Assignment 1

Design and Demonstration of Configuration Management Database (CMDB) for IT Security Management

Tobias Zeier, the 11th of November 2024
University of Essex Online

1. Introduction

This essay will cover the design of a CMDB, specifically tailored to support IT security management functions. Furthermore, it will demonstrate a prototype of the designed CMDB and explain its functionality.

A Configuration Management Database (CMDB) serves as a critical data source not only for managing configurations but also for analysing the present situation and predicting future states. Configuration items may represent either physical or logical resources that describe a company's equipment. The primary objective of the CMDB is to store information about configuration elements and their interrelationships, enabling the provision of accurate, real-time information on configuration elements at any given time or state (Kanaev et al., 2024).

2. Analysis of IT Security Management Requirements

Asset Management: systematic process of identifying, tracking, maintaining, and managing the organisation's IT assets throughout their lifecycle. Key requirements: comprehensive inventory, data accuracy, configuration details and asset ownership (Keller, 2017).

Vulnerability tracking: systematic process of detecting, assessing, monitoring, and managing vulnerabilities. Key requirements: vulnerability identification, classification, prioritisation and risk management (Kerner, 2024).

Compliance monitoring: ongoing process of evaluating and verifying that an organisation's policies, procedures, systems, and actions align with regulatory requirements, internal standards, and industry best practices. Key requirements: definition of standards, regulatory requirements and roles and responsibilities (Rinderle-Ma et al., 2023).

Incident response: structured process of identifying, investigating, and resolving security incidents to minimise damage, restore normal operations, and prevent future incidents. Key requirements: incident response plan, response team (CSIRT) and detection and analysis (Angafor et al., 2020).

3. Conceptual Data Model for the CMDB

The following section lists key configuration items (CI), their attributes as well as their relation to each other regarding IT security management.

Assets (Hardware/Software): Asset ID, Asset Name, Asset Type, Ownership, Location, Criticality, Lifecycle Status, Compliance Status, Configuration, Related Assets

Vulnerabilities: Vulnerability ID, Severity, Detection Date, Detection Source, Impact, Patch Status, Affected CI's, historical log (open, mitigated, resolved)

Compliance Requirements: Compliance ID, Regulation (e.g., ISO 27001, GDPR, HIPAA), Affected CI's, Compliance Status (complaint, non-complaint), Last Audit Date

Incidents: Incident ID, Incident Title, Severity, Impact, Description, Detection Date, Affected CI's, Response Actions, Incident Owner, Timeline

Users: User ID, Name, Role, Access Level, Affected CI's

Services: Service ID, Name, Dependencies, Criticality to Business, Associated Incidents or Vulnerabilities

Asset ↔ Vulnerability
Asset ↔ Compliance Requirements
Asset ↔ Incidents
Asset ↔ Users
Asset ↔ Services
Vulnerability ↔ Incidents
Vulnerability ↔ Compliance
Compliance Requirements ↔ Incidents

4. Detailed CMDB Design Document

The architecture for this CMDB is defined as a central database with a role-based access control (RBAC) and interfaces to connect it to other key IT systems within the organisation.

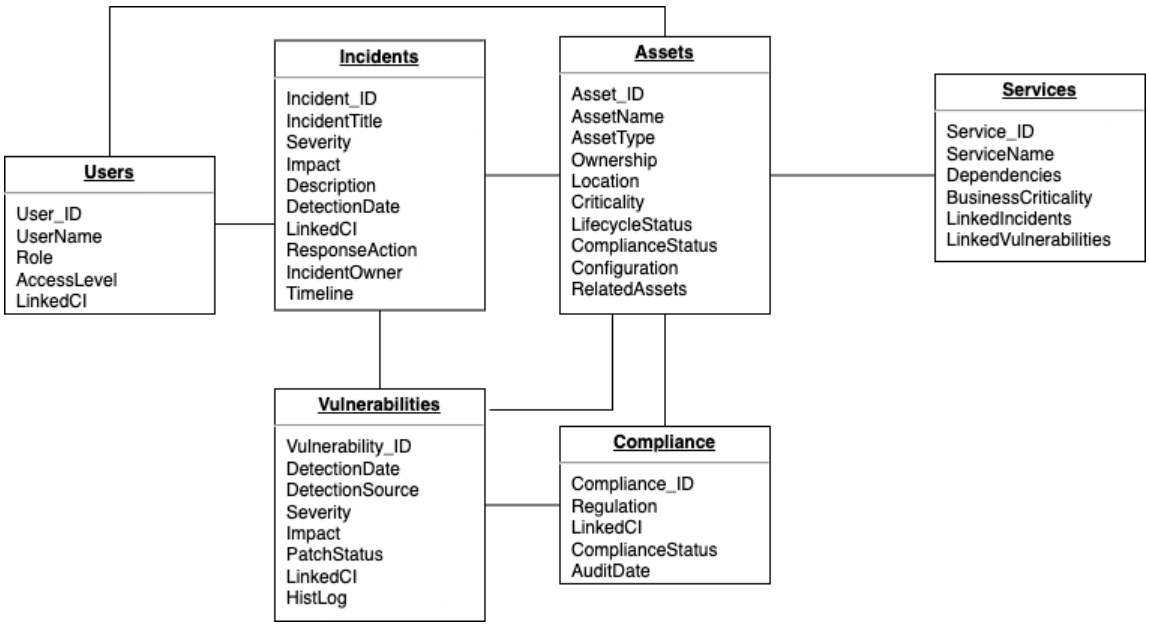

Figure 1: IT Security CMDB Data Schema (Zeier, 2024)

4.1 Data Validation Rules

As in every database, the IDs are unique and function as primary key, hence these can be used as unique identifiers. LinkedCI will be used as foreign key to link it to Asset_ID. The user needs to have the opportunity to select the affected asset based on its name since it would not be user-friendly to ask for the asset ID. Fields like AssetName, UserName and ServiceName are also unique to ensure that there will be no misunderstanding. Detection Dates are in date format which can be selected by the user. Whenever possible fields should be setup as drop downs where users can select a predefined value, as this prevents misspelling, typos, etc. Preventing data duplication in a database is essential for maintaining data integrity, ensuring accurate reporting, and optimising storage. Hence normalisation of data should be applied and triggers on the application level need to be setup to detect data duplication (Stephens, 2023).

4.2 User Interface Specification

The following section describes user interfaces for important operations and daily business tasks for users.

Dashboard: The initial screen is a dashboard where the user can add and remove widgets. These show open incidents, compliance issues, upcoming deadlines and general alerts. On this dashboard users only see records in relation to themselves (i.e. where they are incident owners). The header of this screen shows a search bar where users can search assets, incidents, etc.

Asset Management: By navigation to this menu, the user can view a list with all assets (CI's). Existing ones can be altered, and new ones can be added. If an asset does not exist anymore, it can be disabled but not deleted for the purpose of traceability.

Incident Management: This screen shows a list of open incidents which are assigned to the user viewing the screen. The list can be filtered and sorted. There is an option to view all incidents (not only the ones where the current user is the owner). New incidents can be raised from here.

Compliance Monitoring: Here an overall compliance score can be viewed. Alerts and notifications such as upcoming deadlines and policy changes are shown. Assets with potential compliance issues are listed.

Reporting: This screen allows to create reporting for all key aspects such as assets, incidents and compliance related events. These reports can be exported to a CSV or PDF file which can be used for completing documents for C-Level presentations, internal or external audits and more.

5. CMDB Demonstration

This section will demonstrate four pre-defined scenarios and describe all possible steps in each scenario.

Scenario 1: Adding New Security Assets to the CMDB

- Select asset type and enter basic information
- Define security and compliance attributes
- Establish relationship with other assets
- Enhancement: discovery service which adds assets automatically

Scenario 2: Updating Configuration Data

- Select asset and enter edit mode
- Update technical or compliance attributes
- Save changes and notify stakeholders
- Enhancement: discovery service which updates assets automatically

Scenario 3: Conducting Vulnerability Assessments

- Create new vulnerability with all relevant information and link it to affected assets
- Update vulnerability status
- Enhancement: fetch data from other relevant systems (see 5.1)

Scenario 4: Generating Compliance Reports

- Select compliance standard and define report scope
- Generate compliance report
- Export and share the report
- Enhancement: Automatically regularly generated reports which are sent to key stakeholders

5.1 Challenges and Enhancements

While designing the CMDB several challenges arose. It was difficult to find literature for CMDB’s in relation to IT Security Management, hence there were several factors which were unknown. As an enhancement, the application should provide API’s

which can fetch relevant data from systems such as Black Duck, Vulnerability Scans, CVSS databases and more. The automated loading of data from these sources should then automatically generate incidents, resulting in considerably less effort when maintaining this database.

Word count: 1094

References:

Kanaev, A. K., Login, E. V., & Pudovkina, K. A. (2024) 'Conceptual Foundations for Forming a Configuration Management Subsystem of a Telecommunications Network', *2024 International Russian Smart Industry Conference (SmartIndustryCon)*. Sochi, Russian Federation, 25-29 March. IEEE. 938–942. DOI: https://doi.org/10.1109/smartindustrycon61328.2024.10516161

Keller, A. (2017) Challenges and Directions in Service Management Automation. *Journal of Network and Systems Management* 25(4): 884–901. DOI: https://doi.org/10.1007/s10922-017-9437-9

Kerner, S. M. (2024) What is vulnerability management? Available from: https://www.techtarget.com/searchsecurity/definition/vulnerability-management [Accessed 9 November 2024]

Rinderle-Ma, S., Winter, K., & Benzin, J.-V. (2023) Predictive compliance monitoring in process-aware information systems: State of the art, functionalities, research directions. *Information Systems* 115(102210). DOI: https://doi.org/10.1016/j.is.2023.102210

Angafor, G. N., Yevseyeva, I., & He, Y. (2020) Game-based learning: A review of tabletop exercises for cybersecurity incident response training. *Security and Privacy* 3(6): e126. DOI: https://doi.org/10.1002/spy2.126

Zeier, T. (2024) Assignment 1: Design and Demonstration of Configuration Management Database (CMDB) for IT Security Management. *ITSM November 2024*. Essay submitted to the University of Essex Online.

Stephens, R. (2023) *Beginning database design solutions: understanding and implementing database design concepts for the cloud and beyond* (2nd ed). Hoboken, New Jersey: Wiley.