time log records. The initial focus is on collecting SysLog records from network equipment. Future work will be done to capture web server logs and other application sources.

The system\_event and system\_event\_property tables are used to store Syslog messages that arrive from various network devices. The history versions of the tables will be used to store long term data. The primary version of the table will hold the current day’s data. A nightly process will migrate the data from one table to the other, then clean out the old data.

Future tables will be added to capture logs from other sources that do not use syslog. These will include web logs from IIS, Apache or Nginx.

# Data Mart Diagrams

These diagrams are broken down by fact.

## Implementation Fact

Diagram

Description automatically generated

The implementation\_fact table is designed to be used to estimate the cost of implementation activities. It is designed to be used with the resource\_cost\_fact described below.

The hours\_to\_complete field will be estimated from the employees’ work hours, and number of implementations that were completed during the period. This will produce a weak estimate.

It is likely that additional time tracking features be added to the system to assist with this calculation. New table structures will be needed to track this time information and associate it back to a specific employee. Multiple different approaches may be necessary. The time will at best be an estimate. There is no direct way to precisely estimate the time.

Even with an imprecise estimate it should be possible for management to view for the following:

* Approximate cost of securing different types of assets
* Compare the time required to implement technology by employee
* Estimate the total cost of security implementations

## Audit Fact

Diagram

Description automatically generated

The audit\_fact table is designed to assist company management with estimating the costs associated with verifying that technologies have been correctly implemented. It is designed to be used in conjunction with the resource\_cost\_fact table.

Auditing will have the same lack of precision in the calculation of time to complete an audit that the implementation suffers from. Similar application efforts may be necessary to get the best estimate.

It should be able to answer questions like:

* How long do different types of technology require to audit
* Which implementors are more complete with their implementations of securing technology
* What is the approximate cost of auditing different types of technology
* What is the approximate overall cost of securing technology (when used with implementation\_fact)

## Resource Cost Fact

Graphical user interface, text, application, chat or text message

Description automatically generated

The resource\_cost\_fact table allows tracking average cost per hour for different user roles. These roles are focused on the system admins and security users of the system.

The source of the loaded\_rate\_per\_hour will either come into this system from an HR system, be sourced externally using a Dice Salary Survey or from other government data on employee costs. This rate should include all employee costs per hour (loaded) not just the salary that they receive.

## Log Fact

Diagram

Description automatically generated

The log\_daily\_fact table contains aggregate counts of log activity grouped by severity and source and day. It may be useful in the future to add a log\_hourly\_fact table to aggregate at the hourly level.

# Data Table Descriptions

Tables ordered alphabetically

## ADJUSTMENT

Schema: operational

Allows a specific property of a Rule to be overridden by the company.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| adjustment\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| type | VARCHAR(20) | NOT NULL | Property that is to be over written |
| details | VARCHAR(max) | NULLABLE | Value that is over written. |
| adjustment\_set\_id | UUID | NOT NULL,  FK to  adjustment\_set.  adjustment\_set\_id | Foreign Key |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## ADJUSTMENT\_SET

Schema: operational

Allows one or more rule attributes to be overridden

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| asset\_group\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| rule\_id | UUID | NOT NULL,  FK to  rule.rule\_id | Foreign Key |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |

## ASSET

Schema: operational

A specific piece of hardware or installation of software that is operating in the company. This would cover items like Accounting Database Server, or Engineering GitLab Build Server.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| asset\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| asset\_type\_id | UUID | NOT NULL | Foreign key to ASSET\_TYPE |
| parent\_asset\_id | UUID | NULLABLE,  FK to  ASSET.ASSET\_ID | Self-relationship Foreign key |
| owning\_department\_id | UUID | NOT NULL  FK to DEPARTMENT | Foreign key to DEPARTMENT table |
| contact\_employee\_id | Long | NOT NULL  FK to EMPLOYEE  .EMPLOYEE\_ID | Foreign key to EMPLOYEE table |
| name | VARCHAR(100) | NOT NULL | Name of the asset in the company |
| description | VARCHAR(max) | NULLABLE | Optional. Description of the asset. |
| status | CHAR(1) | NOT NULL, P, A, R | Pending, Active, Retired |
| host\_name | varchar(255) | NULLABLE | The name network name of the device |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## ASSET\_GROUP

Schema: operational

A generic grouping of technology. Items here include Server, Desktop, Laptop, Router, Database, Web Service, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| asset\_group\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| name | VARCHAR(255) | NOT NULL | Name of the asset group |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## ASSET\_HISTORY

Schema: operational

Tracks the history of the asset over its lifespan (Purchased, In Service, Retired).

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| asset\_history\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| action\_date | DateTime | NOT NULL | Date the action occurred |
| action | varchar(255) | NOT NULL | Action that occurred |
| user\_id | UUID | FK to  user.user\_id | Foreign Key |
| asset\_id | UUID | FK to  asset.asset\_id | Foreign Key |

## ASSET\_TYPE

Schema: operational

A specific type of equipment or software that a company may purchase to implement. This would include things like a Dell Latitude 8050 laptop, or a Dell R710 PowerEdge server. It could also include software like SQL Server 2019, or MySQL 5.6.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| asset\_type\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| name | VARCHAR(255) | NOT NULL | Name of the asset type |
| description | VARCHAR(Max) | NOT NULL | Description of the asset type |
| manufacturer | VARCHAR(100) | NOT NULL | Manufacturer of the asset type |
| model | VARCHAR(100) | NOT NULL | Model of the asset type |
| version | VARCHAR(50) | NOT NULL | Version number of the asset type to allow for versioning of the files without disrupting prior implementation and audits |
| implementation\_guide\_id | UUID | NOT NULL,  Foreign Key to  implementation\_guide.  implementation\_guide\_id | Foreign Key |
| asset\_group\_id | UUID | NOT NULL  Foreign Key to ASSET\_GROUP.AG\_ID | Foreign key to ASSET\_GROUP table |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## AUDIT

Schema: operational

The AUDIT table is used to associate the Asset with a STIG Requirement and the auditing of the rules that are required by the STIG.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| audit\_id | UUID | NOT NULL, UNIQUE | Surrogate Primary Key |
| implementation\_id | UUID | NOT NULL,  FK to  implementation.  implementation\_id | Foreign Key |
| assigned\_user\_id | UUID | NULLABLE,  FK to  USER.USER\_ID | Foreign Key |
| start\_date | DATE | NULLABLE | Date that the audit process began |
| end\_date | DATE | NULLABLE | Date that the audit process was completed. |
| status | CHAR(1) | NOT NULL,  O, I, P, F  Default - O | Status of the audit.  Open, In Progress, Pass, Fail |
| summary | VARCHAR(max) | NULLABLE | Details of the audit outcome. |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## AUDIT FACT

Schema: warehouse

Facts related to completed audit records

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| audit\_fact\_id | int | NOT NULL, UNIQUE, | Primary Key, Identity |
| audit\_id | UUID | NOT NULL,  FK to  audit.audit\_id | Foreign Key |
| asset\_id | UUID | NOT NULL,  FK to  asset.asset\_id | Foreign Key |
| implementation\_guide\_id | UUID | NOT NULL,  FK to  implementation\_guide.  implementation\_guide\_id | Foreign Key |
| user\_id | UUID | NOT NULL,  FK to user.  user\_id | Foreign Key |
| started\_on\_period\_id | int | NOT NULL,  FK to  day\_period.  day\_period\_id | Foreign Key |
| completed\_on\_period\_id | int | NOT NULL,  FK to day\_period.  day\_period\_id | Foreign Key |
| hours\_to\_complete | double (decimal, float) | NOT NULL | Number of hours that were required to complete the audit. |

## AUDIT\_RULE

Schema: operational

The AUDIT\_RULE table is used to track an audit to each STIG Rule. It allows the auditors to make comments on the implementation of the Rule and verify it was correctly implemented.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| audit\_rule\_id | UUID | NOT NULL, UNIQUE | Surrogate Primary Key |
| rule\_id | UUID | NOT NULL,  FK to  RULE.RULE\_ID | Foreign Key |
| audit\_id | UUID | NOT NULL, FK to  AUDIT.AUDIT\_ID | Foreign Key |
| comment | VARCHAR(max) | NULLABLE | Any notes that the auditor wishes to enter. |
| status | CHAR(1) | NOT NULL,  O, P, F,  Default - O | Status of the inspection of the associated rule. |
| evidence\_set\_id | UUID | NULLABLE,  FK to  evidence\_set.  evidence\_set\_id | Foreign Key for evidence of success or failure |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## DEPARTMENT

Schema: operational

The DEPARTMENT table is used to track the department information.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| department\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| name | VARCHAR(100) | NOT NULL | Name of the department |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## EMPLOYEE

Schema: operational

The EMPLOYEE table is used to track employee information.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| employee\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| given\_name | VARCHAR(50) | NOT NULL | Employee’s legal first name |
| last\_name | VARCHAR(50) | NOT NULL | Employee’s legal last name |
| job\_title | VARCHAR(100) | NOT NULL | Employee’s job title |
| email | VARCHAR(255) | NOT NULL | Employee’s work email address |
| start\_date | Date | NULLABLE | Date employee started |
| end\_date | Date | NULLABLE | Date employee terminated or exited company |
| department\_id | UUID | NOT NULL  Foreign key to DEPARTMENT table | Foreign key to DEPARTMENT table |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## EVIDENCE

Schema: operational

A generic grouping of technology. Items here include Server, Desktop, Laptop, Router, Database, Web Service, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| evidence\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| data\_location | VARCHAR(255) | NOT NULL | Location of the asset in external storage |
| mime\_type | varchar(50) | NOT NULL | Type of file that was set |
| evidence\_set\_id | UUID | NOT NULL,  FK to  evidence\_set.  evidence\_set\_id |  |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |

## EVIDENCE\_SET

Schema: operational

A generic grouping of technology. Items here include Server, Desktop, Laptop, Router, Database, Web Service, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| evidence\_set\_id | UUID | NOT NULL, UNIQUE | Primary Key |
| type | CHAR(1) | NOT NULL,  (A, I) | Implementation or Audit |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |

## IMPLEMENTATION

Schema: operational

The IMPLEMENTATION table associates a REQUREMENT (STIG) with a specific ASSET and the IMPLEMENTATION RULES that detail how the STIG was applied to the ASSET.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| implementation\_id | UUID | NOT NULL, UNIQUE | Surrogate Primary Key |
| assigned\_user\_id | UUID | NULLABLE,  FK to  USER.USER\_ID | Foreign Key |
| asset\_id | UUID | NOT NULL,  FK to  ASSET.ASSET\_ID | Foreign Key |
| implementation\_guide\_id | UUID | NOT NULL,  FK to  implementation\_guide.  implementation\_guide\_id | Foreign Key |
| start\_date | DATE | NULLABLE | Date that the implementation process began |
| end\_date | DATE | NULLABLE | Date that the implementation process was completed. |
| created\_on | DATE | NOT NULL | Date the implementation was created. |
| status | CHAR(1) | NOT NULL,  O, I, C  Default - O | Status of the implementation.  Open, In Progress, Complete |
| summary | VARCHAR(max) | NULLABLE | High level details of the implementation of the STIG |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## IMPLEMENTATION FACT

Schema: warehouse

Fact table summarizing the time required to complete the implementation. Only completed implementations will exist in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| implementation\_fact\_id | int | NOT NULL, UNIQUE | Primary Key, Identity |
| implementation\_id | UUID | NOT NULL, UNIQUE  FK to  implementation.  implementation\_id | Foreign Key. |
| asset\_id | UUID | NOT NULL,  FK to  asset.asset\_id | Foreign Key |
| implementation\_guide\_id | UUID | NOT NULL,  FK to  implementation\_guide.  implementation\_guide\_id | Foreign Key |
| employee\_id | UUID | NOT NULL,  FK to  employee.  employee\_id | Foreign Key |
| started\_on\_period\_id | int | NOT NULL,  FK to  day\_period.  day\_period\_id | Foreign Key |
| completed\_on\_period\_id | int | NOT NULL,  FK to  day\_period.  day\_period\_id | Foreign Key |
| hours\_to\_complete | double (float, numeric) | NOT NULL | Number of hours needed to complete the implementation |

## IMPLEMENTATION\_GUIDE

Schema: operational

This table is used to keep track of the STIG files that are loaded into the database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| implementation\_guide\_id | UUID | NOT NULL, UNIQUE | Surrogate Primary Key |
| number | VARCHAR(50) | NOT NULL, UNIQUE | This is the Benchmark.id from the STIG file |
| type | VARCHAR(50) | NOT NULL | STIG, Customer, etc. |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

NOTE:

When a IMPLEMENTATION\_GUIDE row is loaded a search should be made for the NUMBER. A new version record will be created if the IMPLEMENTATION\_GUIDE row exists. Rules will be attached this new version record.

## IMPLEMENTATION\_RULE

Schema: operational

The IMPLEMENTATION\_RULE table is an associative table that maps the RULE table rows to the IMPLEMENTATION table. It also tracks the status and result of the implementation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| implementation\_rule\_id | UUID | NOT NULL, UNIQUE | Surrogate Primary Key |
| implementation\_id | UUID | NOT NULL,  FK to  IMPLEMENTATION  .IMP\_ID | Foreign Key |
| rule\_id | UUID | NOT NULL,  FK to  RULE.RULE\_ID | Foreign Key |
| status | VARCHAR(5) | NF, O, NA, NR NOT NULL,  DEFAULT - NF | Not a Finding, Open, Not Applicable, Not Reviewed |
| finding\_details | VARCHAR(max) | NULLABLE |  |
| comments | VARCHAR(max) | NULLABLE | Notes on how the implementation was completed. |
| evidence\_set\_id | UUID | NULLABLE,  FK to  evidence\_set.  evidence\_set\_id | Used for supplying evidence of implementation. |
| severity\_override | CHAR(1) | H, M, L  NULLABLE | High, Medium, Low.  Allows the implementor to override the severity if needed. |
| severity\_justification | VARCHAR(max) | NULLABLE | This is required if the Security override has been set. Implementor must justify why the override was changed |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

NOTE:

When an implementation is created the system should create one record in the IMPLEMENTATION\_RULE table for each row in the RULE table.

## LOG DAILY FACT

Schema: warehouse

Fact table that tracks the number of events for the period of a day. This will allow for tracking of load and compare it to similar days.

Need to determine if multiple rollups are needed. The addition of hourly could be useful for some charting. Smaller aggregates will need to refer to the operation tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| log\_daily\_fact\_id | bigint | NOT NULL, UNIQUE | Primary Key, Identity |
| asset\_id | UUID | NOT NULL,  FK to  asset.asset\_id | Foreign Key |
| day\_period\_id | int | NOT NULL,  FK to  day\_period.  day\_period\_id | Foreign Key |
| log\_severity\_id | tinyint | NOT NULL,  FK to  log\_severity.  log\_severity\_id | Foreign Key |
| log\_source\_id | smallint | NOT NULL,  FK to  log\_source.  log\_source\_id | Foreign Key |
| count | int | NOT NULL | Number of events in the period. |

## LOG SEVERITY

Schema: warehouse

The level of severity.

Graphical user interface, table

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| log\_severity\_id | tinyint | NOT NULL, UNIQUE | Primary Key |
| name | varchar(50) | NOT NULL, UNIQUE | Name of the severity |

## LOG SOURCE

Schema: warehouse

This table tracks the values of the type of service. For SysLog values range from 0 to 23. Future values will be above 23.

Table

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| log\_source\_id | smallint | NOT NULL, UNIQUE | Primary Key |
| name | varchar(50) | NOT NULL | Name of the type of service that generated the log entry. Not name is not unique per Rsyslog information. Further investigation needed. |

## RESOURCE COST FACT

Schema: warehouse

Fact table to track the cost of resources over time

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| resource\_cost\_fact\_id | int | NOT NULL, UNIQUE | Primary Key, Identity |
| user\_role\_id | UUID | NOT NULL,  FK to  role.role\_id | Foreign Key |
| start\_period\_id | int | NOT NULL,  FK to  day\_period.  day\_period\_id | Foreign Key |
| end\_period\_id | int | NOT NULL,  FK to  period.period\_id | Foreign Key. Initially set to max value in period table. Revised when new data is loaded. |
| loaded\_rate\_per\_hour | double | NOT NULL | The fully loaded average cost of an employee in the specified role. |

## RULE

Schema: operational

Rules are defined in the STIG files. They instruct how to securely config software and hardware to minimize the risk of attack from malicious actors.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| rule\_id | UUID | NOT NULL | Surrogate Primary Key |
| version\_id | UUID | NOT NULL,  FK to  VERSION.VERSION\_ID | Foreign Key |
| number | VARCHAR(30) | NOT NULL | This is the id field from the STIG file. |
| severity | CHAR(1) | H, M, L  NOT NULL | High, Medium, Low. |
| version | VARCHAR(50) | NOT NULL | Version number for the rule. This value is found in the STIG file. |
| title | VARCHAR(max) | NOT NULL | Title of the control |
| discussion | VARCHAR(max) | NOT NULL | Discussion on why this control is important. |
| fix | VARCHAR(max) | NOT NULL | Instructions on how to fix this control |
| check | VARCHAR(max) | NOT NULL | Instructions on how to check if this control is implemented |
| cci | VARCHAR(20) | NOT NULL | CCI – Common Control Identifier. Future Use – Investigate creating tables to import CCI data |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

NOTES:

VERSION\_ID and RULE\_NUMBER make up an alternate key and should be implemented as a unique index.

## SYSTEM EVENT

Schema: logging

This ODS table is adapted from the Rsyslog specification. It contains individual log entries for drill through and aggregation purposes. Note many of the columns in this table are not fully understood at this time. Gathering of actual logs is needed to better define the columns.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| system\_event\_id | bigint | NOT NULL, UNIQUE | Primary Key, Identity |
| customer\_id | bigint | NULLABLE |  |
| received\_at | datetime |  | Time the event was received at the server |
| device\_reported\_time | datetime |  | Time that the device reported receiving the log entry |
| facility | smallint |  | Type of service that the log was generated by |
| priority | smallint |  |  |
| from\_host | varchar(60) |  |  |
| message | varchar(max) |  |  |
| nt\_severity | int |  |  |
| importance | int |  |  |
| event\_source | varchar(60) |  |  |
| event\_user | varchar(60) |  |  |
| event\_category | int |  |  |
| event\_id | int |  |  |
| event\_binary\_data | varchar(max) |  |  |
| maximum\_available | int |  |  |
| current\_usage | int |  |  |
| minimum\_usage | int |  |  |
| info\_unit\_id | int |  |  |
| sys\_log\_tag | varchar(60) |  |  |
| event\_log\_type | varchar(60) |  |  |
| generic\_file\_name | varchar(60) |  |  |
| system\_id | int |  |  |

## SYSTEM EVENT HISTORY

Schema: logging

This ODS table is adapted from the Rsyslog specification. It contains individual log entries for drill through and aggregation purposes. Note many of the columns in this table are not fully understood at this time. Gathering of actual logs is needed to better define the columns.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| system\_event\_history\_id | bigint | NOT NULL, UNIQUE | Primary Key |
| customer\_id | bigint | NULLABLE |  |
| received\_at | datetime |  | Time the event was received at the server |
| device\_reported\_time | datetime |  | Time that the device reported receiving the log entry |
| facility | smallint |  | Type of service that the log was generated by |
| priority | smallint |  |  |
| from\_host | varchar(60) |  |  |
| message | varchar(max) |  |  |
| nt\_severity | int |  |  |
| importance | int |  |  |
| event\_source | varchar(60) |  |  |
| event\_user | varchar(60) |  |  |
| event\_category | int |  |  |
| event\_id | int |  |  |
| event\_binary\_data | varchar(max) |  |  |
| maximum\_available | int |  |  |
| current\_usage | int |  |  |
| minimum\_usage | int |  |  |
| info\_unit\_id | int |  |  |
| sys\_log\_tag | varchar(60) |  |  |
| event\_log\_type | varchar(60) |  |  |
| generic\_file\_name | varchar(60) |  |  |
| system\_id | int |  |  |
| archived\_date | datetime |  | Date that the log record was moved into history |

## SYSTEM EVENT PROPERTY

Schema: logging

Supports additional log details beyond the basic set supplied in system\_event. Source Rsyslog.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| system\_event\_property\_id | bigint | NOT NULL, UNIQUE | Primary Key, Identity |
| system\_event\_id | bigint | NOT NULL,  FK to  system\_event.  system\_event\_id | Foreign Key |
| parameter\_name | varchar(255) | NOT NULL | Name of the parameter |
| parameter\_value | varchar(max) |  | Value supplied |

## SYSTEM EVENT PROPERTY HISTORY

Schema: logging

Supports additional log details beyond the basic set supplied in system\_event. Source Rsyslog.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| system\_event\_property\_history\_id | bigint | NOT NULL, UNIQUE | Primary Key |
| system\_event\_id | bigint | NOT NULL,  FK to  system\_event.  system\_event\_id | Foreign Key |
| parameter\_name | varchar(255) | NOT NULL | Name of the parameter |
| parameter\_value | varchar(max) |  | Value supplied |

## USER

Schema: operational

Users extends employee records. These are the employees that have access to the ESC2 system.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| user\_id | UUID | NOT NULL, UNIQUE | Surrogate Primary Key |
| employee\_id | UUID | NOT NULL,  Foreign Key to  EMPLOYEE.EMPLOYEE\_ID |  |
| user\_role\_id | UUID | NOT NULL,  FK to  user\_role.user\_role\_id | Manager, Auditor, Implementor, Read-only |
| is\_enabled | Boolean | NOT NULL,  DEFAULT - True | Indicates if the user can access the system |
| password\_hash | VARCHAR(50) | NULL | Hashed password |
| password\_salt | VARCHAR(50) | NULL | Used to hash password |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## USER ROLE

Schema: operational

Level of rights that a user operates with.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| user\_role\_id | UUID | NOT NULL, UNIQUE | Surrogate Primary Key |
| name | varchar(50) | NOT NULL |  |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

## VERSION

Schema: operational

The VERSION table is used to allow reloading a STIG file without invalidating prior versions of the file.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Notes** |
| version\_id | UUID | NOT NULL, UNIQUE | Surrogate Primary Key |
| implementation\_guide\_id | UUID | NOT NULL,  Foreign Key to  STIG.STIG\_ID | Foreign key to STIG table |
| number | VARCHAR(5) | NOT NULL | Version number from the STIG file. Used to allow reloading STIG without invalidating old versions. |
| status | VARCHAR(5) | NOT NULL  A, DP, DR, IC, IN | Accepted, Deprecated, Draft, Incomplete, Interim |
| status\_date | DATETIME | NOT NULL | Date that the status of the STIG file was set |
| title | VARCHAR(max) | NOT NULL | Title from the STIG File |
| description | VARCHAR(max) | NOT NULL | Description of the intent of the STIG file |
| publisher | VARCHAR(100) | NOT NULL | Publisher of the STIG file |
| source | VARCHAR(100) | NOT NULL | Source of the STIG file |
| filename | VARCHAR(255) | NOT NULL | Filename of the STIG xml file |
| created\_by\_id | UUID | NOT NULL | User that created the record |
| created\_on | datetime | NOT NULL | Date/Time the record was created |
| last\_modified\_by\_id | UUID | NOT NULL | User that last modified the record |
| last\_modified\_on | datetime | NOT NULL | Date/Time the record was last modified |

NOTE:

When a new STIG file is loaded a check should be made for the VERSION. If the version exists a warning should be given before updating all Groups and Rules.