



Critical Infrastructure Protection in the National Capital Region

**Risk-Based Foundations for Resilience and
Sustainability**

**Final Report, Volume 10:
A Database and Architecture for Comparing
Vulnerability Assessment Elements**

September 2005

University Consortium for Infrastructure Protection

Managed by the
Critical Infrastructure Protection Program
School of Law
George Mason University

This Page Intentionally Blank

Critical Infrastructure Protection in the National Capital Region

Risk-Based Foundations for Resilience and Sustainability

Final Report, Volume 10: A Database and Architecture for Comparing Vulnerability Assessment Elements

Submitted in fulfillment of:

Department of Homeland Security Urban Areas Security Initiative (UASI) Grant 03-TU-03; and
Department Justice Office of Community Oriented Policing Services (COPS) Grant 2003CKWX0199

September 2005

Anoop Singhal, Sushil Jajodia, and Terrence P. Ryan

University Consortium for Infrastructure Protection

Managed by the
Critical Infrastructure Protection Program
School of Law
George Mason University



– **Notice** –

This research was conducted as part of the National Capital Region Critical Infrastructure Project, carried out by the University Consortium for Infrastructure Protection, managed by the Critical Infrastructure Protection Program, George Mason University, John A. McCarthy, Director and Principal Investigator.

It was sponsored by the U.S. Department of Homeland Security's Urban Area Security Initiative grant #03-TU-03, and by the U.S. Department of Justice Community Oriented Policing Services Program grant #2003CKWX0199, under the direction of the Senior Policy Group of the National Capital Region.

The views expressed are those of the authors, and do not necessarily reflect the views of the Department of Homeland Security, the Department of Justice, or the Senior Policy Group of the National Capital Region.

Copyright © 2005 by George Mason University

Published in 2006 by George Mason University

**National Capital Region
Critical Infrastructure Protection Plan**

A Database and Architecture for Comparing Vulnerability Assessment Elements

September 2005

**Anoop Singhal
Sushil Jajodia
Terrence P. Ryan**

This Page Intentionally Blank

Table of Contents

1. Introduction.....3
2. Database and Architecture for Comparing Vulnerability Assessment Elements.....4
3. Software Architecture and Design.....8
4. Conclusions.....10
Appendix A: Critical Infrastructure Vulnerability Assessment (CIVA) Software User Guide

Appendix B: Critical Infrastructure Vulnerability Assessment (CIVA) Software Installation
Guide, CIVA Software Release version 1.0

Critical Infrastructure Vulnerability Assessment Software (Enclosed separately)

List of Tables

Table 1: Distribution of VA Tools by Sectors.....4

List of Figures

Figure 1: Data Model.....8

Acknowledgements

This research was conducted as part of the National Capital Region Critical Infrastructure Protection Project, carried out by the University Consortium for Infrastructure Protection, managed by the Critical Infrastructure Protection Program, George Mason University, John A. McCarthy, Director and Principal Investigator. It was sponsored by the U.S. Department of Homeland Security's Urban Area Security Initiative grant #03-TU-03 under the direction of the Senior Policy Group of the National Capital Region. The views expressed are those of the authors, and do not necessarily reflect the views of the Department of Homeland Security or the Senior Policy Group.

The relational database was developed by Anoop Singhal, PhD and Sushil Jajodia, PhD at the Center for Secure Information Systems, George Mason University, Fairfax, Virginia.

1. Introduction

Conducting vulnerability assessments of complex critical infrastructure - with interdependencies among a variety of sectors such as electricity, water, chemical, transportation, telecom, and the related networks - is a challenging task. Owner/operators and industry associations have developed detailed specialized assessment methodologies for their specific facilities or functional area. However, most of these assessment tools do not have specific details regarding functions or infrastructure asset and features outside of the developer's expertise. For example; an assessment tool might be very comprehensive in the area of physical security, but examine relatively little the areas of cyber security or human resource management. This project seeks to overcome this situation through the development of a relational database to categorize the components of risk and vulnerability assessment frameworks to allow users a quick way of comparing and contrasting the components of different tools, and enable users to extract selected elements according to their needs.

1.2. A search for open source vulnerability assessment frameworks was conducted and sixty-three tools and procedures were identified for study. Examples of some of the tools identified are:

- FEMA, Reference Manual to Mitigate Terrorist Attacks Against Buildings¹
- USCG/DHS, Outer Continental Shelf Facility Security Plans Review²
- Vulnerability Assessments and Needs Prioritization for the National Capital Region³
- U.S Department of Commerce, "Department of Justice Standards for Protection of Federal Facilities"⁴
- AMSA, "Asset Based Vulnerability Checklist for Wastewater Utilities"⁵
- Department of Justice (DOJ), NIJ Special report, "A Method to Assess the Vulnerability of U.S. Chemical Facilities"⁶

The following table shows the distribution of documents collected by sector.

Table 1: Distribution of Risk/Vulnerability Assessment Tools by Sectors

Sector	Number Reviewed
Transportation	6
Information Technology and Telecommunications	15
Energy	11
Water	8
Cross Sectors	9
Health	2
Emergency Services	6
Banking and Finance	6

2. Database and Architecture for Comparing Vulnerability Assessment Elements

2.1. The vulnerability assessment framework database is an *entity-relation* model. The three elements of an entity relation model are:

- Entity Sets: This represents a collection of similar entities. For example, if you are doing a data model for a University, *faculty* will be one entity set, *student* will be another entity set and *courses* will be a third entity set.
- Attributes: The properties of an entity set are represented as *attributes*. For example the entity set *faculty* will have attributes such as *name*, *address*, and *Social Security Number*.
- Relationships: These are connections among two or more entity sets. For example *teaches* is a relationship between *faculty* and *courses*. Similarly, *enroll* is a relation between *courses* and *students*.

In the vulnerability assessment framework database, the entities are the following categories:

- Building security / Facilities management
- Computer security / Information Technology systems
- Human Resources
- Interdependencies
- Finance

Each of the categories has associated sub-categories. For example, *Facilities Management* has the sub-categories of:

- Asset Value Assessment
- Threat / Hazard Identification
- Vulnerability Analysis
- Site
- Structural System
- Architecture
- Building Envelope
- Utility Systems: power, gas, water, wastewater, and communications
- Mechanical Systems (heating, ventilation and A/C)
- Plumbing and Gas Systems
- Electrical System
- Fire Alarm System
- Communication and Information Technology System
- Equipment Operations and Maintenance
- Vehicle Depot and Dispatch
- Materials Supply and Storage
- Security Systems
- Security Master Plan
- Hazardous Material: Chemical, Biological, and Radiological Considerations

Each of the sub-categories also contains additional sub-categories. For example, the sub-category “Site” has the following sub-categories:

- Land Use Considerations
- Site Design
- Site Layout and Form

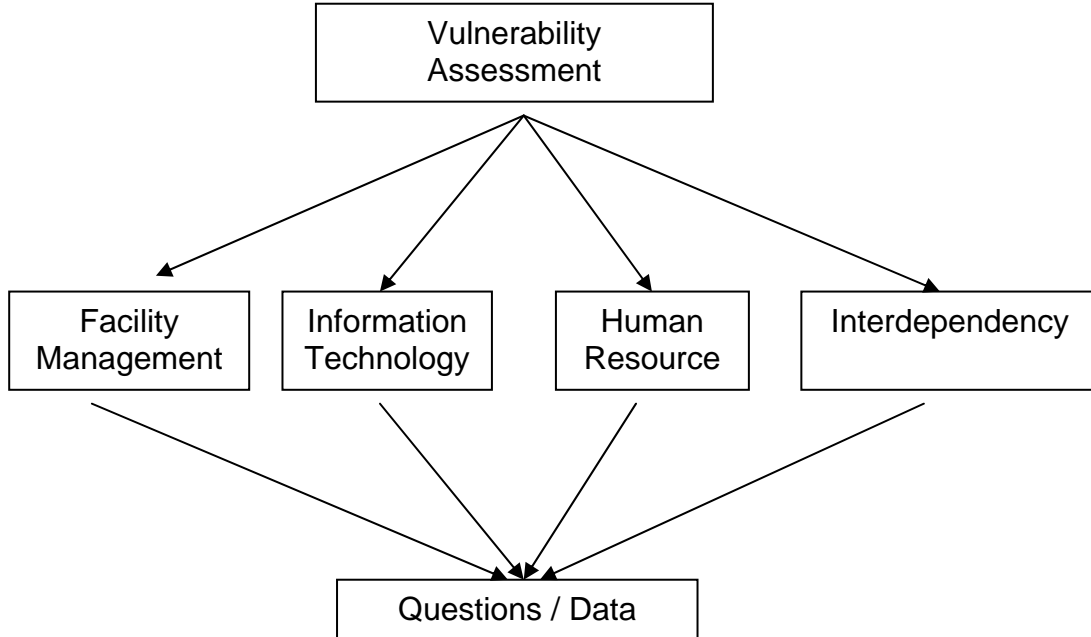
- Vehicular and Pedestrian Circulation
- Infrastructure Connections
- Landscape Design
- Standoff Distance
- Controlled Access Zones
- Entry Control and Vehicle Access
- Parking
- Loading Docks and Service Areas
- Security Lighting
- Other Site Considerations

Lastly, the sub-categories contain the assessment elements or questions. These “questions” are entities with attributes such as source or reference to a section, page, and vulnerability tool from which this question was taken. This final categorization provides the basis for comparison of the tools and procedures.

2.2. A data model was created in Unified Modeling Language (UML) and Extensible Markup Language (XML) to capture the information about the categories, sub-categories, and questions. Sixty-three different vulnerability assessment procedures, processes and tools were loaded into the database at the highest level of categorization. Of these, nineteen were categorized to the lowest level. Figure 1 shows a high level description of the data model. UML is a standard for defining an *object model* for the system. UML supports class diagrams and the following kind of relations

- Associations: These are binary relations among objects
- Aggregation and composition hierarchies: These relations are used to model that one object type contains other object types.
- Generalization/specialization hierarchy: These relations are used to model *is-a-kind-of* or *inheritance* relations among objects.

Figure 1: Data Model



This data model was used to create a database schema with sorting/searching capability in ORACLE Relational Database System. A Graphical User Interface (GUI) was designed using JAVA/HTML to query and analyze this data. Some sample queries are:

- Give all questions for a certain category (e.g. building envelope)
- Give all questions pertaining to a pattern such as “passwords” or “encryption”
- Export the results of the query to a file.
- Given a source document, give all the categories and sub-categories that are contained in that document.

3. Software Architecture and Design

3.1. The components of the system are:

- An object model was created in UML and XML using Rational Rose.
- This object model was used to create a database schema. This database schema was used to load questions, guidelines and categories.
- A data model and a database schema was designed to store information from different vulnerability assessment tools and methodologies into ORACLE DBMS. The information consisted of questions, categories and sub-categories contained in the documents.
- A Web based GUI was designed using JAVA/HTML to navigate, browse and search the information stored in the database. The GUI also enabled the users to browse the PDF files for vulnerability assessment tools and procedures. The different options that are available to the user are as follows:
 - Browse and search the files for assessment tools and procedures.
 - Quick search on questions and categories in assessment tools and procedures.

- Available online for stakeholders (password protected if necessary)
- Searchable by keyword and category
- Customizable database output
- Expandable by further research

4. At the completion of project, database files, a user guide, and installation instructions were recorded on a CD. The user guide and installation instructions are attached as Appendix A: Critical Infrastructure Vulnerability Assessment (CIVA) Software User Guide and Appendix B Critical Infrastructure Vulnerability Assessment (CIVA) Software Installation Guide, CIVA Software Release version 1.0. The software is enclosed separately.

4. Conclusions

In conjunction with the *Criteria and Evaluation of Vulnerability Assessment and Risk Management Tools and Procedures*⁷ this analysis serves as a means to improve the quality of the risk management process. The relational database categorizes the components of vulnerability assessment frameworks and gives users a quick way of comparing and contrasting the elements of different tools, and extracting selected elements according to their needs.

Endnotes:

¹FEMA, *Reference Manual to Mitigate Terrorist Attacks Against Buildings*, Risk Management Series, FEMA 426 (December 2003).

²USCG/DHS, *Outer Continental Shelf Facility Security Plans Review*, (USCG-2003-14759).

³*Vulnerability Assessments and Needs Prioritization for the National Capital Region*, PUBP 710/722 Practicum (June 2003).

⁴U.S Department of Commerce, Manual of Security Policies and Procedures, *"Department of Justice Standards for Protection of Federal Facilities"* (April 2003).

⁵AMSA, *"Asset Based Vulnerability Checklist for Wastewater Utilities"* (January 2002).

⁶Department of Justice (DOJ), NIJ Special report, Final Version: *"A Method to Assess the Vulnerability of U.S. Chemical Facilities"* (November 2002).

⁷ *The Criteria and Evaluation of Vulnerability Assessment and Risk Management Tools and Procedures* is another deliverable of project and has been developed as a stand-alone application that can be hosted on UNIX servers. It is described in detail in another report in this series.

Appendix A: Critical Infrastructure Vulnerability Assessment (CIVA) Software User Guide

1. The Introduction Screen

1.1. The first html screen provides very brief overview of the CIVA tool.

1.2. At the bottom of the page there are three options:

- The first option can be selected if the user is willing to provide details like name, company, sector and contact information. These details are kept confidential, and are just used for book keeping purposes. (Note: User has to enter the details every time he visits the tool.) Once the registration option is selected, “First Name and Last Name” are mandatory, and remaining fields are optional.
- The Second option is selected (by selecting the link provided), when user wants to skip step 1(a) and does not want to provide any details. The user will be registered as Guest User, if this option is selected.
- If the user wants to cancel the action, the third option can to be **selected**

2. CIVA Tool Main Page

2.1. The first page after the user login (either by registration or as guest user) has three options:

- Browse and Search CIVA Tool
- Browse and Search Categories and Questions, and
- View Analysis of CIVA Tools

2.2. The First option displays a Page containing a list of 63 CIVA documents, and links to all the PDF documents are provided.

2.3. The Second option is selected to search CIVA questions database.

2.4. Third option is selected to view reports pertaining to CIVA Tool evaluation.

3. The CIVA Categories and Questions Search Page

3.1. As described previously, this page is reached by selecting the second option provided in **CIVA Tool Main Page**.

4. This page has three options:

4.1. The “**Search by Keyword**” option is selected, if user wants to select all the questions containing a particular *keyword*. This *keyword* search page contains a *textbox* in which the required *keyword* can be entered, or one of the *keywords* from *pop-up list* is to be selected. Once the *keyword* is typed and *submit* button is pressed, a page containing all the matched questions is

displayed. Each question has a unique *question identifier* and *source document*. The format of the *question identifier* can be obtained by clicking the highlighted link. *Source document* is the original document from which the question has been extracted. Click on the *Help* button provided at the top right corner of the page to view the detailed list of all the documents. All the questions displayed can be exported to a *Microsoft Excel Sheet* by clicking on *Export* button.

4.2. All the questions in the database have been classified into several *categories*, *subcategories* and *subsubcategories*. The “**Search by Category**” option is selected if user wants to search the questions based on a particular classification of categories. In the *Search by Category* page, please select a *category* from the pop-up menu that takes you to a page containing pop-up list of all *subcategories* of the selected *category*. By selecting a *subcategory* from the list, all related *subsubcategories* are displayed. Finally, all the questions related to the selected *subsubcategories* are displayed along with their *source code* and *source document* as described in *keyword search*. All the questions displayed can be exported to a *Microsoft Excel Sheet* by clicking on *Export* button. At any stage in this traversal the user can go back main search page by clicking *back to search* link provided in every page.

4.3. The “**Search by Document**” option is selected if user wants to search the questions based on a particular *source document* from which questions have been extracted. In *Search by Document* page, please select a *Document* from the popup menu that takes you to a page containing pop-up list of all *categories* of questions contained in that document. By selecting a *category* from the list all related *subcategories* are displayed. Finally, all the questions extracted from the *document* and classified under the selected *subcategory* are displayed along with their *source code* and *source document* as described in *keyword search* and *category search*. All the questions displayed can be exported to a *Microsoft Excel Sheet* by clicking on *Export* button. At any stage in this traversal, user can go back main search page by clicking *back to search* link provided in every page.

Appendix B: Critical Infrastructure Vulnerability Assessment (CIVA) Software Installation Guide CIVA Software Release version 1.0

1. System Requirements

Operating System:	SUN Solaris/LINUX/UNIX (any Flavor)
Web Server:	Apache HTTP Server Version 2.0 or above
Application Server:	Apache Tomcat/5.0.24 (or above)/ Any Latest Servlet Engine
Database:	Oracle9i Enterprise Edition Release 9.2.0.1.0 (or above)
Other Software:	J2SE 1.4.2 or above/ J2EE 1.4, MS Office 2003, JDBC driver compatible with version of Java

2. Placing JSP and other necessary Files in Tomcat Container (required files in “JSPFiles” directory of this package)

2.1. Install Tomcat version described in **system requirements** and make sure that “CATALINE_HOME” environment variable is properly set.

2.2. Make sure that a *context* is properly created with a relevant name for the *CIVA Tool*. (For simplicity, we refer to the context directory as *context* in the rest of this documentation.)

2.3. Place “JSP” directory in “./JSPFiles/” in the *context* directory created in step 1(b).

2.4. Also, place “SQLBEAN” directory of “./JSPFiles/” in the context directory created 1(b). (Make sure that in the *context* directory we have placed two subdirectories -- **1. JSP** and **2. SQLBEAN**)

2.5. Edit “OracleBean.java” file in */context/SQLBEAN/* (in step 1(d)) following the commented instructions provided within the file.

2.6. At the *context* directory type following commands to compile the bean file: *UNIX> javac -d ./SQLBEAN/OracleBean.java*

2.7. Go to */context/WEB-INF/classes/* directory. Create a sub-directory titled “SQLBEAN”.

2.8. Place the class file (“OracleBean.class”) generated in step 1(e) in the “SQLBEAN” directory created in step 1(f).

3. Placing Files in Apache (required files in “htmlFiles” directory of the package)

3.1. Place “CIVA” sub-directory of “./htmlFiles/” in appropriate sub-directory of “public_html” directory of Apache (or any web server you have installed)

3.2. After placing in Apache, edit “Assessment.html” file in CIVA directory as follows: Search the file for “bladerunner.csis.gmu.edu:8080/Addada/..” and replace with appropriate path pointing to the directory (JSPFiles) in your Tomcat (or other Servlet engine deployed on your server) where the JSP files are placed.(See step 1(c)).

4. Importing CIVA questions Database (required files in “OracleTableDump” directory)

4.1. Make sure that Oracle 9i is installed properly and necessary table space is assigned properly.

4.2. Get the file “CIVA-Tables.expdat” from “./OracleTableDump” directory.

4.3. At UNIX prompt type the following import statement: *UNIX> imp userid/password tables=(questions,cat_subcat,subsubcat,documents,guidance) file=CIVATables. Expdat*

4.4. All the tables are now imported to the local database server. For verification, schemas for the tables are provided in the appendix.

5. Schemas for Database Tables

Table 1. documents (docno number(3) not null,
doctitle varchar2(200),
primary key (doctitle));

Table 2. cat_subcat(category varchar2(200),
subcategory varchar2(500),
primary key (category,subcategory));

Table 3. subsubcat(category varchar2(200),
subcategory varchar2(500),
subsubcategory varchar2(500),
primary key(category,subcategory,subsubcategory),
foreign key(category,subcategory) references cat_subcat);

Table 4. questions (question varchar2(4000) not null,
category varchar2(200),
subcategory varchar2(500),
subsubcategory varchar2(500),
question_code varchar2(100),
sourcedoc varchar2(200),
primary key(question_code),
foreign key(category,subcategory,subsubcategory) references subsubcat,
foreign key(sourcedoc) references documents(doctitle));

Table 5. userslist (firstname varchar2(15),
lastname varchar2(15),
company varchar2(30),
title varchar2(30),
email varchar2(20),
address varchar2(200),
sector varchar2(30),
primary key(lastname));

Table 6. guidance (question varchar2(4000),
question_code varchar2(100),
primary key(question_code));

This Page Intentionally Blank