BUILDING IN WEB APPLICATION SECURITY AT THE REQUIREMENTS STAGE: A TOOL FOR VISUALIZING AND EVALUATING SECURITY TRADE-OFFS

A thesis presented in partial fulfilment of the requirements for the degree of

Master of Information Science
in
Information Systems

at Massey University, Albany,
New Zealand.

Natalia Alekseevna Nehring

2007

Abstract

One dimension of Internet security is web application security. The purpose of this Design-science study was to design, build and evaluate a computer-based tool to support security vulnerability and risk assessment in the early stages of web application design. The tool facilitates risk assessment by managers and helps developers to model security requirements using an interactive tree diagram. The tool calculates residual risk for each component of a web application and for the application overall so developers are provided with better information for making decisions about which countermeasures to implement given limited resources for doing so. The tool supports taking a proactive approach to building in web application security at the requirements stage as opposed to the more common reactive approach of putting countermeasures in place after an attack and loss have been incurred. The primary contribution of the proposed tool is its ability to make known security-related information (e.g. known vulnerabilities, attacks and countermeasures) more accessible to developers who are not security experts and to translate lack of security measures into an understandable measure of relative residual risk. The latter is useful for managers who need to prioritize security spending.

Keywords: web application security, security requirements modelling, attack trees, threat trees, risk assessment.

Acknowledgements

This work has been made possible with the significant help and contribution of a number of special people. I would like to express my sincere thanks to all of them.

First, I would like to thank my supervisor, Dr. Ellen Rose, for her guidance, support, expertise, patience and friendship. She has always been passionate with research and patient with me in explaining different things and available to help me whenever I needed. I was welcomed into her office for help at any time. Dr. Ellen Rose has always shown other angles and made perceptive remarks that have guided my research in the right direction. She has contributed significantly to my research. I really appreciate this help; especially as English is not my first language. Ellen is the person who has set up a valuable example for me to follow, and from whom I have learnt a lot in the last couple of years. She has helped me in developing academic skills which were important for researching at the required level. She has encouraged me in my efforts, prompted me to correct my mistakes and taught me that the best way of doing things is to do your best to make progress as you go along rather than rushing through at the last minute.

I would like to acknowledge the value of two scholarships: the Massey University Masterate scholarship and the NZ Federation of Graduate Women North Shore Branch Top scholarship, which provided me with financial support during the time of my research.

I would like to express my great appreciation for members of *The Code Project* website (www.codeproject.com). This web site is a great source of programming materials, including articles, examples and source code for different programming languages. Some components from this web site made the implementation of the proposed tool easier. Also, I used examples of code from this web site to learn C#; a new programming

language for me.

I must also thank the staff of two companies for giving me their time and helping me with the process of evaluating my tool.

I would like to express very special thanks to my husband Croydon Nehring for being such a supportive person. He has been supportive and encouraging in my doing this research. He has also been a care person for our kids, and has replaced me for them when I was busy. He was with me in my ups and downs that go with a research effort of this scope, and encourages me to be a warrior, to do my best and to look forward.

Other special thanks go to my parents, Alexei and Nelli Gorobets, for their support with providing childcare for my kids. Big thanks go to my children, Maria, Anastasia, Timofei and Nicolai, for their enthusiastic and unconditional love, which has always cheered me on.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction and Background

1.1 Introduction

The security of web applications has become a central issue for online businesses. The e-Crime Watch Survey (2004) found that 40% of businesses feel hackers represent their greatest cyber security threat (CSO magazine, 2004). The 2006 Deloitte Touche Tohmatsu Global Security Survey of top financial institutions recently reported a shift from infrastructure to application layer attacks (p. 35) as well as the following findings. Only 7 percent conduct quarterly security code reviews, 2 percent do semi-annual reviews, 65 percent do ad hoc reviews and 13 percent never do reviews. The number of online attacks reported in this annual survey grew by 25 percent with 78 percent reporting security breaches from external attacks. In the Asia-Pacific region, excluding Japan, the number of online attacks grew from 16 percent in 2005 to 100 percent in 2006; every organisation surveyed in the Asia-Pacific region had been attacked a minimum of once during the 12 month period (Deloitte Touche Tohmatsu, 2006). As current web sites are more likely to be complex online information systems and not just simple HTML pages, web site security has become more complicated. The security of a web site has a number of dimensions; one of them is web application security. John Pescatore, an analyst at Gartner Inc. in Stamford, Connecticut said "web application security is a serious problem for two-thirds of all corporate web sites" (Verton, 2002, p. 9). Unfortunately, the growth in security problems is keeping pace with growth in the number of Internet users and companies using web sites to carry out business online.

In contrast to the predominantly reactive security practise of detecting and correcting web application security problems, this thesis work seeks to design and develop a tool to support web application developers in taking a proactive approach to building in web application security at the requirements stage. The IT community knows about countermeasures, security patterns, attack patterns and existing vulnerabilities but people are still developing web applications which are not secure. To solve this problem, there is a need to make this information more accessible. Since managers must see a reason to invest in security measures, the ability to more effectively assess

risk and the potential loss of not implementing security is needed. This research has designed and prototyped a tool that provides support to both managers and developers in making these tough decisions.

1.2 Research Objectives and Contributions

The Design Science research approach has been taken (Gregg, Kulkarni, & Vinze, 2001; Hevner & March, 2003; Hevner, March, Park, & Ram, 2004; Zelkowitz & Wallace, 1998) in order to achieve the following research objectives:

> Objective 1: To design and prototype a tool for use by managers and developers for visualizing and evaluating security trade-offs and risks in alternative web application designs.
>
> Objective 2: To demonstrate the utility of the tool via evaluation in a real web application development environment. Utility has been measured in terms of user satisfaction with the tool's ability to support risk assessment and to facilitate identification of vulnerabilities during the requirements stage.

The proposed solution is a tool for visualizing and evaluating security trade-offs in alternative web application designs. The tool is designed to help developers visualize attack patterns and build threat trees in order to identify potential vulnerability points in web applications. It also provides the ability to assess risk and to identify trade-offs in order to determine which security requirements should take priority. The tool can generate visual representations of attacks and vulnerabilities for different kinds of web applications to help developers identify and prioritise security requirements rather than reacting to security problems after they happen.

The prototype tool stores information on language independent web application vulnerabilities. Any language-specific problems are delimited to web applications written using PHP and the MySQL database. To get businesses to take proactive security measures more seriously we need to reduce the up front cost for security risk analysis. There is a need to reduce the learning curve and improve access to existing knowledge about potential threats, web application vulnerabilities, countermeasures and potential losses from not implementing countermeasures.

1.3 Background to the Study

The May 2007 Netcraft survey reported the existence of 118,023,363 web sites, an increase of 12.8 million from the 2006 total of 30.9 million (Netcraft Ltd, 2007). The current state of security of such sites was underlined by Auronen (2002, p. 2) who stated that sensitive data is "usually protected by only weak access control mechanisms vulnerable to many types of attack". Database driven web applications are the heart of today's web sites. Given their central role, security requirements should be considered from the initial stage of web application development. Writing code with security in mind could help to make web sites more secure against a wide variety of known attacks. However, a 2006 survey of top financial institutions around the world by Deloitte Touche Tohmatsu shows only 26 percent of respondents named application security as a top priority. 56 percent of respondents stated that poor software development compromises quality and may become a security threat in the future (Deloitte Touche Tohmatsu, 2006).

Potential for exposure must be continually assessed during the iterative process of web application development to ensure changes don't introduce new vulnerabilities and to ensure that protection exists from newly discovered types of attacks. Security breaches can affect the organisation that owns the web site, but can have an even greater impact on customers when private information is revealed or financial losses are incurred (Schneier, 2004).

Security assessment should be thought of as an ongoing process, not a one shot accident according to the Open Web Application Security Project (OWASP, 2005c). This process includes a number of steps. First of all it is necessary to define and know your enemy – vulnerabilities of web applications. Organisations such as the Computer Emergency Response Team (CERT) often publish known vulnerabilities. A list of the ten most dangerous can be found at OWASP (2005b). The second step is taking a proactive approach to ensure security, like building security into the design of web applications. The remaining steps are reactive. They include monitoring web site activity regularly and using this information to maintain running web applications in terms of security enhancements to ensure changes in requirements will not compromise security. Figure 1.1 shows a set of best security practices, meant to be

followed during software development. This research focuses on two early stages of web application development: security requirements and risk analysis, highlighted in Figure 1.1. The study produced a prototype of a software-based security tool to support these two stages. The tool brings together existing security knowledge to reduce the effort required to conduct risk assessments for web applications.

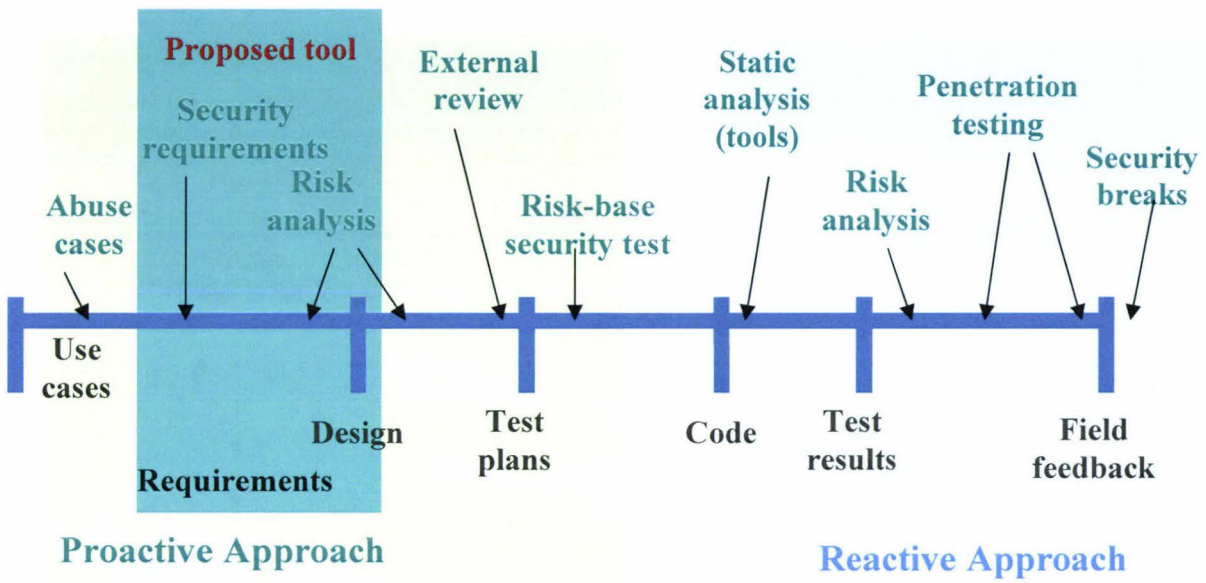


Figure 1.1 Best security practices for software development.

Adapted from: (McGraw, 2004)

An extension of the model shown in Figure 1.1 is illustrated in Figure 1.2 (Barnum and McGraw, 2005). The extension points out specific types of security knowledge (e.g. principles) and identifies the security activities (e.g. risk analysis) in the software development life cycle (e.g. design stage) where the knowledge is likely to be of greatest use. An understanding of these relationships provides a solid base for software security best practice. This becomes extremely important in practical software development given the industry faces a shortage of experienced security experts (Barnum & McGraw, 2005). Barnum and McGraw define three knowledge categories: prescriptive, diagnostic, and historical. The prescriptive knowledge category includes actions or procedures which need to be followed, like data principles, guidelines, and rules. Attack patterns, exploits, and vulnerabilities help in determining the capability of a component to perform its functions and are therefore classified as diagnostic knowledge. Prior diagnostic knowledge helps the practitioner

to understand the real problem based on extensive experience with the same or a similar problem. Common security problems like vulnerabilities and corresponding attacks can be detected and dealt with using prior experience with these problems. This category of knowledge helps in recognising common problems and is invaluable during the development stage. Information that helps to define previously existing risks belongs to the historical knowledge category (Barnum & McGraw, 2005). In relationship to Figure 1.1, this research seeks to design a tool which supports the definition of security requirements in terms of vulnerabilities, known attacks on each type of vulnerability and known countermeasures to reduce the potential damage from an attack. The tool provides a database of vulnerabilities, attacks and countermeasures to support doing a risk assessment in the early stages of web application development. The tool calculates the risk for each component of the web application being assessed and stores this information so it will be available for managers to view at later dates for the purposes of doing what-if analyses and making comparisons between different risk mitigation strategies in terms of residual risk (i.e. unmitigated risk) and the costs associated with implementing countermeasures.

Different types of security knowledge can trigger security activities at different stages of software development. Conducting security assessment activities during the early stages of development is referred to as the proactive approach. Knowledge about attack patterns can be applied at the requirements and design stages to conduct risk assessments. This knowledge is also useful at the test plan creation stage for running risk based security tests. Figure 1.2 claims knowledge about vulnerabilities is only used in the later stages of development. In reality, vulnerability knowledge can also be useful in the early stages of web application development as part of a proactive approach. In contrast, a reactive approach seeks to discover vulnerabilities in the code after it has been released. A proactive approach seeks to prevent or reduce vulnerability (i.e. weaknesses) in the code during development. Developers need to have knowledge of potential vulnerabilities and attacks before they can consider countermeasures to reduce or remove vulnerabilities. Knowing about vulnerabilities before coding helps to save time at later stages where these identified problems can cause significant delays in further development and/or releasing the software. Knowledge about attack patterns can assist developers in writing security requirements and in providing protection against particular identified attacks. This

knowledge can also be used to write risk-based security tests. A principle is defined as a statement of existing security knowledge, which comes from an experienced practitioner and from real-world knowledge of building secure systems (Barnum & McGraw, 2005). Principles are helpful in two ways: in detecting architectural defects in software, and in promoting good security practices. A principle is often documented using a title, description, examples, references, related rules and guidelines. Guidelines can be useful for creating security requirements and evaluating alternative designs (See Figure 1.2.)

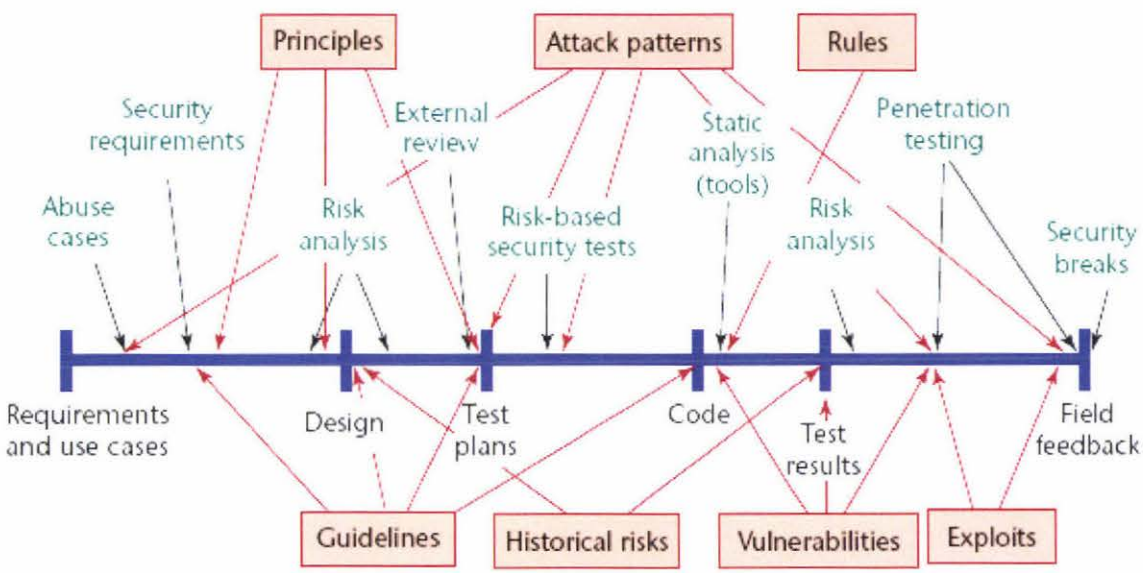


Figure 1.2 Applying security knowledge during the software development life cycle. Source: (Barnum & McGraw, 2005, p.3)

Historical risks are detailed catalogues with descriptions of specific issues which were discovered in real-world software development. For example, a list of discovered software vulnerabilities (i.e. weaknesses) is a type of historical risk. Each risk item of this catalogue has a statement of impact on the business. Historical knowledge can become a valuable security resource, which helps to identify similar issues in new software development. A catalogue of historical knowledge can save developers time and effort in identifying potential security issues. The proposed tool makes a database of historical security information available to web application developers.

Gathering and interpreting available data on vulnerabilities can be an onerous task, taking a considerable amount of a developer's or analyst's time. The old proverb: "A picture is worth a thousand words" implies people may absorb complex information more readily from pictures than from large volumes of text. Visual representations of complex relationships amongst web application components, their vulnerabilities, attacks based on these vulnerabilities, and the magnitude of potential losses can quickly highlight areas of major concern, facilitating security requirements analysis and risk assessment.

1.3.1 Risk

Security assessment is often associated with the concept of risk. Risk can be viewed as a function of the likelihood that a threat will materialise, the level of vulnerability and the potential for loss of resources. Thinking about negative scenarios in these terms is an essential skill for a test engineer (Alexander, 2003). In this sense, a web application designer should also think about requirements in terms of negative scenarios, that is, from a hacker's point of view. An understanding of vulnerability, threats and attacks is relevant to risk measurement (Amoroso, 1994) where:

- A vulnerability is a characteristic (or weakness) of the software that makes it possible for a threat to occur.
- A threat can be defined as an event which can have an undesirable effect on assets and resources (e.g., loss of data, corruption of data, exposure of confidential information).
- An attack is an action by a malicious user that involves exploiting vulnerabilities in order to cause a threat to occur.

Vulnerability and attacks are only of concern if they introduce the potential for threats that would involve significant resource loss (Amoroso, 1994). If you increase any of these three variables, risk also increases. If you reduce them, it decreases. More formally risk has also been defined as "the probability of a vulnerability being exploited in the current environment, leading to a degree of loss of confidentiality, integrity, or availability, of an asset" (Microsoft Corporation, 2006b, p.27). The potential impact of a threat is related to the degree of a resource's vulnerability as well as the resource or asset's value. The higher the value of an asset, the greater the

potential loss will be from realisation of a threat, and consequently the higher the degree of risk (Amoroso, 1994).

Risk needs to be assessed and managed. Microsoft Corporation recently published a document called "The Security Risk Management Guide" (2006). The document outlines an iterative, four phase (i.e. assessing risk, conducting decision support, implementing controls and measuring program effectiveness) proactive approach to risk management based on industry standards. The goal of Microsoft's approach is to balance cost and effectiveness. Qualitative steps for identifying the most important risks are followed using a process which starts with identifying roles and responsibilities. Managers are responsible for assessing asset value and potential impact of a risk, security personnel are responsible for identifying the likelihood of a risk occurring by taking current and proposed countermeasures/controls into consideration and developers are responsible for implementing the countermeasures for risks identified as unacceptable. In this guide, risk management is defined as "the overall effort to manage risk to an acceptable level across the business" and risk assessment is defined as "the process to identify and prioritize risks to the business" (Microsoft Corporation, 2006b, p. 16).

Similarly, Boehm (1991) identified the two main stages of risk management as risk assessment and risk control (Boehm, 1991). Risk assessment includes risk identification, risk analysis, and risk prioritization where (Boehm, 1991):

- Risk identification results in a list of project-specific vulnerabilities which can be dangerous for a project
- Risk analysis assesses the loss probability and loss magnitude for each vulnerability and
- Risk prioritisation ranks the risks in order of those to be dealt with in descending order of urgency.

Today, most organisations understand the importance of risk management and assessment but still experience difficulty with the application of modelling techniques to both risk management and risk assessment. One reason it is difficult to utilise these techniques, is the lack of advice on what to do and how to do it. Security analysts can

define vulnerability, but it can be difficult to see the overall picture with respect to evaluating the impact in terms of costs incurred for either choosing to mitigate or not mitigate specific threats associated with specific vulnerabilities. It is rare for an organization at the project management or portfolio level to use a risk management tool or framework to assess a risk, and identify its impact (Steven, 2006).

Organizations need more accurate information and more accessible information for risk assessment. The right information should be present in a familiar way and be easy for non-security experts to use. A risk calculation in these terms can give a business an improved ability to make better decisions on how much to spend in order to achieve a desired level of security (Steven, 2006). Numbers are not magic, but with the right information from experts they can serve as advisory indicators for a security decision. Serious application level security problems are still present in professionally designed web applications. To address web application security problems, decision-support tools and techniques are needed (Scott & Sharp, 2003).

1.3.2 Why software tools for risk assessment are needed

Proactive approaches to security involve consideration of the risk level which in turn depends on the likelihood of particular threats, the potential for loss, the effort required to execute particular kinds of attacks and the level of vulnerability as well as dependencies between these factors. The following sections summarise key aspects of the need for software tools to support risk assessment, as a part of a proactive security practice. "Security is a process, not a product, but we still need accurate and reliable products to calculate security quantitatively to improve security" (Sahinoglu, 2005, p. 23). Security should not be treated as an add-on feature. Security should be considered from the requirements stage, as a key system requirement, especially for systems that utilise components in both public and private networks. All possible security requirements can not usually be implemented, because available resources are limited. Every software project has limitations in terms of available time, budget and expertise. A change in mindset is required where the following points are considered (Feather, Sigal, Cornford, & Hutchinson, 2001):

- quality and risk estimation can be as important as budget and schedule

- limited project resources should be more optimally allocated.
- trade-off opportunities should be identified and evaluated

To achieve more optimal allocation of resources, managers need better information and more cost-effective ways of analysing that information. It is important for managers to understand the potential costs of not implementing countermeasures in order for them to make more informed decisions about allocating limited resources to security measures. In addition, security issues become more understandable to a business when they are expressed in familiar form. The question of "What data needs to be collected and what needs to be measured?" arises. Security risk assessment tools can provide decision support for managers who aim to balance the cost of a loss with the cost of countermeasures. Business leaders should ask the following questions about security (Geer, Hoo, & Jaquith, 2003):

- How secure am I?
- Am I better off than I was this time last year?
- How do I compare with my peers?
- Am I spending the right amount of money?
- What are my risk-transfer options? (Geer et al., 2003)

Due to the nature of the Internet, web based systems are vulnerable to outsider and insider threats. A number of reasons why web based systems are vulnerable include (Zhou, 2002):

- Web-based information systems can be accessed by any Internet user.
- System applications can be invisible and difficult to review.
- Unauthorised access can be hard to trace.
- The possibility of security breaches in web information systems is higher than in centralised systems. The effects can be costly: systems can be destroyed and sensitive information can be stolen.
- Data records can be accessed indirectly and modified by unauthorised persons.
- Numbers of attacks are rising due to two factors: the Internet is now widely available and many financial systems are now linked to the Internet (e.g. online banking, online trading).

To address security across a software project's lifecycle a number of factors need to be considered. They include security requirements specification from the viewpoints of

various stakeholders, specification of the environment in which the software will operate, specification of the software and hardware modules, and specification of the expected length of time the software will be used. The recent interest shown by companies such as Microsoft in the development of new security tools for the analysis and modelling of security requirements for web applications reveals a change in attitudes towards web application security in the industry. A computer-based security analysis tool can be a valuable aid to the process of risk assessment. Such a tool can provide assistance in the evaluation of which software risks need to be addressed first, helping to mitigate risk, and show the effectiveness of countermeasures (D.P. Gilliam, 2004).

1.4 Overview of the Thesis

This chapter outlined a current problem with web application security, namely the need to think about security early on and to make existing knowledge about vulnerabilities, attacks and countermeasures more accessible to developers and their managers so they can conduct risk assessments. In addition, the research objectives were stated and a brief background provided on why such a study is needed and who might benefit from the outcomes. Chapter 2 outlines the nature of the Design Science research method and explains why it is an appropriate approach for achieving the research objectives. The second chapter begins with a discussion of the generic steps in the Design Science research method then continues with specific details on how this research was done. Iteration through the design process, the changes made to the prototype after each cycle through the process, details on how the tool was evaluated and, how the findings led to prototype changes, as well as the steps involved in the implementation of changes are discussed. The remaining chapters are organised based on the steps and outcomes of the Design Science research method.

Chapter 3 discusses the proposed security tool's functional and non-functional requirements. Requirements are defined as what a system must do. They describe a necessary attribute, capability, characteristic, or quality in order to provide value for the tool's intended users (Sommerville, 2007). Functional requirements are necessary application related capabilities in terms of what the system should be able to do for

the user. Non-functional requirements are quality aspects such as usability, performance, reliability and safety. Chapter 4 discusses the tool's architecture and the detailed design of its major components. Chapter 5 presents the results of three rounds of external evaluations of the tool in terms of its ability to meet the requirements stated in Chapter 3. Chapter 6 relates this study to prior work and compares the proposed tool to two similar security analysis tools. This chapter also discusses how the positive and negative results from the final evaluation led to implications for the use of the tool in practice by different groups of users. Finally, Chapter 7 briefly restates the contributions of this research, discusses the limitations of the tool and draws implications for further research based on the identified limitations.

Chapter 2: Research Method

The Design Science research approach (Hevner & March, 2003; Hevner et al., 2004) was taken. This research approach utilises the system development lifecycle, prior theoretical knowledge and evaluation procedures, in order to build a specific artefact for a particular task (March & Smith, 1995). An artefact can be defined as a human-made object (Hevner et al., 2004). Problem solving in design research is basically a search process used to determine a useful solution (Hevner et al., 2004). This useful solution is utilised to develop an IT artefact which expands into, "the boundaries of human problem solving and organizational capabilities by providing intellectual as well as computational tools" (Hevner et al., 2004). Problems associated with the desire to decrease costs and optimise expenditures, through a redesign of business processes, are often suitable for this method. Information systems can play a major role in enabling effective business processes to achieve these goals (Hevner et al., 2004). The Design Science research approach has been used for creating artefacts, several categories of which will be discussed in this chapter.

Firstly, this chapter discusses the generic steps in the Design Science research method. Secondly, it explains the specific steps and procedures followed for this particular research, providing specific details on the activities used, and their outcomes. Included in this discussion are details on how each of the four major steps in each iteration was conducted. Changes to the artefact's (i.e., the prototype's) design, based on each evaluation, and for each iteration, are also discussed. The chapter explains when evaluations were done, who did them, how they were conducted, and how the findings from each evaluation fed back into changes in the tool's design.

Design research focuses on designing, building and evaluating an innovative IT artefact for a particular purpose (Hevner & March, 2003). This study's research design was based on the generic steps of the Design Science research approach, shown in Figure 2.1 (Vaishnavi & Kuechler, 2005). The approach requires multiple iterations through four major steps: Developing awareness of a problem's requirements; suggesting a tentative design as a solution to meet those requirements;

developing an artefact as an implementation of the suggested design; and finally evaluating the artefact and feeding back the results to refine the problem, leading on to the next iteration.

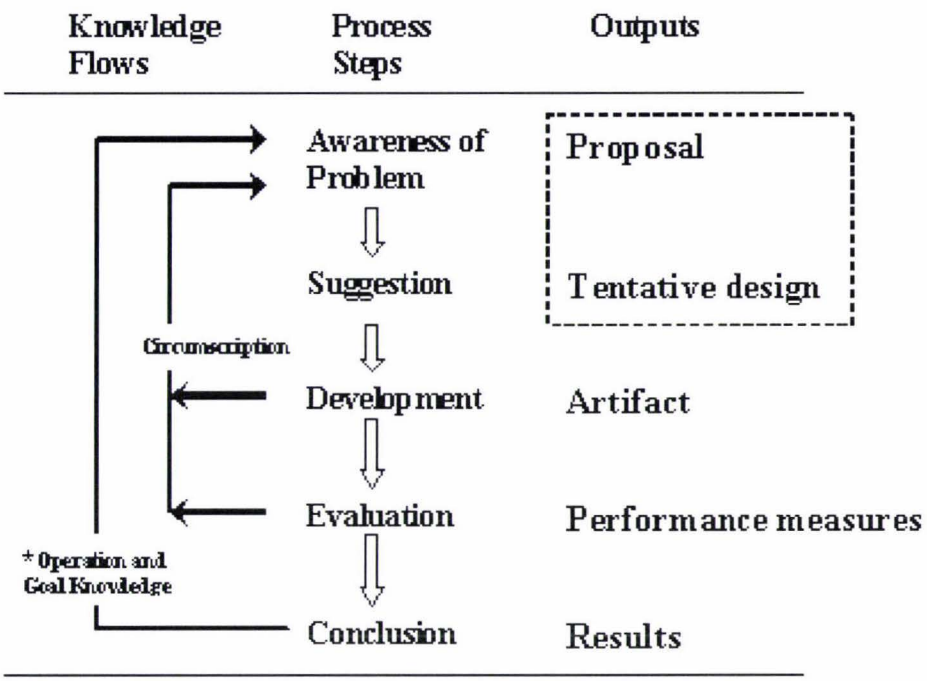


Figure 2.1 Steps in the Iterative Design Science Research Process

Source: (Vaishnavi & Kuechler, 2005, p.12)

The first step is to become aware of a problem. Literature, the popular press and Internet information can become primary sources of awareness of interesting problems in information systems. A new research problem can emerge from examining the interface between information systems and other disciplines, like engineering and business. A clear statement of the research problem; in the form of a research proposal; is the output of the first step (Vaishnavi & Kuechler, 2005). The research proposal identifies the problem, explains why it is important, who it is important to and identifies how the proposed study aims to address the identified problem. This entails stating the objectives of the research and the requirements of the problem, in terms of what needs to be done.

The second step is making a suggestion as to how to meet the problem's requirements; that is, proposing a tentative design to solve the problem. The suggested design may be part of the proposal, as shown by the dotted line in Figure 2.1

(Vaishnavi & Kuechler, 2005). The tentative design will be altered during the research process. The researcher may propose, revise, and evaluate, a number of alternative designs, in order to meet requirements which may be altered as a result of each evaluation of the prototype. Generally, trade-offs in terms of conflicting non-functional requirements may need to be considered when choosing a particular design solution as the input to the next step of the process.

The third step of Design Science research is labelled Development. The development step differs depending on the type of artefact and its purpose. According to March and Smith (1995) an artefact is the output of a Design Science research effort. It can be a construct (or concept), a model, a method, or an instantiation where:

- A construct is a high-level, well-defined concept, or key component of a theory which represents a key characteristic of a phenomenon;
- A model is a higher order construction, which is used to describe tasks, situations, or artefacts;
- A method is a way of describing how to perform goal-directed activities; and
- An instantiation is the implementation of an artefact in its environment. Information systems and computer based support tools are examples of instantiations (March & Smith, 1995).

Table 2.1 *The Design Science Research Framework*

		Research activities			
		Build	**Evaluate**	**Theorise**	**Justify**
Research Output	**Constructs**				
	Models				
	Methods				
	Instantiations				

Source: (March & Smith, 1995, p.5)

The research activities will differ depending on the artefact's purpose. Table 2.1 shows research activity as a vertical dimension and research output as a horizontal dimension. The Design Science intent is to build and evaluate artefacts (constructs,

models, methods and instantiations). The Natural Science intent is to theorise and justify (March & Smith, 1995). The novelty of an artefact is primarily reflected by the uniqueness of its design and/or how it is implemented (Vaishnavi & Kuechler, 2005). Problems, or opportunities, discovered during development may lead to design changes, which are shown as a feedback loop in Figure 2.1.

The fourth step is labelled Evaluation. Table 2.2 describes five Design Evaluation Methods. The Field Study, an observational method, was used in this research in combination with the Descriptive method. The Field Study provides independent, external evaluation of the tool's utility by multiple types of stakeholders (i.e. managers, developers and security experts). Scenarios and informed arguments based on the literature were also used to assess the artefact's utility.

Table 2.2 *Design Evaluation Methods*

1. Observational	Case Study - Study artefact in depth in a business environment Field Study - Monitor use of artefact in multiple projects
2. Analytical	Static Analysis - Examine structure of artefact for static qualities (e.g., complexity) Architecture Analysis - Study fit of artefact into a technical IS architecture Optimisation - Demonstrate inherent optimal properties of an artefact, or provide optimality bounds on artefact behaviour Dynamic Analysis - Study artefact in use for dynamic qualities (e.g., performance)
3. Experimental	Controlled Experiment - Study artefact in controlled environment for qualities (e.g., usability) Simulation – Execute artefact with artificial data
4. Testing	Functional (Black Box) Testing - Execute artefact interfaces to discover failures and identify defects Structural (White Box) Testing - Perform coverage testing of some metric (e.g., execution paths) in the artefact implementation
5. Descriptive	Informed Argument - Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artefact's utility Scenarios - Construct detailed scenarios around the artefact to demonstrate its utility

Source: (Hevner et al., 2004, p.22)

The criteria for artefact evaluation may be stated in the proposal as: the functional and

non-functional requirements which the artefact must satisfy in order to resolve the research problem. Functional requirements dictate what the artefact should do, while the non-functional requirements reflect quality criteria such as usability, extendibility, maintainability and performance. The evaluation phase has a sub-phase, which involves analysis of the differences between expected and actual behaviours of the artefact, as well as explaining them. The evaluation of a working artefact and its performance measures provides feedback as an input to the next round of suggestions for improvement to the artefact's design (i.e. the circumscription arrows in Figure 2.1) (Vaishnavi & Kuechler, 2005).

The final step is labelled the Conclusion. This step involves summarising the contributions and limitations of the study, in terms of the benefits the artefact provides and the problems which still need to be addressed to more fully mediate the problem which the artefact was designed to address. Suggestions for how future research might improve on the artefact's design and implementation should be made at this stage. This is shown as the feedback loop labelled *operation and goal knowledge* in Figure 2.1. Knowledge acquired during the research process may reflect relationships which can be tested, or applied, in other designs in future research. Furthermore, explanations of unusual behaviours can be useful subjects for further research (Vaishnavi & Kuechler, 2005). The following sections provide specific details on how each of the above generic steps within the Design Science research approach was executed in this study.

2.1 Awareness of a Problem

The first step is to become aware of a research problem. There were three main sources used to discover the research problem.

2.1.1 First source: Literature review

The Information Science literature was searched for relevant studies on web application security and risk assessment, using the following databases:

- ACM (Association for Computing Machinery) Digital Library;
- Business Source Premier; and

- IEEE Xplore (Institute of Electrical & Electronics Engineers).

As a result, a number of published articles from different academic sources were obtained, read and analysed. The review of the literature shows that extensive literature exists regarding risk assessment (Boehm, 1991) and there are a number of different approaches to carrying out risk assessment. However, there is a distinct lack of practical research in the area of tool-based support for security risk assessment (Kotulica & Clark, 2004). In addition, the literature review showed that web application security has become an important issue. The rapid development cycle for web applications makes it possible to release unsecured code. Several researchers have pointed out the need to build decision support tools to assess vulnerability, or risk, in web applications and web based information systems (Huang, Huang, Lin & Tsai, 2003; (Huang, Tsai, Lee, & Kuo, 2004; Huang, Yu et al., 2004).

New trends and new types of attacks relating to web application security have been discovered and discussed in prior studies (Holz, Marechal, & Raynal, 2006). Various approaches have been suggested to deal with specific problems. For example, a number of solutions directed at specific types of vulnerabilities, such as SQL vulnerabilities have been proposed (Buehrer, Weide, & Sivilotti, 2005; McClure & Kruger, 2005). Other researchers have suggested dealing with web application security by integrating security issues throughout the life cycle of the product, in order to improve overall security (Meier, 2006). The majority of existing tools (e.g. WAPT 5.0 (SoftLogica, 2007), WebInspect 7 (SPI Dynamics Ltd., 2007) and Paros (Chinotec Technologies, 2007) and security assessment services (Foundstone, 2007; Security-Assessment.com, 2007), however, focus on scanning completed web applications for a particular pre-specified set of vulnerabilities.

An additional source of web application security issues is the publicly available information on the World Wide Web. This information was searched by making queries on the search engine, Google. There exist a number of online centres for registering security problems (CERT Coordination Center, 2005; Internet Storm Center, 2005; OWASP Foundation, 2005a; Web Application Security Consortium, 2005), along with a large number of security assessment companies (Kavado Inc., 2005; Security-Assessment.com, 2007; Watchfire-Corporation, 2006).

From these investigations, an overall picture of current web application security was drawn. The IT community is aware of growing security problems with web applications. A recent report by Symantec showed that the number of web application vulnerabilities is growing (Symantec, 2007). Symantec documented 2,526 vulnerabilities in the second half of 2006. This is the highest volume in any six month period on record and is 12% higher than the volume reported in the first half of 2006. Web application vulnerabilities made up 66% of the total disclosed vulnerabilities during this period and 79% of all documented vulnerabilities were easily exploitable. Furthermore, 77% of all easily exploitable vulnerabilities affect web applications and 7% affect servers. Most of the easily exploitable vulnerabilities (94%) were remotely exploitable (Symantec, 2007). The above statistics indicate the existence of a significant problem. The design and development of a computer based decision support tool could make knowledge of known web application vulnerabilities and countermeasures for addressing them more accessible to developers who lack training in security requirements.

2.1.2 Second source: Existing tools

The second source for problem awareness was an investigation of the existing computer based tools used to deal with web application security. Two such tools were identified, downloaded and evaluated. They are the:

- Microsoft Application Security Threat Analysis & Modelling tool (Microsoft Corporation, 2006a). This tool was designed for use by developers, especially non-security subject matter experts. A user can build a new threat model from scratch or by using a wizard, template or previously created threat model. The process of threat modelling entails defining the application requirements in terms of user roles and use cases, defining the application architecture, building the threat model for this architecture and measuring the level of risk. The tool produces a threat model, a number of security artefacts and reports, which can be used by different types of users such as developers and testers; and the
- Practical Threat Analysis (PTA) tool (PTA Technologies, 2006). This tool was developed by PTA Technologies, an Israeli company. The tool aims to

support software developers and security specialists in conducting a quantitative risk assessment and in developing risk mitigation policies. The tool can be used to identify threats to each asset and to calculate the overall risk to the system as a whole. Risk level, potential damage and countermeasures are measured in dollar values. Countermeasures are prioritised based on the values of the assets they protect, the potential degree of damage the asset can incur, the likelihood of threats being realised and the degree of mitigation each countermeasure provides. Countermeasures can be marked as already implemented and a report on the reduction in overall risk can be produced (PTA Technologies, 2006).

2.1.3 Third source: Discussions with experts in the field.

The third source of problem awareness was interviews conducted with various web application developers and meetings with two representatives of a company which carries out security assessment and penetration tests for web applications. One important thing learned from these interviews was that current trends in the development of web applications remain separate from the processes used to implement security.

Activities to deal with security issues are conducted at the endpoint in the process; after the design and development of all other features of a web application have been implemented. Often security testing is done a few days before the application goes online. Another conclusion based on the interviews was that developers are often not aware of web application vulnerabilities. Some of the companies involved with these initial interviews said they have rules which developers are supposed to follow to ensure security, but often a developer can apply these rules without having full knowledge of what these rules protect against, or how effective they might be. Security for web applications seems to be treated as a stand alone activity and is often not embedded into the full process (i.e. entire life cycle) of web application development.

The Design Science approach is appropriate for addressing the problem of making knowledge regarding security more accessible to developers, and to ensure that

security requirements are well established and understood before the development process is started. This problem requires the development of a new approach to a solution, or alternatively a more effective (i.e., good value for effort), or efficient (i.e., saves time, or money), version of a current solution. There is the need for more effective ways of utilising existing security knowledge and more efficient means of determining how to allocate limited budgets for risk assessment. Even though countermeasures, security patterns, attack patterns and existing vulnerabilities are known, unsecured web applications are still being developed since developers are not usually security experts.

There is also a need to make this information more accessible to non-security experts. Security information for projects should be managed in a way that provides clear reasoning for making an investment in countermeasures. Effective assessment of risks and the potential losses of not implementing security countermeasures are needed. Security information needs to be available to developers in an accessible form which identifies countermeasures to be implemented, identifies their effectiveness in reducing the threat (or loss) from an attack and gives advice on how the implementation of countermeasures should be carried out.

Web application developers and project managers require easy access to prior knowledge about deficiencies inherent in web site developments. This is a major requirement for a tool meant to support such developers and managers. This tool must be easy to use and highlight security problems in web application designs at the earliest stage of development. This problem was discussed in the proposal for this research and constituted the output for the first step of the research process.

2.2 Suggestions

This research seeks to design and develop a software prototype of a decision support tool as a more effective and efficient proactive solution to the web application security problem. The intent was to provide tool-based support to both managers and developers in making security decisions during the early stages of web application development.

Table 2.3 *Awareness of Problem and Suggestions*

Conclusions from the awareness step	**Suggestions**
Security activities should be embedded in a web application's development from the outset. Security requirements should be easy to understand and clearly stated during the requirements stage of web development.	Develop a software tool which can help in stating security requirements based on the components that make up a web application. Web applications of different types contain a number of common components.
A developer needs to be aware of the possible vulnerabilities, attacks and countermeasures for a particular project. All security knowledge should be easy to access for non-security specialists.	A supportive tool should provide comprehensive security knowledge to non-security specialists. This knowledge should be in an easily consumable form that shows vulnerabilities, related attacks and related countermeasures for a particular project.
A manager must be able to easily evaluate and assess risk and the costs of both dealing with, and ignoring risks, for a web application project.	A risk assessment formula should include security entities, such as vulnerabilities, attacks and countermeasures. Security specialists should assign a relative rating (1=low severity to 10=high severity) for different types of security entities. They should also establish links between the security entities. This information will be stored and retrieved by the tool.

This stage of the research process focused on producing the requirements and a tentative design of a software-based tool for meeting those requirements. Table 2.3 provides details on how the requirements were determined and how these requirements were translated into an initial design. Firstly, from the awareness step, a number of conclusions were drawn leading to the suggestions shown in Table 2.3.

From the suggestions in Table 2.3, a number of questions arose:

- What data needs to be stored in a security knowledge database?
- How should the user interface be designed?
- What risk calculation formula should be used to balance accuracy with available data and usability? and

- What programming language should be used to implement a support tool?

The literature review and search for other tools continued during the next stage; that of development. During this time, more precise suggestions were made. These suggestions became the requirements for the initial prototype of the security tool and influenced further revisions, in subsequent iterations.

2.3 Development and Evaluation

The development and evaluation are two processes which are bound together in the research design with a number of repetitions, as shown in Figure 2.1. This is an iterative process. The development processes of this step were concerned with implementing or building the artefact. It was closely tied to the evaluation processes of this step, which provided feedback for improving either the design, the implementation of the design, or both.

The development of the Visual Security Analysis Tool went through a number of iterations. During this process, three versions of security tool prototypes were developed. These early prototypes helped to establish the design of the user interface, the structure of the database, the way risk would be computed and the inputs required for the risk calculation.

The main purpose of this evaluation process in a Design Science study is to demonstrate the utility of the proposed solution, and identify the design problem. The perceived value, effectiveness (i.e. quality) and efficiency of a design artefact must be carefully evaluated (Vaishnavi & Kuechler, 2005). The utility is generally assessed in terms of how well the proposed solution meets the proposed requirements, relative to other existing solutions. The guidelines presented by Hevner et al. (2004) underline that the evaluation of the artefact is crucial. This evaluation should demonstrate that the artefact solves a known problem in a more effective or efficient manner (Hevner et al., 2004). The reader should clearly see the proof of why the proposed solution is acceptable (or better than other available solutions) (Bernstein, 2005). The main purpose of the evaluation step is to assess the perceived utility of the proposed solution for practitioners and to determine whether or not they saw the tool as being something which they would adopt to improve web application development in terms of security. The other purposes of the evaluation were: 1) to determine whether or not

the prototype met the specified functional and non-functional requirements; and 2) to obtain suggestions for improvements to the design. The main purpose of evaluation in the early iterations is to identify the key issues and problems with the proposed tool. The users' suggestions, based on three external evaluations, were used to add additional requirements and to revise the design during each round of the tool's development.

Since the proposed security tool would be used by three different groups of people, it was evaluated by representatives of these different user groups. Security experts, managers and developers served as evaluators. The managers and developers were not security experts.

The final outcome from all iterations is to create a functional prototype of a security analysis tool, which satisfies the requirements summarised in the next chapter, Chapter 3. These requirements are the founding criteria for the artefact's evaluation and came from the literature review. They were revised later on based on suggestions made by participants during the three stages of evaluation. The criteria are specified in Table 5.1 (see Chapter 5) and the evaluators were asked to consider these criteria, as well as to use their own experience in terms of making suggestions for possibly missed, but important requirements. Suggestions from each evaluator were recorded during the relevant evaluation. Each evaluation provided summarised information for the next round of the Suggestion phase (see Figure 2.1).

The focus of each of the three iterations of development and evaluation are discussed in the remainder of section 2.3. The prototype's design initially focused on fulfilling the requirements identified in the literature and the previously discussed interviews. The design was then refined based on the results of three rounds of user evaluations of the prototype.

2.3.1 Iteration One – Initial prototype development and evaluation

The first prototype utilised the Security Meter risk calculation, proposed by Sahinoglu (Sahinoglu, 2005). Sahinoglu's model provides a quantitative technique to calculate risk, based on a tree structure diagram (see Figure 2.2) involving probabilistic values

of vulnerabilities, threats and lack of countermeasures. This aspect of the tool design was prototyped and evaluated by the researcher. One advantage of this approach is that it uses a tree diagram to visually model the risk calculation. Threat trees have been used in prior work as an analytical alternative in brainstorming possible threats, or simply for keeping track of threats in a list as they occur (Burns, 2005; Gegick & Williams, 2005; Howard, 2004; Thompson, 2005).

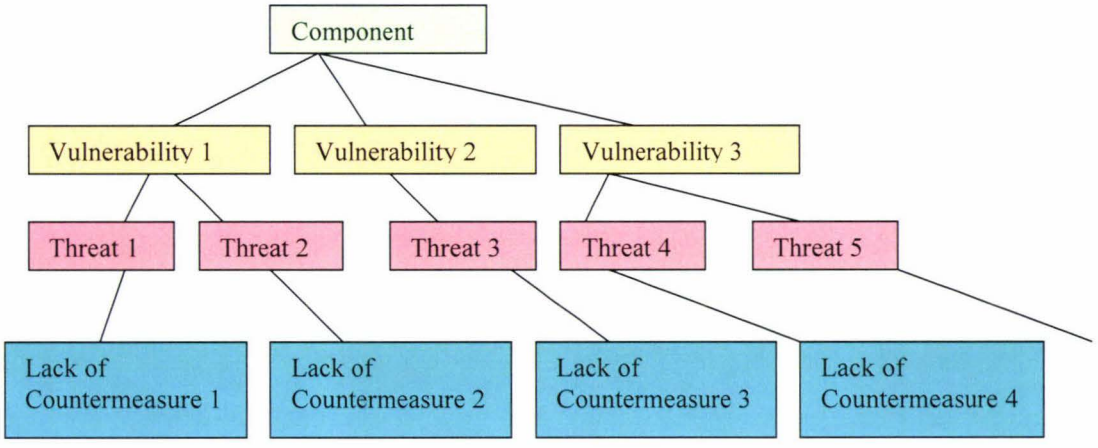


Figure 2.2 The Tree Diagram used by Security Meter.
Adapted from: (Sahinoglu, 2005)

The threat tree approach was designed to offset the negative aspects of arbitrarily listing threats (Amoroso, 1994). Threat tree modelling can be used as a proactive approach to securing a web site. Threat models can be created and maintained in parallel with the system specifications, or could be used in a bottom-up fashion to associate identified vulnerabilities to higher level threats. The purpose of threat trees is to reduce the vulnerability of the system. The tree diagram used by the Security Meter (see Figure 2.2) technique shows that any component can have a number of vulnerabilities. A particular vulnerability can cause one, or more, threats to the system. The lack of countermeasures for a particular threat can create a security problem. The tree diagram provides a visual representation of security system issues. There are two main drawbacks to the Security Meter approach. The main one is the difficulty in accurately determining probabilistic values for security entities such as threats, vulnerabilities, or countermeasures. The second drawback is that this model allows only one countermeasure for each particular threat, but in reality there can be

more than one. Due to these drawbacks, a new risk model using relative ratings was designed for use in the risk calculation, but the tree diagram was retained as a way of visualising the security entities comprising a web application.

An additional review of the literature regarding risk models was carried out with the aim to investigate risk calculations for tree-based models. Sima's (Sima, 2005) risk calculation formula seemed more useful for this research, for several reasons. First, it was easy to adapt for use with the tree structure diagram. Second, this approach is advantageous in the sense that it uses relative weights instead of precise probabilities which are difficult or time intensive to obtain. Relative weights were used for system components (to show their value for the business) for vulnerabilities (to show how severe they are) and for attacks (to show how likely they are). Upon evaluation, relative weights were shown to be easily understood by the users. Relative weights ranged from one to ten; where one is lowest, and ten is highest. The risk formula only includes three values: Asset value (from 1 to 10, 1-low, 10-high value); severity of vulnerability (from 1 to 10, 1-less severe, 10- most severe); and likelihood of attack (from 1 to 10, 1-least likely, 10-most likely). It was important to consider assets; components; vulnerability; attacks; and countermeasures. Therefore, Sima's risk calculation was modified in order to include a relative countermeasure rating (from 1 to 10, 1-least protective, 10-full protection), to represent relative level of protection provided by implementing a particular countermeasure. Each countermeasure can minimise the risk of exposure to a particular attack. If a countermeasure's weighted value is at its highest (i.e. ten), then the countermeasure will provide full protection against a particular attack.

Sima's (2005) risk calculation formula, along with a modification including a relative countermeasure rating, was also adapted to work with the overall tree-based risk calculations. The formula utilises relationships between the security entities which are stored in the database. A database was designed and built with the full description being provided in Chapter 4, to store the relative ratings and descriptions of each of the five types of security entities (i.e. asset, component, attack, vulnerability and countermeasure). Data on these five types of security entities and their relationships was accumulated from a number of sources (Melbourne & Jorm, 2003; Microsoft Corporation, 2006a; OWASP Foundation, 2005a; Symantec, 2007; Web Application

Security Consortium, 2005).

The aim of the first prototype was to provide essential functionality for the tool to operate. This essential functionality includes the ability to build a tree, modify it, store its data in the database, use existing security information and to modify tree presentation settings. The first prototype's functionality was defined in terms of the following use cases: UC-1 *Start new project* (see Appendix A, from Figure 1 to Figure 4); UC-2 *Open an existing project* (see Appendix A, Figure 5), UC-3 *create a new template*; UC-4 *Open an existing template* (see Appendix A, Figure 8 and Figure 9); UC-6 *Create new settings for graphics*, UC-7 *Edit existing settings for graphics*; UC-8 *View project data as a table* (see Appendix A, from Figure 18 to Figure 23); UC-9 *Edit tree diagram*, UC-10 *Draw tree*; UC-11 *Access security database* (see Appendix A, from Figure 10 to Figure 16); and UC-12 *Save tree as image file* (see Appendix A, Figure 17).

The first round of evaluations was done onsite at a web development company in New Zealand. The main focus was to test all available prototype functionalities, understand how friendly the tools' interface was for the user and obtain suggestions for improvements. Four users were involved in the first round of evaluations. They were:

- A security expert;
- A project manager, who was not a security expert; and
- Two web developers, who were also not security experts.

There was a need for different user groups to participate in the evaluation to ensure that there was a variety of views expressed with respect to the proposed tool. This is due to the tool being targeted to be used by these different groups of people. The evaluation's results became the basis for the development in the next iteration step. The details of these evaluation results are explained in Chapter 5.

2.3.2 Iteration Two – Development and evaluation of revised prototype

Based on evaluation from iteration one, several improvements were made to the design of the user interface. They included providing movable windows (which can be dragged around and resized) for the property panel and for the start panel. A number

of errors were found by the evaluation and fixed in this design iteration. For example, when a user tried to cancel a *delete project* action, the message displayed indicated that the project had been deleted. The capability of producing a graph of a project's history was also added after the first evaluation. This prototype's functionality relates to use case: UC-5 *See project history*.

For the second round of evaluation a large financial corporation in New Zealand was chosen, since hackers frequently target financial company web sites for monetary gain. As recently reported in the 2006 Deloitte Touche Tohmatsu Global Security Survey of top financial institutions, 100% of those surveyed were attacked in 2006. In fact, every financial organisation surveyed in the Asia-Pacific region had been attacked (Deloitte Touche Tohmatsu, 2006). The other reason for carrying out an evaluation in this particular company was that large financial corporations (over 6000 employees), have a greater need to take care regarding security policies. The participating organisation also has a security specialist team, which is responsible for establishing policies and security practices. Financial companies make security for their applications a high priority.

The second evaluation was organised as a meeting with a developer. The main focus was to test how easy it was for a developer to become familiar with using the tool, to understand how friendly the tool's interface was for the user, and to obtain suggestions for improvements, as well as additional functionalities. The meeting included giving explanations of the purpose of the tool and the theory of risk calculation. The researcher also used the tool to demonstrate how a risk assessment could be carried out during this meeting. Suggestions for improvements were obtained. The results of this evaluation (results are detailed in Chapter 5) provided the foundation for the next iteration.

2.3.3 Iteration Three – Development and evaluation of next revised prototype

A suggestion for the previous evaluation was to provide developers with a list of threats/attacks and countermeasures, showing which countermeasures provide the best benefit and were the easiest to implement. This additional functionality was implemented, it relates to use case: UC-14 *See a list of countermeasures*, (see Figure

2.3). The functionality of producing a component report sorted by risk was also added after the second evaluation. This prototype's functionality relates to use case: UC-13 *Produce a component report* (see Appendix A, Figure 7).

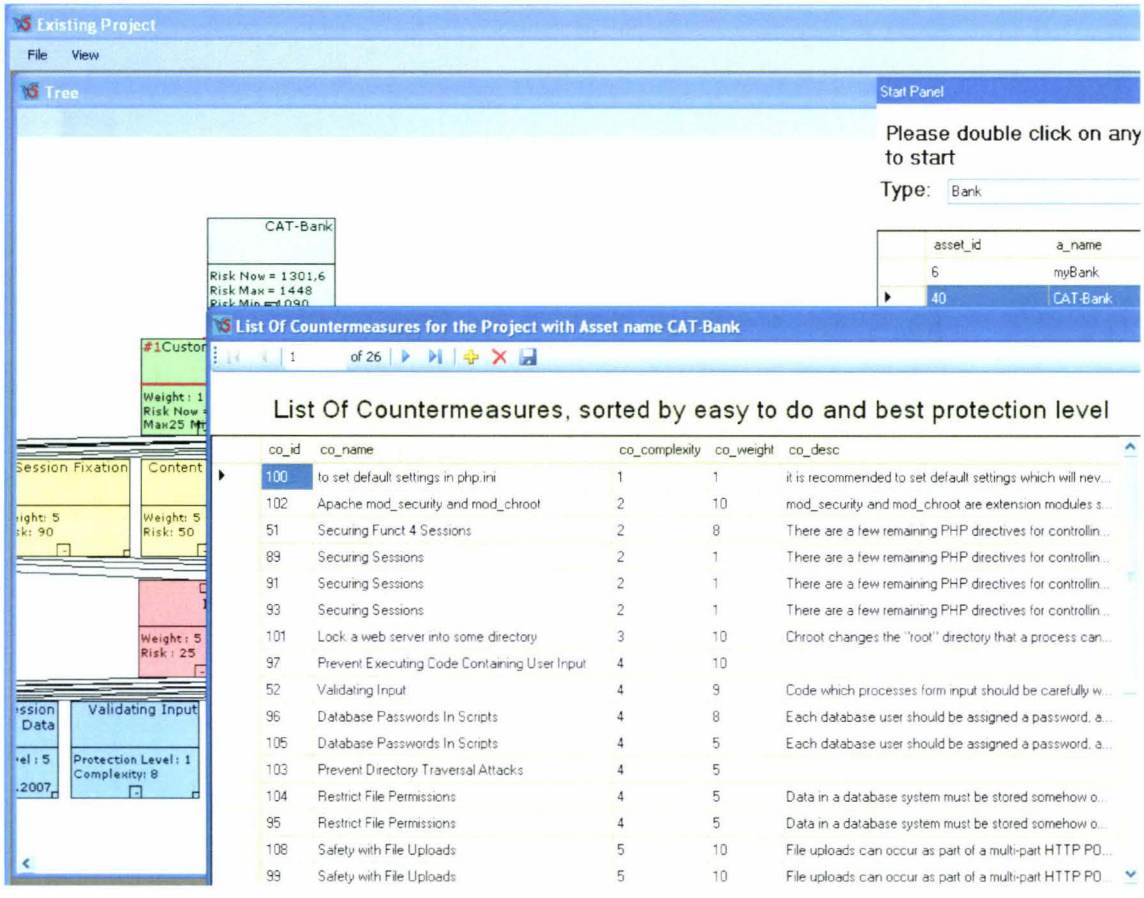


Figure 2.3 Screen shot of a window with a list of countermeasures for an existing project.

The third round of evaluations, which took place in the same company as the second evaluation, was organised as a group meeting with a security team consisting of six individuals. The focus of the evaluation was to test how understandable the proposed risk calculation formula (based on the use of relative ratings as weights) is for security specialists. This meeting included a PowerPoint presentation by the researcher, with an explanation of the research purpose and the risk calculation, with the prototype again being used to demonstrate the tool. At the conclusion of the session the participants completed a questionnaire which included both open, and closed, questions. Suggestions were also obtained as to what additional improvements were needed. They included adding more report functionality and a help system. As a result

of this round of evaluations, a simple help system was developed and implemented. The suggested development of a customised report function was not implemented in this study but will be discussed in terms of future development.

The results of the user satisfaction survey, along with practical time limits associated with a Masters thesis, were used as the criteria to determine the endpoint of the project. The end result of the final evaluation iteration was a working prototype of a computer-based decision support tool for conducting risk assessments in the context of web application security. The final prototype is described in Chapter 4. This artefact was designed to deliver an expected set of features and behaviours, documented as the functional and non-functional requirements in Chapter 3.

2.4 Final Descriptive Evaluation

After completing the third iteration of development and the corresponding field evaluations, the results of the final evaluation of the prototype were reviewed by the researcher and an assessment of how well the tool met each of the requirements was produced. In addition, comparisons were made with two existing tools: Microsoft's Threat Analysis & Modelling Tool (Microsoft Corporation, 2006a) and PTA Technologies' Practical Threat Analysis (PTA) tool (PTA Technologies, 2006). The comparison focused on similarities and differences in the functionality of these tools. The details of this comparison are discussed in Chapter 6.

Chapter 3: Problem Specification and Requirements

The main objective of the research was to produce a decision support tool for security risk assessment of web applications by non-security experts, such as developers and managers. The tool is meant to be used to support security considerations at the requirements, and early design, stages. This objective can be broken down into the following goals:

1. Develop a database of web application security information, to make this information more accessible to developers;
2. Integrate a security risk assessment process into the requirements, and early design, stages of the software development process; and
3. Provide a tool to support both of the above processes.

The focus of this chapter is on the specific requirements which the artefact must meet in order to achieve the research goals. Therefore, this chapter begins with a discussion of functional requirements represented as use cases which the artefact must meet in order to achieve the research goals. This is followed by a discussion of the non-functional requirements, data requirements and risk calculation along with its input requirements.

3.1 Requirements

The requirements specify what the artefact will be able to do and how it is supposed to work. There are two types of requirements: functional; and non-functional. Functional requirements define; what the software is capable of doing, what is included in the calculations, the technical details, data manipulation, and other application specific functionalities. The functional requirements state what functions the system will have, but not how it will operate. The functional requirements are supported by non-functional requirements, which are also referred to as quality requirements. Non-functional requirements provide constraints on the design and/or implementation. These constraints can be; performance requirements, security requirements, quality standards, usability criteria, or other special design considerations (Leffingwell & Wigrig, 2003; Sommerville, 2007).

Clearly stating the functional and non-functional requirements to be met by the artefact is a key step in the Design Science research process. In this research, the functional requirements describe what the targeted users can do with the tool in order to achieve specific goals in the context of risk assessment for web application security. Non-functional requirements specify criteria that can be used to judge the quality of the tool in terms of the services it delivers. Functional requirements can be expressed through the use of differing methods (Sommerville, 2007), as described below:

- Viewpoint (orientated requirements method) provides the framework of discovering conflicts in the requirements proposed by different stakeholders and helps to recognise multiple viewpoints;
- Interviewing, whereby requirements are derived from the formal, or informal, interview process;
- Scenarios (understandable descriptions of real-life examples). This is very useful in gathering details in order to formulate actual system requirements; and
- Use cases (scenario based technique) help to identify the actors involved and the different possible interactions to be represented in the system.

In this study, the functional requirements have been documented as use cases. The next section documents the functional requirements which the tool must satisfy in order to meet the research goals discussed earlier. The following sections present the functional requirements as use cases, the input requirements for the risk calculation, the data requirements, and the non-functional requirements for the tool.

3.2 Functional Requirements for a New Tool represented as Use Cases

Use cases are a technique used to document the functional requirements of systems. Use cases provide a story about what a system should be able to do for its users and a description of the sequence of steps which need to be taken in order to achieve the desired result. Each use case provides one, or more, scenarios that are described in simple terms of how the system should communicate with the different types of users

(represented as actors) and vice-versa, in order to achieve a specific business goal. Use case actors represent either a role played by different types of users, or external software systems with which the system under design will communicate (Leffingwell & Wigrig, 2003; Sommerville, 2007). The proposed tool is called Vi-Secanto (VIsual –SECurity ANalisys TOol). This tool aims to be useful to three groups of users; namely, security specialists, managers, and developers. These groups of users correspond to user roles represented as actors named Manager, Security Specialist, and Developer. Actors with these names appear in the fourteen use cases described in the sections below. Each use case has a name, a description, a unique number, a priority level, one or more primary actors and a flow of events. The priority helps to define the order in which the use cases should be developed. More important use cases should be developed first. Some of the use cases have pre-conditions, post – conditions and extensions. A pre-condition is a state that needs to exist before a use case can start. A post-condition is a state of the system that needs to exist after finishing a use case (Leffingwell & Wigrig, 2003). A use case extension represents calls to do alternative steps based on some condition that occurs; it is a point to plug additional functionality. Each use case is described in a table. The use cases presented in the next section use the following terms:

- Project – Stored information on an asset (each web application is viewed as a project or asset). Each asset has associated components, components have vulnerabilities, vulnerabilities are targeted by attacks, countermeasures can prevent or reduce the loss from particular attacks; and dependencies between all these objects are saved. A project can be saved as a new template for future use.
- Project name – The name for the stored set of information about an asset or project.
- Project type – A way to categorise different kinds of web application projects based on domain information (e.g. E-Commerce, Online bank). Each project type (e.g. Bank) can be associated to many stored projects (e.g. My Bank, CAT-Bank).
- Template - A saved security entity which can be added to an asset (i.e. component, vulnerability, attack or countermeasure) containing predefined information. A security entity can be retrieved from the database and used in an existing or new project. This does not include templates for assets which

are called "Project templates".

- Project template – Stored security information which contains the structure of an asset, including its components, vulnerabilities, attacks, countermeasures and dependencies between all these objects. Project templates can be used as a starting point for new projects and are stored in the same table as projects but with a flag set to indicate whether or not it is a template. Templates and projects have similar structure and content but are accessed differently.
- Template type – A way to categorise different kinds of templates based on domain information (e.g. E-Commerce, Online bank).
- Component template – A template that contains predefined information about a component. It includes all known existing vulnerabilities, attacks and countermeasures for that component.
- Vulnerability template – A template that contains predefined information about a vulnerability and all known existing dependencies related to it, such as attacks and countermeasures.
- Attack template – A template that contains known information about an attack and known existing countermeasures to prevent it or reduce the amount of damage.
- Countermeasure template – A template that contains predefined information about a known existing countermeasure.

3.2.1 Start new project, UC-1

Managers can create a new project and specify the name of the project. Each project is presented as a tree diagram. There are five levels in the tree diagram. They are; asset, components, vulnerability, attacks, and countermeasures. Managers can build a tree diagram by adding security objects such as components, vulnerabilities, attacks and countermeasures at the different levels. The manager then saves the project. Details for the use case (UC-1) are presented in Table 3.1. Screen shots related to this use case are shown as Figures 1, 2, 3 and 4 in Appendix A.

Table 3.1 *Use Case "Start New Project"*

Details	*UC-1, Priority 1*
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager*	

Flow of Events

Main Success Scenario:

1. Manager chooses "Project/New" from Main menu to start a new project.
2. Manager chooses "from Template" to start a new project based on a template; If Manager chooses "without Template " see E1
3. System presents a list of available project types; each type has a list of templates.
4. Manager chooses a project type and then double clicks on the desired template.
5. Manager builds and edits the project as an interactive tree diagram.
6. Manager saves the project.
7. Manager exits the system.

Extensions:

E1 – 1. System presents a list of project types.

2. Manager chooses one project type.

3. System presents an empty form to start building a project without a template then resumes at step 5 of the main scenario.

3.2.2 Open existing project, UC-2

A Manager identifies and chooses the appropriate project based on information presented by a drop-down box. The project then opens. A Manager can make changes by editing the tree diagram. These changes can include adding, or deleting objects (leaves) on the different levels of the tree. It is possible to identify and choose the appropriate security objects (component, vulnerability, attack, and countermeasure) from the database, or to add new objects. A Manager can change the object information from the property window. A Manager can then save the updated project to the database. Details for the use case (UC-2) are presented in Table 3.2. Screen shots related to this use case are shown as Figure 27, Figure 28 and Figure 5 in Appendix A.

Table 3.2 *Use Case "Open Existing Project"*

Details	*UC-2, Priority 1*
Parent: *Vi-Secanto*	
Primary Actors: *Manager*	

Flow of Events
Main Success Scenario:
1. Manager chooses "Project" "Open" from the Main menu to open an existing project (See Figure 27 in Appendix A).
2. System presents to the Manager a list of the available types of projects, each type has a list of existing projects.
3. Manager chooses a project type from a drop down box.
4. Manager chooses one of the existing projects, by double clicking on it in the list associated with the project type chosen in the previous step (See Figure 28 in Appendix A).
5. System opens this project, and presents it as a tree diagram (See Figure 5 in Appendix A). If the Manager edits the tree diagram, then do E1.
6. Manager exits system.
Extensions:
E1 –Run UC-9, Edit Tree Diagram, and then resume at step 5 of the main scenario.

3.2.3 Create a new template, UC-3

A Security Specialist can create a new template and specify the name of the template. A Security Specialist can build a template, graphically presented as a tree diagram, by adding objects to the different levels. A Security Specialist can use the tool to access the database in order to identify and choose the appropriate security objects. A Security Specialist can then save the template to the system. Details for this use case (UC-3) are presented in Table 3.3. Templates can be created for all the types of security entities which are stored in the tool's database. A screen shot related to this use case is shown as Figure 25 in Appendix A.

Table 3.3 *Use Case "Create a New Template"*

Details	*UC-3, Priority 1*
***Parent**: Vi-Secanto*	
***Primary Actors:** Security Specialist*	

Flow of Events
Main Success Scenario:
1. Security Specialist chooses "Template"/"New" from the Main Menu (See Figure 29 in Appendix A).
2. System presents a list of project template types and levels of security objects.
3. Security Specialist chooses a project template type from a drop down box and a template level from a drop down box and presses the "Start" button (See Figure 30 in Appendix A).
4. System presents an empty form to start building a tree for the chosen template of particular type (see Figure 31 in Appendix A).
5. Security Specialist builds and edits a tree diagram by choosing either "Add Object" to create a new object for the tree or "Add from Database" to retrieve a previously created, stored object to be added to the tree.
6. Security Specialist saves the tree diagram as a new template.
7. Security Specialist exits the system.

3.2.4 Open existing template, UC-4

A Security Specialist identifies and chooses an appropriate template from a drop-down box. The chosen template is opened and presented as an interactive tree diagram. A Security Specialist can edit the tree diagram by adding objects on different levels. New objects can be identified and chosen from the database. A Security Specialist can delete or change objects as he or she builds the tree. Finally, changes in the template can be saved to the database. Details for the use case (UC-4) are presented in Table 3.4. Screen shots related to this use case are shown as Figure 8 and Figure 9 in Appendix A.

Table 3.4 *Use Case "Open Existing Template"*

Details	*UC-4, Priority 1*
***Parent**: Vi-Secanto*	
***Primary Actors:** Security Specialist*	

Flow of Events

Main Success Scenario:

1. Security Specialist chooses "Template"/"Open" from the Main menu.
2. System presents a list of project template types, with a list of existing templates for each type.
3. Security Specialist chooses a project template type from a drop down box.
4. Security Specialist chooses one of the existing templates associated with the chosen type, by double clicking on a table (See Figure 8 in Appendix A).
5. System opens the template, and presents it as a tree diagram. If the Security Specialist edits the tree diagram, then do E1.
6. Security Specialist exits the system.

Extensions:

E1 –Run UC-9, Edit Tree Diagram, and then resume at step 6 of the main scenario.

3.2.5 See project history, UC-5

A Manager can identify and choose an appropriate project from a drop-down box. A Manager can see the project's history, presented as a graph. This picture can be saved to a file, or printed. Details for the use case (UC-5) are presented in Table 3.5. A screen shot related to this use case is shown as Figure 5.2 in Chapter 5.

Table 3.5 *Use Case "See Project History"*

Details	*UC-5, Priority 2*
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager*	

Flow of Events
Main Success Scenario:
1. Manager chooses to open the history of an existing project from the Main menu.
2. System presents Manager with the list of existing projects.
3. Manager chooses one of the existing projects.
4. System draws a graphic of the project history.
5. Manager can save the graphic to a file, or print it.
6. Manager exits system.

3.2.6 Create new settings for graphics, UC-6

A Manager can specify and define a new colour scheme for the settings, the connection type between a tree's leaves and the setting's name. The colour scheme defines which colours are used to present the different security objects in the tree diagram, as well as the background colour of the tree diagram. The security objects are assets, components, vulnerabilities, attacks and countermeasures. Different companies may have different preferences for their projects. Details for the use case (UC-6) are presented in Table 3.6. A screen shot related to this use case is shown as Figure 26 in Appendix A.

Table 3.6 *Use Case "Create New Settings for Graphics"*

Details	*UC-6, Priority 1*
Parent: *Vi-Secanto*	
Primary Actors: *Manager*	

Flow of Events
Main Success Scenario:
1. Manager chooses "Settings"/"Graph Property" and then chooses "New" from the Main menu.
2. System opens a new window which shows settings and colour options.
3. Manager chooses a colour for each setting.
4. Manager chooses a connection type between tree leaves.
5. Manager specifies the name of the new colour schema.
6. Manager saves the new colour scheme and exits this use case.

3.2.7 Edit existing settings for graphics, UC-7

A Manager can edit the existing colour scheme for the settings and change the connection type between a tree's leaves and the name of the settings. The colour scheme defines which colours are used to present different security objects in the tree diagram, as well as the background colour of the tree diagram. Security objects are asset, components, vulnerabilities, attacks and countermeasures. Details for the use case (UC-7) are presented in Table 3.7. A screen shot related to this use case is shown as Figure 4.3 in Chapter 4.

Table 3.7 *Use Case "Edit Existing Settings for Graphics"*

Details	*UC-7, Priority 1*
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager*	

Flow of Events

Main Success Scenario:

1. Manager chooses "Settings"/"Graph Property", and then chooses "Edit" from the Main menu.
2. System opens a window with existing settings, which are listed in the dropdown box.
3. Manager selects a colour scheme name from drop down box for editing.
4. System changes the button colours, (each button's colour represents the current settings) and connection type to match the settings represented by the chosen scheme's name.
5. Manager edits the settings he or she wants to change. The colour scheme for one or more levels of the tree (asset colour, component colour, vulnerability colour, attack colour, countermeasure colour), the background colour, connection type between tree leaves or the name of the group of settings can be altered.
6. Manager saves and exits this use case.

3.2.8 View project data as a table, UC-8

It is possible to view the project data as a table. A Manager can identify and choose a project then display the project information as a set of related tables. The Manager opens a project and then views it level by level. The Manager can make changes and save them to the database. Details for the use case (UC-8) are presented in Table 3.8. Figures 18 to 23 in Appendix A show screen shots of this functionality. Tabs are used to navigate to each level within a project (i.e. stored information on a web application asset).

Table 3.8 *Use Case "View Project Data as a Table"*

Details	*UC-8, Priority 1*
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager*	
Flow of Events	
Main Success Scenario: 1. Manager chooses "Project"/ "As Table view" from Main Menu. . 2. System displays a set of five tabs for asset, components, vulnerabilities, attacks, and countermeasures (See Figure 18 in Appendix A). 3. Manager selects a project asset from the displayed table under the Asset tab. 4. System populates the table under components with the components for the selected asset. 5. Manager chooses items from the tables under each tab and double clicks on any line of the table. 6. System updates tables under each tab as items are selected for components, vulnerabilities, attacks, or countermeasures. 7. System brings up an online form where the selected object's properties can be changed (See Figure 23 in Appendix A). 8. Manager makes changes, then saves and exits this use case.	

3.2.9 Edit tree diagram, UC-9

Editing a tree diagram includes adding leaves, deleting leaves, editing leaf details and then saving the changes. A leaf can represent any security object from the diagram, such as an asset, component, vulnerability, attack, or countermeasure. Each of these objects has different property fields. Details for the use case (UC-9) are presented in Table 3.9. Screen shots related to this use case are shown as Figure 5 and Figure 10 in Appendix A.

Table 3.9 *Use Case "Edit Tree Diagram"*

Details	*UC-9, Priority 1*
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager*	
***Preconditions**: Open existing project, or Open existing template*	

Flow of Events

Main Success Scenario:

1. Manager points to one leaf in the tree diagram and does a right mouse click.
2. System presents a menu based on the context (i.e. the leaf) where the click occurred.
3. Manager chooses one or more of the following sub-flows:
   - 3.1. If the Manager chooses to delete the leaf, the program will ask for confirmation and if given it will carry out the deletion.
   - 3.2. If the Manager chooses to add a leaf, the program will add a 'child' leaf to the existing one.
   - 3.3. If the Manager chooses 'property' (or double clicks on the existing leaf), the program will show the property window, wherein the manager can edit the leaf's property.
4. Manager chooses the "save" option from the context menu.
5. System saves the project and exits this use case.

.

3.2.10 Draw tree, UC-10

Every project (or template) can be presented as a tree diagram. The diagram is drawn on the basis of the information stored in the database. Details for the use case (UC-10) are presented in Table 3.10. This use case is included by several of the above use cases which display the project as a tree diagram. Screen shots related to this use case are shown as Figure 5, Figure 9 and Figure 10 in Appendix A.

Table 3.10 *Use Case "Draw Tree"*

Details	*UC-10, Priority 1*
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager*	
***Preconditions**: 1. The database contains at least one stored project or sample template of a project.*	
2. A project type list is visible and a type has been selected.	

Flow of Events

Main Success Scenario:

1. Manager double clicks on an existing project (or template) associated to the project type.
2. System retrieves information for a particular project (or template) from the database.
3. System draws the interactive tree diagram for this asset and each of its components, which includes:
   - 3.1. System retrieves information for vulnerabilities belonging to each component from database and draws it in the form.
   - 3.2. System retrieves information for attacks belonging to each vulnerability from database and draws it in the form.
   - 3.3. System retrieves information for countermeasures belonging to each of these attacks from database and draws it in the form.
4. System does a risk calculation prior to displaying the tree.

3.2.11 Access security database, UC-11

A Manager can retrieve security information from the tool's database. A Manager can specify whether, or not, he/she needs the object's information to be added with all other dependent objects, or just one particular object (e.g., component, vulnerability, attack, or countermeasure). Details for the use case (UC-11) are presented in Table 3.11. Figures 10 to 16 in Appendix A show examples of screen shots corresponding to this use case.

Table 3.11 *Use Case "Access Security Database"*

Details	***UC-11, Priority 1***
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager, Security Specialist, Developer*	
***Preconditions**: 1. The database contains at least one stored project or sample template of a project.* *2. A tree diagram is visible.*	

Flow of Events

Main Success Scenario:

1. Manager right clicks on a leaf node at any level in the tree. The leaf can be an asset, or component, or vulnerability or attack, or countermeasure.
2. System presents a context menu based on the type of node clicked on in the previous step.
3. Manager chooses "Add from database".
4. System retrieves information about the selected security object and displays it as a list.
5. Manager chooses the desired security database object from the list by double-clicking.
6. System retrieves the object from the database and draws the new object as a new leaf on the tree. If Manager clicks the Radio Button "With Dependencies" do E1.

Extensions:

E1 – System draws the new object as a new leaf node on the tree with all its dependent child leaves.

3.2.12 Save tree as image file, UC-12

The "Save tree as image" action includes saving a graphical view of a project as an image file, in one of the following formats: JPeg, Bitmap, or Gif. Details for the use case (UC-12) are presented in Table 3.12. Figure 17 in Appendix A shows a screen shot of this functionality.

Table 3.12 *Use Case "Save Tree as Image File"*

Details	*UC-12, Priority 1*
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager*	
***Preconditions**: A project tree diagram should be open.*	

Flow of Events
Main Success Scenario:
1. Manager clicks "Save Tree as Image File" from menu.
2. System displays the image file.
3. Manager clicks the button "save to file".
4. System displays a save dialog window.
5. Manager browses to a location and enters a file name using this dialog window.
6. System saves the file to the specified location.

3.2.13 Produce a Component report, UC-13

The "Produce a Component Report" is a report, what shows list of project component sorted by most current risk in the descendant order. Details for the use case (UC-13) are presented in Table 3.13. A screen shot related to this use case is shown as Figure 7 in Appendix A.

Table 3.13 *Use Case "Produce a Component Report"*

Details	*UC-13, Priority 2*
***Parent**: Vi-Secanto*	
***Primary Actors:** Manager*	
***Preconditions**: The database contains data for at least one component.*	

Flow of Events

Main Success Scenario:

1. Manager chooses "Reports"/"Component Report" from the Main menu.
2. System displays a window with a list of project types.
3. Manager chooses a project type.
4. System shows a list of projects which are of the chosen project type.
5. Manager chooses a project by clicking on it.
6. System displays a component report in a new window.
7. Manager chooses one or more of the following sub-flows:
   - 7.1. Save. Manager presses the "Save" button to save the report. Manager chooses one of two options: save as PDF format or Excel. Manager specifies name of the file. System saves the file to the specified location.
   - 7.2. Print. Manager presses the "Print" button. System prints the report.

3.2.14 See a List of Countermeasures, UC-14

The "*List of countermeasures*" is a report that shows a list of project countermeasures sorted by ease of implementation and best protection level. Details for the use case (UC-14) are presented in Table 3.14. Screen shots related to this use case are shown as Figure 32 and Figure 33 in Appendix A.

Table 3.14 *Use Case "See a List of Countermeasures"*

Details	*UC-14, Priority 2*
Parent: *Vi-Secamo*	
Primary Actors: *Manager*	
Preconditions:	
1. The database contains data for at least one project; this project has at least one countermeasure.	
2. Project is open, a tree diagram is visible.	

Flow of Events

Main Success Scenario:

1. Manager chooses "List Countermeasures" from the context menu.
2. System displays a window with a list of countermeasures, sorted by easy to do and protection level.
3. Manager double clicks on a countermeasure from the list to view details on how to implement it.

3.3 Non-functional requirements

The following sections discuss non-functional requirements, criteria for judging the operation of a system. The International standard ISO 9126, for the evaluation of software, states the different specific characteristics and sub-characteristics of non-functional requirements (see Table 3.15) in ISO 9126-1, the first part of the standard (International Organisation for Standardization, 2001). The ISO 9126 standard combines functional and non-functional requirements. The functional requirements for the tool were already discussed in Section 3.2 as use cases, so only the non-functional requirements are discussed here.

Table 3.15 *Specific Requirements' Characteristics ISO 9126-1*

Specific requirements' characteristics	**Description**	**Sub-characteristics**
Functionality	A set of attributes that bear on the existence of a set of functions (i.e. application specific behaviours) and their specified properties. The functions are those that satisfy stated, or implied, needs.	Suitability Accuracy Interoperability Compliance Security
Reliability	A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.	Maturity Recoverability Fault Tolerance
Usability	A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated, or implied, set of users.	Learnability Understandability Operability
Efficiency	A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.	Time Behaviour Resource Behaviour
Maintainability	A set of attributes that bear on the effort needed to make specified modifications.	Stability Analysability Changeability Testability
Portability	A set of attributes that bear on the ability of software to be transferred from one environment to another	Installability Replaceability Adaptability

The aim of the research is to provide a prototype tool. This is not a final software product, which is why only some of the above requirements' characteristics were considered. They are: Functionality (with sub-characteristic Compliance); and Usability (with sub-characteristics Learnability and Understandability). Each sub-characteristic can be divided into attributes. Each attribute can be measured, or verified, in a certain manner through testing of the software product.

Compliance is related to the requirements that the product needs to comply with such as existing business procedures, methods, or practices. The requirement for the proposed tool to support an existing security assessment method is discussed in Section 3.3.2. Usability requirements are related to the design of the Graphical User Interface and user interaction with the software. The usability of the software product

can be enhanced by providing a user manual, informative error messages and a help system. The Graphical User Interface and Usability requirements are discussed in the following section.

3.3.1 Usability Requirements.

Usability requirements have sub-characteristics: Learnability and Understandability. Understandability is about how quickly a user becomes familiar with software. Learnability reflects how long it would take for a user to complete tasks using the software. Measures of these attributes would be *time to learn a specific task* or *time to learn how to use a software tool.* Learnability can be considered as part of Understandability (Urrego-Giraldo, 2004); if a user has a good understanding of new software, learning becomes easy.

Usability requirements are based on the Human Computer Interaction (HCI) principles. Existing methods to measure software usability are based on these principles. Instruments for measuring software usability include the Purdue Usability Testing Questionnaire (PUTQ) and the Questionnaire for User Interaction Satisfaction (QUIS) (Lin, Choong, & Salvendy, 1997). The PUTQ instrument is based on the following HCI principles (Lin et al., 1997).

1. Compatibility - This is ensuring that the results of any control entry will be consistent with user expectations.
2. Consistency - This refers to having local coherence in the sense of using the same style for common interface items on different screens; for example, the colour schema of interface items should follow conventional colour perceptions.
3. Flexibility - Means allowing for customizing of windows and outputs (attributes such as: font, size of the window, foreground/background colour).
4. Learnability - This means ensuring logical organisation of menus and sequencing of steps to complete a task.
5. Minimal action – This refers to how easy it is to enter data (use of default data and choice lists).
6. Minimal memory load - If possible, do not use abbreviations and acronyms

at all.

7. Perceptual limitation – This means the user interface should provide easily distinguished colours; no more than four should be used simultaneously.
8. User guidance – This means providing system feedback in the form of helpful error messages (Lin et al., 1997).

The software development should follow these HCI principles with the aim to make software user friendly. The prototype's aim is to make security information more accessible and more understandable. Security knowledge should be presented in a form which is easily consumed and understood by the user. The interface for the tool should provide visualisations of the relationships between key factors, such as asset vulnerabilities, attacks and countermeasures, in a risk assessment. These relationships should be presented as tree diagrams, since tree diagrams are easily understood and are commonly used in IT (Burns, 2005; Gegick & Williams, 2005; Howard, 2004; Sahinoglu, 2005; Salter, Saydjari, Schneier, & Wallner, 1998; Schneier, 1999; Thompson, 2005). IT has a history of using tree diagrams to show hierarchical relationships, such as in fault trees, event trees and attack trees. The use of visualisation in the form of an interactive tree diagram quickly highlights relationships between vulnerabilities, attacks and countermeasures related to each component of a web application.

For example, an attack tree structure provides a formal description of how an attack on a system can be performed. The root node represents the goal of the attack, while branches represent different ways of achieving that goal, from a high level to more specific levels, which are the leaf nodes of the tree (Balzarotti, Monga, & Sicari, 2005). The literature describes attack trees as a systematic way to assess security risks and identify countermeasures (Moore, Ellison, & Linger, 2001; Schneier, 1999). The attack information is presented as an upside down tree, whereby each branch indicates a way an attacker can compromise the system security, with respect to a particular goal associated with a particular asset or resource (e.g. intercepting a network connection for a particular user). If the event occurs, then the system security will be compromised (Moore et al., 2001). The base requirements for the Graphical User Interface of the proposed software tool are defined in Table 3.16.

Table 3.16 *Requirements for the Graphical User Interface*

Name	Requirement description
General	The MS Windows' user interface guidelines are followed, since the tool is designed to run on the Windows operating system and uses C# classes based on these guidelines. Placement of menus, names and icons should follow mainstream Microsoft application guidelines so that users will be intuitively familiar with the interface, through past experience with Microsoft applications, or applications designed for the Windows platform.
Navigation	Top menu (i.e. Main Menu) Context menus
Project tree	Any project or template should be visually presented as an interactive tree with each asset associated with vulnerabilities, attacks and countermeasures. Projects (or templates) may be edited and saved for later use. An interactive tree may be saved as a graph for later printouts. A tree should allow for opening and collapsing branches in order to facilitate the presentation of all, or just parts, of the tree.

3.3.2 Compliance with a AS/NZS 4360 Risk Management Standard

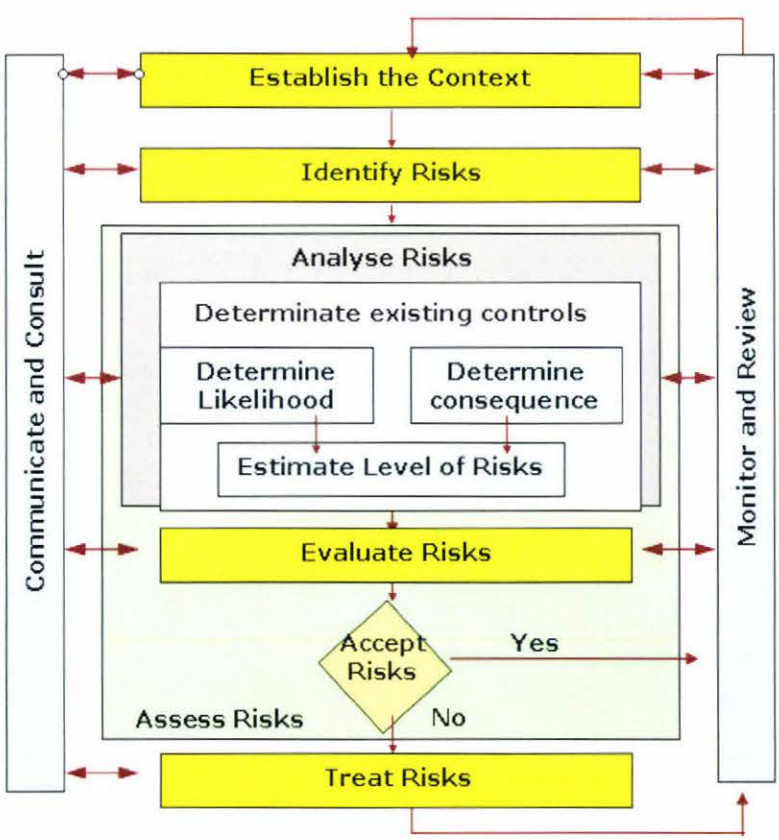


Figure 3.1 Risk management process

Source: (Standards Australia and New Zealand, 1999)

The tool should provide support for a business to follow an existing security standard such as AS/NZS 4360, a standard which is becoming a common requirement for all

businesses. The risk management process defined in AS/NZS 4360 (see Figure 3.1) contains a number of steps, which are described in Table 3.17.

Table 3.17 *Steps for the Risk Management Process*

AS/NZS 4360 steps	**Short description**	**Requirements supported by Vi-Secanto**
1. Establish the context	Includes understanding the system in different contexts. Some sub-steps need to be considered: • Strategic context • Organisation context • Risk management context • Develop criteria • Decide the structure	1) Support the input, storage, retrieval and display of information on system assets.
2. Identify risks	This step requires a systematic approach for the identification of: • What can happen? • How it can happen?	2) Support the input, storage, retrieval and display of information on vulnerabilities for components of the system assets. 3) Support the input, storage, retrieval and display of information on system attack linked to existing vulnerability.
3. Analyse risks	Includes estimated levels of risk for the likelihood of each event and the severity of its consequences. Determination of existing controls is taken into account when estimating the risk.	4) The Tool should do a risk calculation for system components. 5) The Tool should do a current risk calculation. 6) The Tool should do maximum and minimum risk calculation. The Tool should show which countermeasures need to be implemented for maximum protection.
4. Evaluate risks	Includes comparison with the levels of risk estimated in the previous step, with the criteria established at the beginning. The priorities need to be set for the management of the risks.	7) The Tool should visually show the most critical components of the system.
5. Accept risks - Decision step	Two directions: accept a risk or don't accept a risk. In the case of non-acceptance it needs to be treated as in step 6.	8) The Tool should visually present a tree to make it easier for management to make a decision.
6. Treat risks	Decisions have to be made about risk treatment options. These decisions are based on the following steps: • Identify treatment options • Evaluate treatment	9) Support the input, storage, retrieval and display of information on protection (countermeasures). 10) The Tool should re-calculate risk for system components when protections options are changed. 11) The Tool should provide a list of system components prioritised by risk.

AS/NZS 4360 steps	Short description	Requirements supported by Vi-Secanto
	options • Select treatment options • Prepare treatment plan • Implement plans	12) The Tool should provide a list of countermeasures prioritised by better protection and ease of implementation. A description of how to implement it and/or an example of implementation should be included.
7. Monitor and review process	Continual during all steps and over time, because the risks to a system can change: risks need to be re-analysed. Modifications and changes need to be monitored and taken into account.	13) The Tool should support editing for any security objects, with a risk re-calculation for the system. 14) The Tool should support recording, storage and display of historical data for on risk changes over time.
8. Communicate and consult process	Going through all these steps as well, to insure consultation with all involved parties: internal and external to the organisation.	15) The Tool should assist all involved parties in taking decisions by visually presenting a tree diagram; it should allow for trying out different scenarios.

The risk management process has a number of steps and sub-processes which are useful in the context of web applications. The first step is *Establish the context. This step* includes making decisions about web application structure and defining which information to record about assets. It sets the criteria to be measured and monitored. The second step is *Identify the risks*, a step that involves understanding what can happen with each component and the overall system if an attack was to occur. This step also includes the assessment of all vulnerabilities that may lead to an attack, an undesirable event. The risk management process, the third step, includes the risk assessment process: this is shown in green in the middle of Figure 3.1. The risk assessment process is held together with three steps from Risk management. The *Analyse risk* step aims to estimate the level of risk through the determination of its potential likelihood and consequences. The risk calculation is part of the risk assessment process: the risk calculation can be done differently (for more discussion on different risk calculation approaches please refer to Chapter 6). Each organisation can have their own criteria for the level of risk acceptance in the *Risk Evaluation* step (step four). Risk priorities can also depend on the different requirements of organisations. The Accept risks step is a decision step (step five). If risk needs to be minimised, the next step: *Treat Risk*, proceeds. Along with all the steps of risk management, two processes carried out are: *Communicate and Consult*, and *Monitor*

and Review.

The description of the steps in the risk management process becomes a source for the tool requirements, which are summarised in the last column of Table 3.17. Requirements numbers 1, 2, 3 and 9 in Table 3.17 relate to the data requirements which are discussed in the data requirements section in this Chapter. Requirements numbers 4, 5, 6 and 10 stated in Table 3.17 relate to risk calculations; the algorithm for the risk calculation is provided in the last section of this chapter. Requirement numbers 7, 8 and 15 relate to the use case UC-10, *Draw tree*. Requirement number 13 in Table 3.17 relates to the use case UC-9, *Edit Tree Diagram.* Requirement number 11 in Table 3.17 relates to the use case UC-13, *Component report.* Requirement number 12 in Table 3.17 relates to the use case UC-14, *Component report.* Requirement number 14 in Table 3.17 relates to the use case UC-5, *See Project History.*

3.3.3 Compliance with a known security assessment method

The tool should provide support for the business to follow an existing security practice, or method, such as the OCTAVE method for risk assessment and management. The OCTAVE method is a process that consists of guidelines for examining organisational and technological issues, and puts together a comprehensive picture of the information security needs of an organisation. The OCTAVE process has three major phases (see Table 3.18).

OCTAVE phase one involves conducting an organisational evaluation to enable the building of asset-based threat profiles. The process of building these profiles includes the examination and identification of important information assets, and defining the threats to each asset. Protection strategy practices are also defined. This includes the security requirements of the assets, what the organisation is currently doing for information protection, and weaknesses, or vulnerabilities, in the organisational practice. Phase two involves the identification of infrastructure vulnerabilities. The evaluation of the information infrastructure includes the examination of all vulnerabilities that might lead to an undesirable action. Finally, phase three develops a

security strategy and plan for asset protection. Risk calculation is also performed in phase three. The information gathered in the two previous phases is analysed in order to identify risks to the organisation and to evaluate the risks based on their impact to the business. As a result, the mitigation plans for the organisation are developed. A list of prioritised risks to the organisation is included in the protection strategy (Alberts & Dorofee, 2001).

Table 3.18 *Requirements for Compliance of the Proposed Tool with OCTAVE*

OCTAVE phase	**Short description**	**Requirements for tool support**
Phase 1	Examination and identification of important information assets.	1) Support the input, storage, retrieval and display of information on system assets.
Phase 2	Define vulnerabilities for each asset. Examine all vulnerabilities that might lead to an attack.	2) Support the input, storage, retrieval and display of information on vulnerabilities for components of the system assets. 3) Support the input, storage, retrieval and display of information on system attack linked to existing vulnerability.
Phase 3	Define protection. Do a risk calculation. Develop a list of prioritised risks.	4) Support the input, storage, retrieval and display of information on protection (countermeasures). 5) Tool should do a risk calculation for system components. 6) Tool should provide a list of system components prioritised by risk.

On the basis of the description of the OCTAVE phases, the requirements for the tool have been summarised in Table 3.18. Requirement number five stated in Table 3.18 relates to risk calculation; the algorithm for the risk calculation is provided in the last section of this chapter. Requirement number six in Table 3.18 relates to the use case UC-13. The remaining requirements (numbers 1, 2, 3 and 4) in Table 3.18 relate to the data requirements which are discussed in the next section.

3.4 Data requirements

The data requirements, in terms of what needs to be stored about each aspect of the security knowledge held by the tool, are presented in Table 3.19.

Table 3.19 *Data Requirements for the Proposed Tool*

Table	**Field (data type)**	**Foreign key**	**Description**

Table	Field (data type)	Foreign key	Description
asset	Id (auto number) Name (text) Description (text) Risk (number)		Keep information about asset – web application
component	Id (auto number) Name (text) Description (text) Weight (number) Risk (number)	Asset_id	Keep information about system components (components of web application)
vulnerability	Id (auto number) Name (text) Description (text) Weight (number)	Component_id	Keep information about vulnerabilities
attack	Id (auto number) Name (text) Description (text) Weight (number) Type_attack_id (number)	Vulnerability_id Type_attack_id	Keep information about attacks
countermeasure	Id (auto number) Name (text) Description (text) Weight (number) Implemented(yes/no) Date_Implementation Complexity (number) Platform_id (number)	Attack_id (attack which can be prevented by this countermeasure) Vulnerability_id (vulnerability which can be prevented by this countermeasure) Platform_id	Keep information about countermeasures
attack-type	Id Name Description		Keep information about different types of attack
asset-type	Id Name Description		Keep information about different types of asset
vulnerability-type	Id Name Description		Keep information about different types of vulnerabilities
platform	Id Name Description		Keep information about different types of platform
known- component	Id (auto number) Name (text) Description (text) Weight (number) Risk (number)		Keep information about known components (components of web application)
known-vulnerability	Id (auto number) Name (text) Description (text) Weight (number)		Keep information about known vulnerabilities
component-vulnerability		Component_id Vulnerability_id	Keep information about which vulnerability belongs to which component
known - attack	Id (auto number) Name (text) Description (text) Weight (number)	Type_attack_id	Keep information about known attacks

Table	Field (data type)	Foreign key	Description
	Type attack id (number)		
vulnerability-attack		Vulnerability_id Attack id	Keep information about which attack uses which vulnerability
known - countermeasure	Id (auto number) Name (text) Description (text) Weight (number) Implemented(yes no) Date_Implementation Complexity (number) Platform id (number)	Platform id	Keep information about known countermeasures
vulnerability-attack-countermeasure		Vulnerability id Attack id Countermeasure id	Keep information about which countermeasure can prevent vulnerability, attack

3.5 Risk calculation.

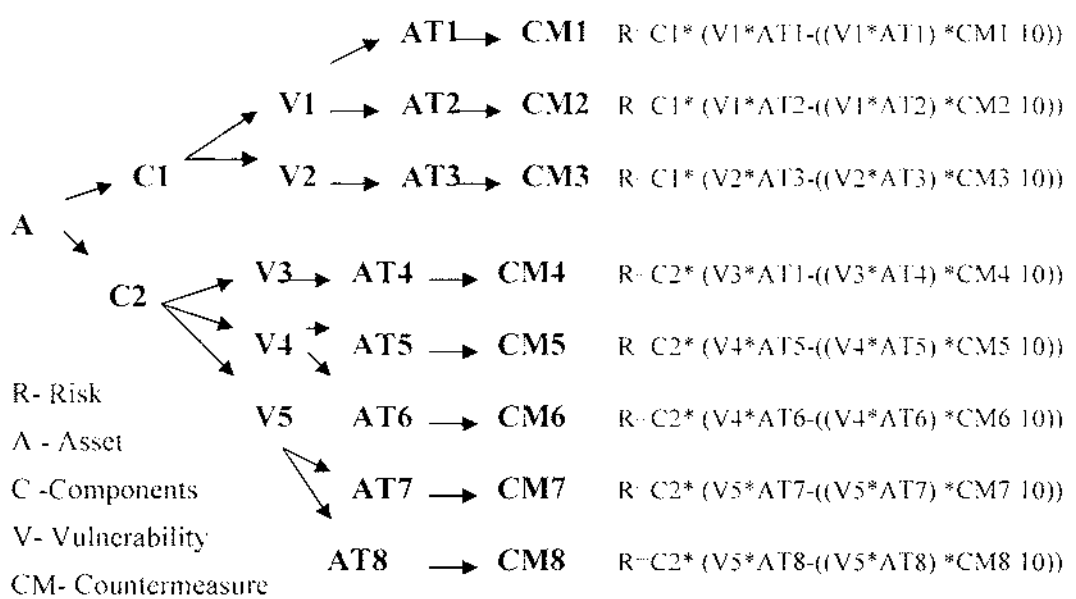


Figure 3.2 Security Risk Calculation tree diagram.

Section 3.3.2 of this Chapter stated a requirement: The tool should do risk calculations for the system components. This section describes the Risk calculation formula. The tool calculates risk using an extension of Sima's (2005) formula, shown as Equation (1). Each of the three components of risk in Equation (1) is rated on a scale of 1 to 10 (10 being the most severe or highest value, severity or likelihood).

Risk associated with each asset is calculated on the basis of a decision-tree, illustrated in Figure 3.2 as a summary of risk for all components.

Risk = C x V x AT (1)

C is a Component value,

V is a Vulnerability severity

AT is Likelihood of an Attack

Table 3.20 *Symbols used in the Risk Calculation Formula*

Symbol used in formulas	**Definition and range of values**
C_i	The value of system component i, where i =1,...n. (n is the number of components in the system); $1 <= C_i <= 10$
V_{ij}	The severity of a vulnerability j, where j =1,...m. (m is the number of vulnerabilities for a particular component i) $1 <= V_{ij} <= 10$
AT_{ijk}	The likelihood of an attack k, where k =1,...p. (p is the number of attacks, which use vulnerability j, which exists in component i) $1 <= AT_{ijk} <= 10$
CM_{ijk}	The average of the countermeasures implemented; where $CM_{ijk} = \left(\sum_{h=1}^{r} CM_{ijkh} \right) / r$, where h=1,...r (r is the number of countermeasures, which can protect from attack k, on vulnerability j, in component i)
Min_Risk(C_i)	Minimum Risk (Residual) for component C_i is risk after all possible countermeasures are implemented for this component.
Min_Risk(A)	Minimum Risk (Residual) for asset A is risk after all possible countermeasures are implemented for the components of this asset. A user project is also called an asset, so one asset per project.
Max_Risk(C_i)	Maximum Risk for a component C_i is the risk when no countermeasures are implemented for this component.
Max_Risk(A)	Maximum Risk for asset A is the risk when no countermeasures are implemented for the components of this asset. Only one asset for each user project.

The system calculates Maximum Risk (Equation (2)) with the assumption that none of the countermeasures were implemented. The symbols used in and the equations are described in Table 3.20. The relative ratings specified by security experts as weights

for vulnerabilities (V_{ij}), attacks (AT_{ijk}) and components (C_i) as inputs are used.

$$\textbf{Max\_Risk}\ (C_i) = \sum_{k=1}^{p} \sum_{j=1}^{m} (\ C_i * (\ V_{ij} * \ AT_{ijk})) \tag{2}$$

$$\textbf{Max\_Risk}\ (A) = \sum_{i=1}^{n} \ (\textbf{Max\_Risk}(C_i)) \tag{3}$$

The system calculates Minimum Risk from the assumption that all possible countermeasures (CM_{ijk}) are implemented. The relative ratings: as weights for vulnerability (V_{ij}), attacks (AT_{ijk}), components (C_i), and countermeasures (CM_{ijk}): are taken as the inputs.

$$\textbf{Min\_Risk}\ (C_i) = \sum_{k=1}^{p} \sum_{j=1}^{m} (C_i * (\ V_{ij} * \ AT_{ijk} - (\ V_{ij} * \ AT_{ijk} * \frac{CM_{ijk}}{10}\ ))) \tag{4}$$

$$\textbf{Min\_Risk}\ (A) = \sum_{i=1}^{n} \ (\textbf{Min\_Risk}(C_i)) \tag{5}$$

The Current Risk for the system is calculated from the knowledge that some countermeasures are implemented and some are not. The Current Risk value will always be greater, or equal to, the Minimum Risk and less than, or equal to, the Maximum Risk.

The Security Risk Calculation uses a hierarchical tree, as shown in Figure 3.2, to represent the residual risk calculation for each component of a web application asset. Each path represents the joint occurrence of the vulnerability, an attack and a level of use, or non-use, of countermeasures. The residual risk for each path is calculated as the product of these weighted inputs and then summed over all branches to arrive at the total residual risk for each component. Every event, or object, that is able to reduce the risk to the system can be defined as a countermeasure (CM). For example, an action, a device, or a procedure, can be represented as a CM. When the CM is applied, the risk is reduced and the remaining risk is defined as residual risk. If a countermeasure can reduce risk completely, the value of the residual risk will be zero. New components can be easily added to the system, modelled as shown in Figure 3.2,

and the overall system risk recalculated.

The next chapter discusses the high level architecture and detailed design of the solution that evolved as a way to implement the requirements detailed in this chapter.

Chapter 4: Architecture and Detailed Design of Vi-Secanto (Visual Security Analysis Tool)

This chapter focuses on the architecture and detailed design of the proposed tool, Vi-Secanto. The tool has the aim of providing support for the integration of a security risk assessment process into the requirements and design stages of the software development process and to help make existing security information more accessible to developers, who are not security experts, as described in Chapter 3. The layers of the tool's architecture and detailed design of the classes of objects making up each layer of the tool's architecture are discussed in this chapter. The chapter concludes with a description of how to work with the proposed tool.

4.1 Architecture of Vi-Secanto

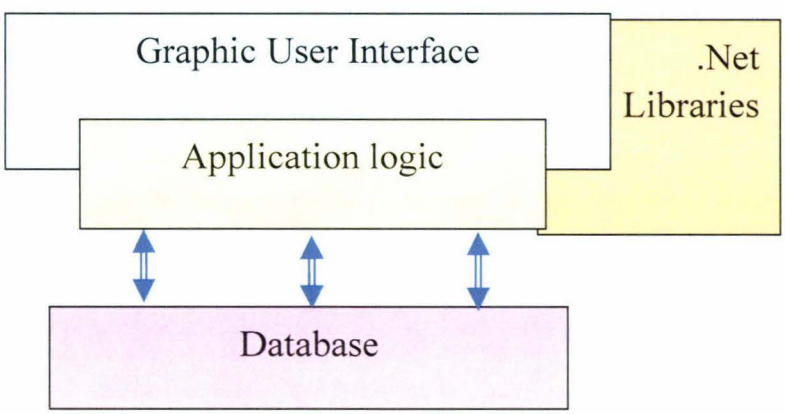


Figure 4.1 Vi-Secanto's architecture

Installing the prototype version of the Vi-Secanto tool requires the files shown in Figure 4.2. The Vi-Secanto tool serves as an integrated environment for security information presentation and the manipulation of security data for particular projects, including the presentation of a tree view and the risk calculation. The architecture of the tool can be divided into the layers shown in Figure 4.1 as follows:

1. The Graphical User Interface (GUI) layer (i.e. classes that produce input forms, output diagrams);
2. The application layer (i.e. classes that implement the application logic); and
3. The .Net libraries used as a basis for the GUI and the application layer;

4. The storage layer (i.e. the MS Access database and classes to interface with the application layer).

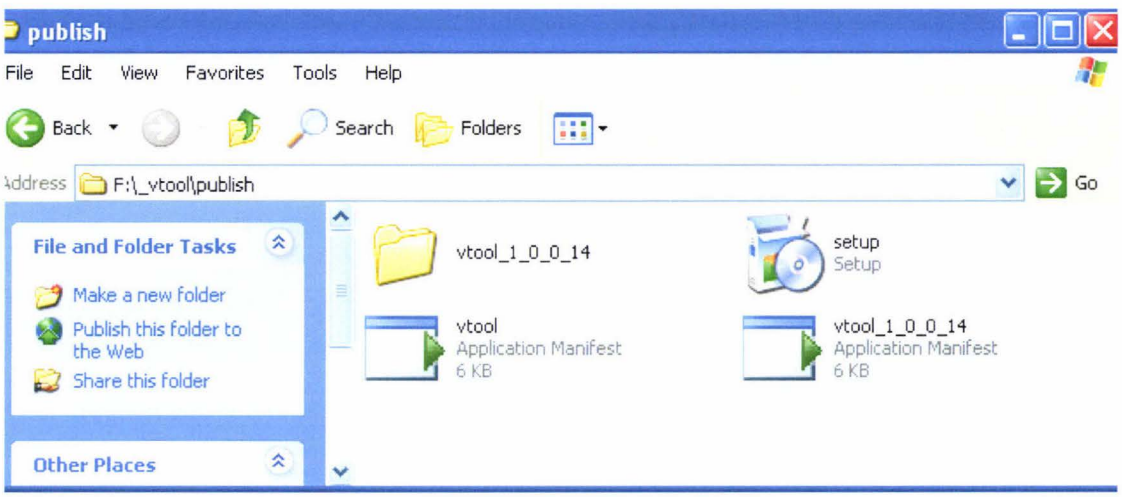


Figure 4.2 Folder of files used to install the Vi-Secanto Tool prototype.

Vi-Secanto was written in Microsoft Visual C#, using Microsoft Visual Studio 2005 .Net. For storage, the Vi-Secanto tool utilises a Microsoft Access 2002 database. The database contains nineteen tables. Logically the database can be divided into two parts: structure and content. The structure of a table represents a particular type of security object (e.g. an asset) and its attributes (e.g. asset name, asset description). The structure of a table also includes a primary key attribute which can be duplicated as a foreign key attribute in a related table to show a relationship. The content of the tables represents stored data about particular security objects and particular relationships between those objects.

The database tables can be classified into two categories, project data and template data. Pre-defined security knowledge can be conceptualised by users as templates which can be saved under specific project names. Essentially, templates can be saved as projects and a project can be saved as a template to be used to create other projects. The details on the table structure of the database component of the storage layer will be discussed later in this chapter. The next section explains the Graphical User Interface (GUI) of the Vi-Secanto tool.

4.2 Graphical user interface (GUI) for Vi-Secanto

The Graphical User Interface (GUI) is a collection of forms based on C# System Windows Forms classes. A GUI can be defined as a visual interface area, which can display output while allowing input for the simultaneous running of computer processes (Microsoft Corporation, 2007). The graphical user interface for Vi-Secanto was developed with principles based on Human Computer Interaction (HCI). These principles were stated in Chapter 3. For example, the compatibility principle ensures that the results of any control entry will be compatible with the user's expectations. Any user changes in a property of a security object will be reflected in the tree diagram view of the project. An example of the Consistency principle is that all colours for labels, forms and other GUI elements are consistent and the tool uses a standard colour schema. The Flexibility principle is reflected by the user being able to customize the tree presentation by choosing one of the existing colour schemas or by specifying their own. For the new colour schema the user defines the background colour and colours for each tree level. The Learnability principle ensures that the menu has a logical organisation and the tool provides the sequence of steps to complete different user tasks. The tool follows the principle of Minimal action. It provides default data and multiple choices for the user and a number of different Drop-down Boxes, Radio Buttons and Data Grid Views which assist the user to enter data easily. The Minimal memory load principle ensures that the tool doesn't use abbreviations and acronyms at all. The Perceptual limitation implementation principle shows that the user interface provides easily distinguishable colours for the user. Currently, as Vi-Secanto is only a prototype, HCI Principle 8 - *System Feedback should be friendly for the user* is only partly realised. All messages should be helpful to the developer. At the production stage system feedback needs to be friendly for all types of users.

Vi-Secanto utilises a number of GUI components from the .Net libraries: forms; Menus; toolbars; buttons; panels; tabControl; pictureBox; dataGridViews; and ColorDialogs; among others. Each form, presented as a window for the user, is designed to perform a number of tasks in order to achieve the desired functionality of the system. The form's elements provide possibilities for the user to perform an

action, such as entering data, or clicking on a button. The other GUI elements are represented by objects from classes designed and implemented by the researcher. These object classes support user interaction with the project tree view data. The tree view layout is the project data presentation provided by the tool. The vertical tree layout is currently used by the tool. The user can interact with the tree view in different ways: For example, to place additional nodes on the down level; to delete one of the nodes from the tree; to edit a node property; or to collapse branches.

The forms making up Vi-Secanto's GUI are listed and described in Table 4.1. Each form is a complex object which contains a number of control components. A screenshot of the application window and main menu bar along with the dialog used for changing the colour scheme is shown in Figure 4.3. The window displays a main menu, and provides the user with a number of choices, in order to achieve the desired result. The main menu can cascade in to number of submenus for item selection. For example, if the user first chooses the *Project* option from the Main menu, the next possible choice for the user will be either to open a new project, to open an existing project, to view project as a table or to look at the project history. The example shown in Figure 4.3 shows a window which allows the user to change the current settings for the tree's colour scheme. This GUI has a number of elements, which allow the user to interact with the application. These are buttons, a combo box, a text box, and radio buttons. Other examples of forms, used in Vi-Secanto's GUI, are shown in Appendix A.

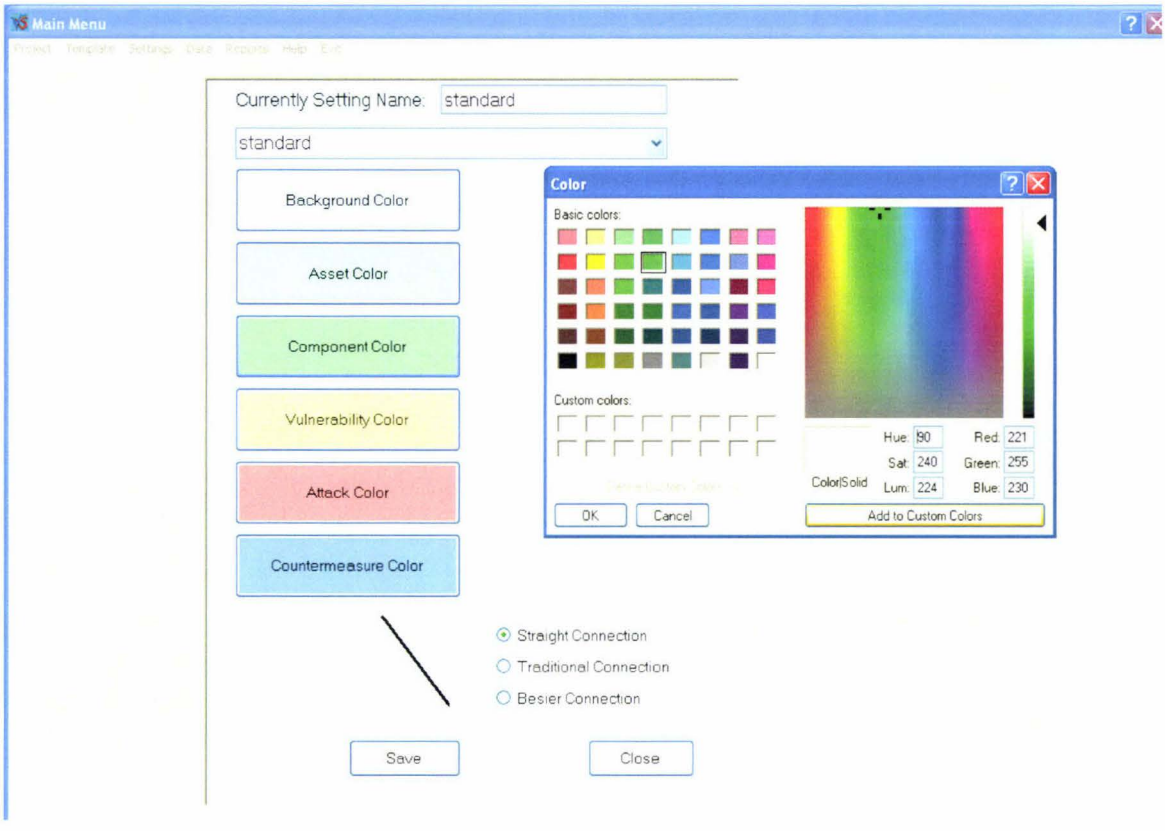


Figure 4.3 Vi-Secanto's Main Menu Form with Open Settings Panel, each button shows current colour settings for a particular tree level.

Table 4.1 *Forms in the Vi-Secanto GUI*

Form Name	**Descriptions**	**Types of the GUI components used**
Main	The main form presents the first tool window for the user. The menu contains: 1. Project (New, Open, History) 2. Templates (New, Open) 3. Settings (Graph Property (New, Edit)) 4. Data (Project Data, Library Data (Asset Templates, All Tables)) 5. Reports (Component Report) 6. Help 7. Exit This form provides links to other forms for the user.	MenuStrip, Panels, ColorDialog, TextBox, ComboBox, PictureBox, Button, radioButton, Label.
TreeBase	This form is used to present a list of projects, or templates.	MenuStrip, DataGridView, ComboBox, Buttons, PropertyPanel, Panel, Label.
Tree	This form is used to present a project, or template tree diagram, as an interactive diagram. This diagram can be changed by adding new leaves, or removing existing leaves, with the property of the leaves	Context Menu, MenuStrip, DataGridView, ComboBox, Buttons, RadioButtons, Panel,

Form Name	Descriptions	Types of the GUI components used
	then updated. All updates can be saved. If it is an existing template it can be saved as a new project. (the program code for this class Tree is presented in Appendix C)	ProgressBar, PictureBox, Label.
Form_Pict	This form is used to present a project, or template tree diagram, as a picture. It is possible to store it as a picture file.	PictureBox, Button.
form_Data	This form is used to present project data in the form of tables.	TabControl, DataGridView, BindingNavigator, Button
form_Lib_Data	This form is used to present template data in the form of tables.	TabControl, DataGridView, BindingNavigator, Button
form_Graph_Base	This form is used to present a list of projects in order to show the project's history.	MenuStrip, DataGridView, ComboBox, Buttons, PropertyPanel, Panel, BindingNavigator
form_Graph	This form presents the project's history graphically. The user can resize the graphic representation.	ZedGraphControl
form_Data_Asset	This form presents all the data fields for one Asset record.	BindingNavigator, TextBox, Label, Button.
form_Data_Attack	This form presents all data fields for one Attack record.	BindingNavigator, TextBox, Label, Button.
form_Data_Component	This form presents all data fields for one Component record.	BindingNavigator, TextBox, Label, Button.
form_Data_Countm	This form presents all data fields for one Countermeasure record.	BindingNavigator, TextBox, Label, Button.
form_Data_Vuln	This form presents all data fields for one Vulnerability record.	BindingNavigator, TextBox, Label, Button.

4.3 Classes for Vi-Secanto

These are classes from the .Net libraries. Some of them were used for creating database interaction within the program. The other classes were developed to serve the needs of the logic of the current program. For a full description of all object classes in Vi-Secanto, please refer to Table 4.2

Table 4.2 *Vi-Secanto Class Descriptions*

Class name	**Description**	**Derived from**	**Methods**
Asset	Presents Asset, keeps Asset data.	ShapeBase	GetStringProperty, Check_value, Invalidate
Attack	Presents Attack, keeps Attack data.	ShapeBase	GetStringProperty, Check_value, Invalidate
Component	Presents Component, keeps Component data.	ShapeBase	GetStringProperty, Check_value, Invalidate
Connection	Presents Connection between other security objects, keeps Connection data.		GetRegionConnection Paint
ConnectionCollection	Keeps collection of Connections	CollectionBase (.NET Framework Class Library)	Add, PaintConnector, RemoveObject GetIndexParentConnection
Countermeasure	Presents Countermeasure, keeps Countermeasure data.	ShapeBase	GetStringProperty, Check_value, Invalidate
ShapeBase	A base class to define a common property to present security objects as a rectangular shaped leaf in the tree diagram		Paint, AddChild, GetChildShape, GetChildConnect, GetStorageArea
ShapeCollection	Class to keep collections of different shapes	CollectionBase (.NET Framework Class Library)	Add, GetSelectArea, GetRoot, GetRootDB_ID, SelectObject, GetIndexSelectObject, GetIndexShape, PaintObjects, RemoveObject, GetPrShapesSumWidth, CountShape, SetNewPositionShape, GetDbId, GetParGId
Vulnerability	Presents Vulnerability, keeps Vulnerability data.	ShapeBase	GetStringProperty, Check_value, Invalidate

4.4 Database of Vi-Secanto

Technology has enabled the retention of electronic libraries of data, which can easily be mined for information. The solution for the proposed tool is to have security information stored in a database which represents a library, to fulfil the first goal of this research. The first goal states the aim is to make existing security information more accessible for developers. This database has two logically separated parts in

terms of content. One part is the security library for the storage of known web application security information such as countermeasures. The second part is for the storage of user specific project data. The Entity Relationship Diagram (ERD) is split into two logically separate parts (Figure 4.4 and Figure 4.5). The first part of the ERD displays tables, which provide storage for the users' project data. The ERD is presented in Figure 4.4.

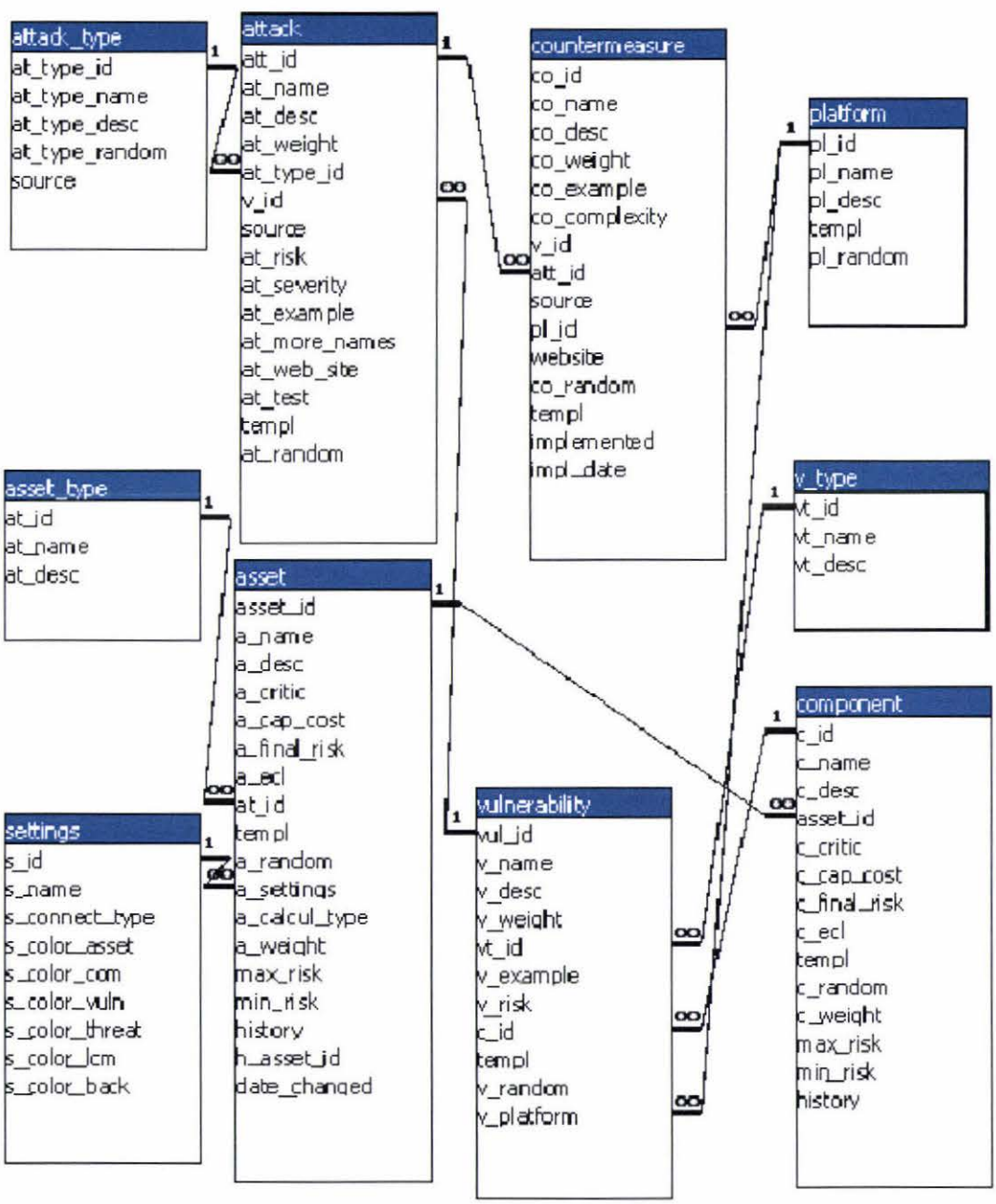


Figure 4. 4 Entity Relationship Diagram for the Vi-Secanto Database, Part 1: Project data

The second part of the Entity Relationship Diagram contains tables, which are

designed to store security library data (Please refer to Figure 4.5). Each part of the database stores different security entities: system components; vulnerabilities; attacks; and countermeasures; which are all linked to each other.

The Vi-Secanto tool utilises the Microsoft Access database system. Access is not a large database system, but it fully supports SQL and can be accessed from the external program. It also has an internal GUI for data manipulation. The database structure consists of eighteen tables.

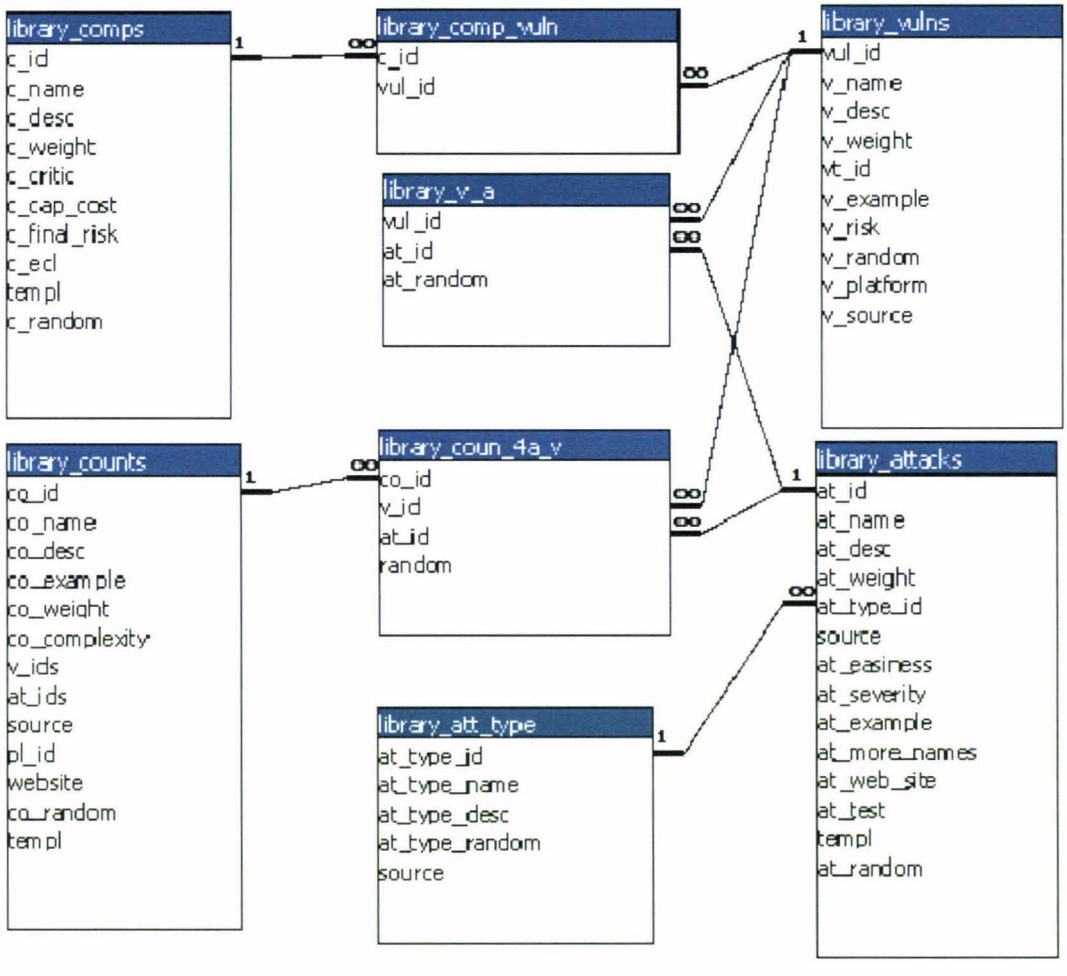


Figure 4.5 Entity Relationship Diagram for the Vi-Secanto Database, Part 2: Predefined Security data.

4.5 Working with Vi-Secanto

New projects can be initialised from the available templates. The templates represent

common knowledge for different web application types (see Figure 4.6 and Figure 4.7 and, see Appendix A: Figure 8 to Figure 16). They capture known vulnerabilities, attacks and countermeasures, and provide the associations between them. The user can make a choice by double clicking on the desired type of project; for example, *bank*. The program is then opened on the predefined template for the bank. The user can change the name of this abstract bank (*Bank XYZ*) to the desired name; for example, *CAT-Bank*.

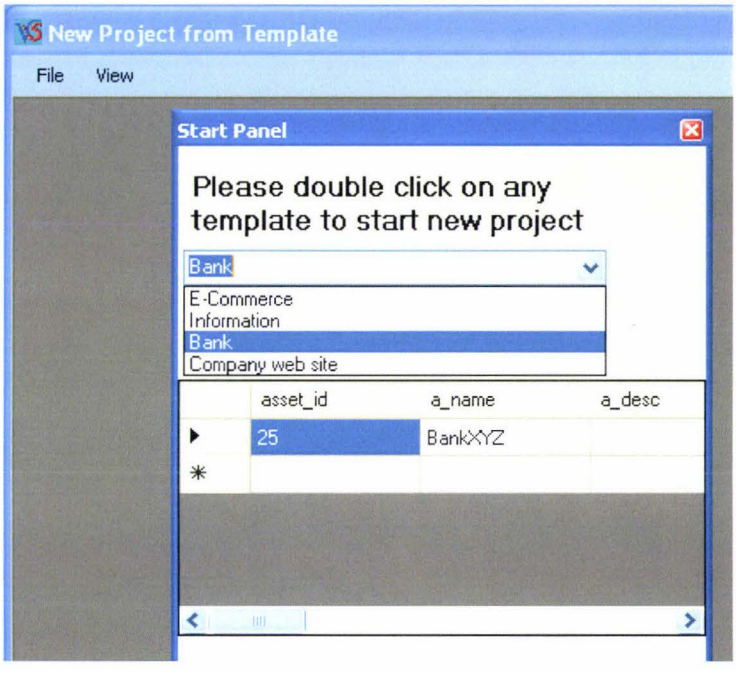


Figure 4.6 Vi-Secanto provides users with domain specific project templates

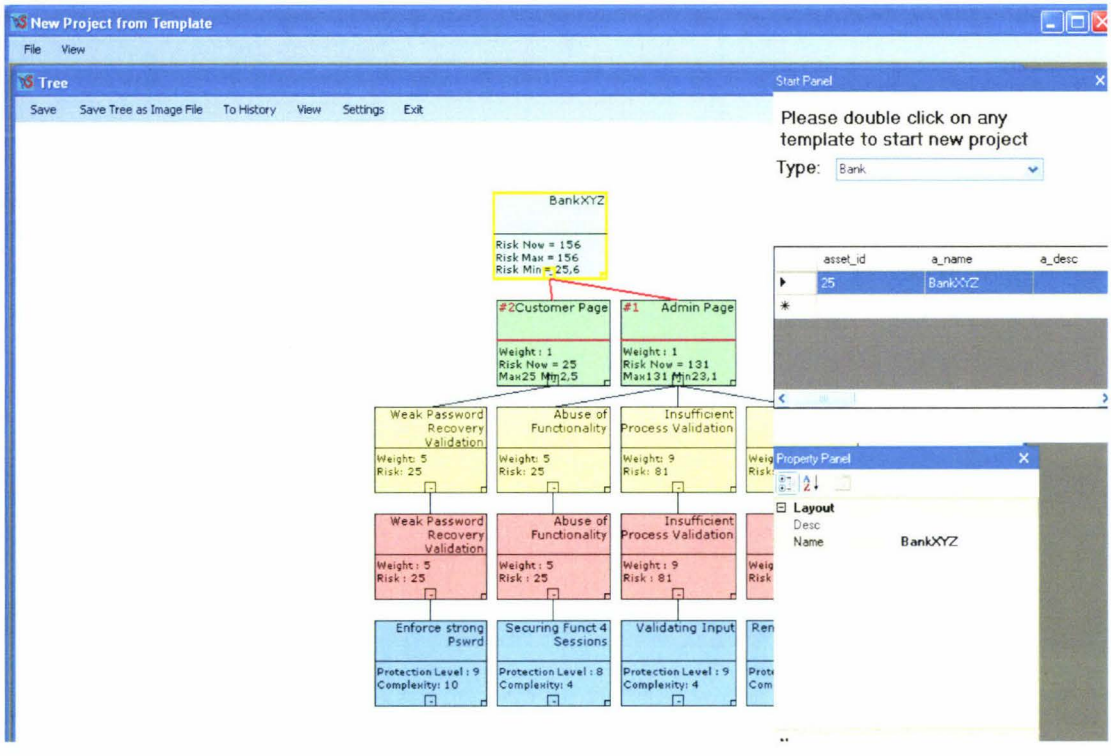


Figure 4.7 Example of the project template for a bank.

The project is presented to the user as an interactive tree. The tree has a number of branches, and five levels. The top level represents an asset (e.g. web application). There can only be one top-level asset per project. Each asset has a number of components (e.g. Home page and Shopping cart). Components are shown on the second level. Each component can have a number of vulnerabilities (i.e. weaknesses). Vulnerabilities are shown on the third level down. There can be one, or more, attacks that utilise a certain vulnerability to compromise an application. Attacks are shown on the fourth level. Countermeasures can minimise the impact of or prevent attacks from taking advantage of vulnerabilities. Countermeasures appear on the fifth level.

The following diagram (Figure 4.8) represents the navigation routes between different GUI forms (windows). The diagram shows that the main menu provides a link to all other windows. This functionality allows the user to fulfil all desired tasks within the tool.

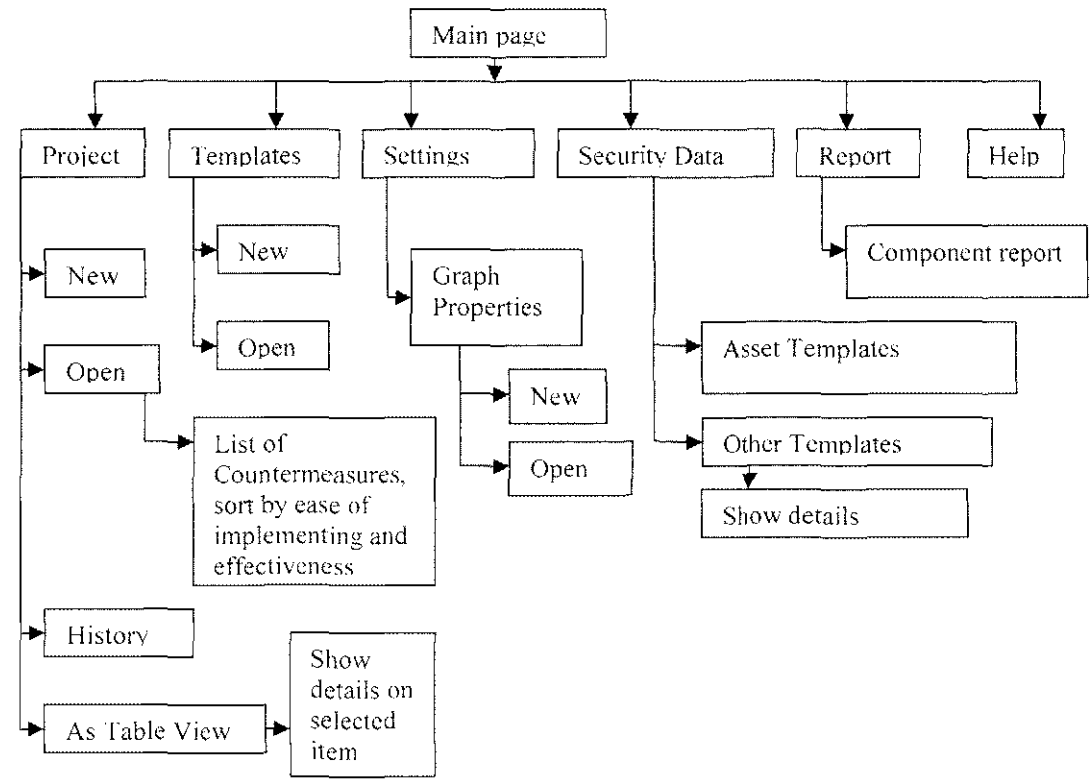


Figure 4.8 Navigation Diagram for the Vi-Secanto tool.

Chapter 5: Results of the Evaluation

The purpose of the evaluation steps was stated earlier in Chapter 2. There were three iterations of the development and evaluation stage. The results of each evaluation of the three iterations are discussed in the following sections. Each round of evaluations provided feedback. This feedback provided suggestions which helped to identify either additions, or revisions, to the existing functional and non-functional requirements of the tool, which were then used to improve the design of the tool. The settings and procedures for conducting these evaluations were discussed in the method chapter, Chapter 2.

5.1 General Procedures

The use of the working prototype was demonstrated at each evaluation session. There were differences as to how the three evaluations were organised. First, these differences were based on the fact that companies had different times for meeting with the researcher. The second reason was that, after the first evaluations of all the functionality features of the tool, it wasn't necessary to test all of the functionalities at all of the evaluations. The third reason was that, in the case of the third evaluation, the security team participated with the aim of assessing the proposed risk calculation algorithm, and assessing the functionality and usability of the security templates.

The participants were asked to record their perceptions of the tool in terms of satisfaction marks (from 1 to 10, where 10 is highest) for the functionalities of the tool, as well as answering general questions about the concept of the tool. The participants were asked to state additional suggestions and ideas. The evaluations focused primarily on usability, functionality and perceptions of the practicality of the design of the risk assessment procedure and its representation as a tree. The evaluations were based on the explicit criteria stated previously in Chapter 3.

5.2 Results from the First evaluation

The first evaluation was carried out by a web application development company. The company has fifteen staff. Four people participated in the tool's first evaluation. The four participants held three distinct roles: One security specialist (ss); one manager (m); and two developers (d1, d2), who were not security specialists. The evaluation of the tool was carried out as a structured meeting with each individual. At the beginning of the meeting, the participant received an explanation of the main purpose of the tool and a detailed explanation of the risk calculation method used. The next step for each participant was to use the tool. Participants were given a copy of the tool to try out. The length of this step was user dependant, but ranged from half, to one hour, in duration. The purpose of the session was for each participant to become familiar with the functionality of the tool. They were asked to carry out a number of tasks (see Table 5.1) and to record their perceptions of the tool in terms of satisfaction with the tool's functionality and their views on the value of having such a tool. A questionnaire with two main areas was used to record these responses. The first part of the questionnaire contained questions measured with a satisfaction scale (1 = very dissatisfied to 10 = very satisfied) to assess the current functionality of the working prototype. The second part of the questionnaire contained open-ended questions, which asked participants to think about what other functionality and/or security knowledge could be added to the tool in order to make it more user friendly and useful to their organisation. The median level of satisfaction across all questions of functionality was greater than eight (see Table 5.1).

Table 5.1 Results of the First Evaluation in the Web Development Company

Use Case (Main Menu Options)	**Alternative sub-flows (items under each main menu item)**	**Description**	**Satisfaction Median (Std. Dev) [range 1 to 10]**	**Comments**
Create a project (Project/New)	New from template	Choose the appropriate template from a table. Save it as a project with a new name.	8.5 (0.96) [8 to 10]	Place dropdown box in movable window (ss), (start panel should be resizable and movable)

Use Case (Main Menu Options)	Alternative sub-flows (Items under each main menu item)	Description	Satisfaction Median (Std. Dev) [range 1 to 10]	Comments
	New without template	Create a new project from scratch and save it.	9(1.50) [7 to 10]	
Open (Project/Open)	Open existing project	Open a saved project by choosing it from a list	10(0.50) [9 to 10]	
Edit tree diagram	Edit asset	Change the selected asset's properties.	10(0.00) [10 to 10]	Edit in movable window (ss) (property panel should be resizable and movable)
	Edit component	Change the selected component's properties. The tool recalculates residual risk for the component.	10(0.00) [10 to 10]	Edit in movable window (ss) (property panel should be resizable and movable)
	Edit attack	Change the selected attack's properties. The tool recalculates residual risk for the component it belongs to.	10(0.50) [9 to 10]	Edit in movable window (ss) (property panel should be resizable and movable)
	Edit counter-measure	Changes the selected countermeasure's properties. The tool recalculates residual risk for the component it belongs to.	10(0.50) [9 to 10]	Edit in movable window (ss) (property panel should be resizable and movable)
	Create a component -add new from template (See Figure 10 and 11 from Appendix A).	Choose a component template from a list to add to the current project.	9.5(0.96) [8 to 10]	
	Create a component - add new without template	Add a new component to the project.	9.5(0.58) [9 to 10]	
	Delete asset	When user delete's asset it is a deletion of the whole project.	10(0.50) [9 to 10]	Fix delete project. (then click cancel) (ss)
	Delete component	Delete a component and all nodes under it.	10(0.00) [10 to 10]	
	Delete attack	Delete an attack and all the nodes under it.	10(0.00) [10 to 10]	

Use Case (Main Menu Options)	Alternative sub-flows (Items under each main menu item)	Description	Satisfaction Median (Std. Dev) [range 1 to 10]	Comments
	Delete counter-measures	Delete countermeasures	10(0.00) [10 to 10]	
	Save new project	Save the current project as a new project.	9.5(0.96) [8 to 10]	
	Save project as template	Save current project as a new template.	10(4.50) [1 to 10]	Position in menu as well (ss)
	Save project template as new project	Save the current project as a new project under a new name.	10(0.50) [9 to 10]	Need to do (d1)
View project data as a table (Project/As Table)	Information about project presented in tables	The user shall be able to view project information in the table view.	9.5(0.58) [9 to 10]	
Access security database	Project templates (See Figure 8 and 9 from Appendix A).	Project templates (e.g. Shop, information site, bank) - includes components, vulnerabilities, Attacks, countermeasures and associations between them.	9.5(0.96) [8 to 10]	
	Component templates (See Figure 10 and 11 from Appendix A).	Component templates have components, vulnerabilities, attacks, countermeasures and associations between them.	9(0.50) [9 to 10]	
	Vulnerability templates (See Figure 12 and 13 from Appendix A).	Vulnerability templates have vulnerabilities, attacks, countermeasures and associations between them.	9.5(0.58) [9 to 10]	
	Attack templates (See Figure 14 and 15 from Appendix A).	Attack templates have attacks and countermeasures and associations between them.	10(0.00) [9 to 10]	
	Countermeasure templates (See Figure 16 from Appendix A).	Countermeasure templates contain details on countermeasures.	10(0.50) [10 to 10]	
Information on Project (project tree open)	From context menu choose "Save tree as Image file"	Information on the current state of the selected project. shows max Risk, min Risk and current Risk.	10(0.50) [9 to 10]	

Use Case (Main Menu Options)	Alternative sub-flows (Items under each main menu item)	Description	Satisfaction Median (Std. Dev) [range 1 to 10]	Comments
	From context menu choose "List Attacks" (See Figure 6 from Appendix A).	List of top attacks ordered by residual risk level. (see Figure 5.1)	10(0.50) [9 to 10]	
Report	(Report/ Components Report) (See Figure 7 from Appendix A).	Reports on components sorted by risk to show which components should be given high priority for protection via countermeasures.	9.5(0.58) [9 to 10]	

Table 5.2 *Results of the First Evaluation in the Web Development Company; General Questions*

Questions	Aim of the question	Satisfaction Median (Std. Dev) [range]	Comments
Overall Idea of the tool	This question asks for the user's opinion on the concept of having such a tool to see if they see it as useful in their own work.	10(0.50) [9 to 10]	Excellent idea. Would be ideal if the tool can be sold (or account of tool given to companies) once security experts have done their work.(m) It will be very easy to assess the risks, the tool will help junior/medium level programmers (m) Good idea, a multiple users version will be needed in the production stage (ss)
How useful you find it for your company	This question is asked to find out if the tool is seen as useful for their company.	10(0.00) [10 to 10]	Useful for large projects and for educational purposes. Very useful. Will consider using in our company (ss)
Possibility to use it in your company in future	This aim of this question was to solicit the user's opinion on future use of the tool in their company.	10(1.00) [8 to 10]	Yes (m) (ss)

Overall, the above results of the first evaluation show that the working prototype was perceived as easy to use and useful by the individual evaluators, who represented the three types of potential users. The main goals of the research were to; make existing security information more accessible to developers, and to integrate the security risk assessment process into the requirements and early design stages of the software

development process. The proposed tool supports both of the stated goals. First of all, the tool stores security knowledge in a database and displays that knowledge to users as templates (see Figures 8 and 9 in Appendix A). Secondly, the tool provides a modular way for the user to carry out a risk assessment (see the Edit tree diagram section of Table 5.1) which allows for making changes to related security entities and having those changes automatically incorporated into an update of the asset's residual risk value. The other features of the tool; such as easy to use navigation mechanisms and an easily understandable interface; met the user requirements. The users learned to use the tool quickly and gave very high marks on the tool's functionalities (see Table 5.1).

The overall concept of the proposed tool was found by the evaluators to be very interesting. Examples of comments were:

- "Excellent idea. Would be ideal if the tool can be sold (or account of tool given to companies) once security experts have done their work." (m)
- "Good idea, multiple users will be needed in the production stage" (ss)
  [The participant meant that in a real situation; a number of people work on the same project. They wanted the tool to work like a group decision support system to provide the same information for the same project, at the same time, to several users and to keep track of changes made by each user.]

The participating company expressed an interest in adopting the working tool for everyday practice. Participants found it to be useful in supporting their work. They stated that the tool was:

- "Very useful. Will consider to use in [our company]" (The real name of the company has been deleted to preserve the anonymity of the participants)(ss)
- "Useful for large projects and for educational purposes" (m)

Table 5.3 *Additional Comments from the First Evaluation in the Web Development Company*

	The description can be presented more user friendly by pop up window after double click (ss)
	Great tool to try. Hope it will be in the market (d1) Absolutely good work! (d2)

As a result of the evaluation, a number of suggestions for improvements were made by the participants. Most of the suggestions and comments came from the security specialist. Some of the comments were related to tool interface improvements, such as "Place dropdown box in the movable window" (see Table 5.1) and "The description can be presented more user friendly by pop up window after double click" (see Table 5.2). These comments refer to the Start Panel and the Property Panel displayed below in Figure 5.1. For evaluation purposes a number of sample projects were created. One of them was *CAT-Bank* shown in Figure 5.1.

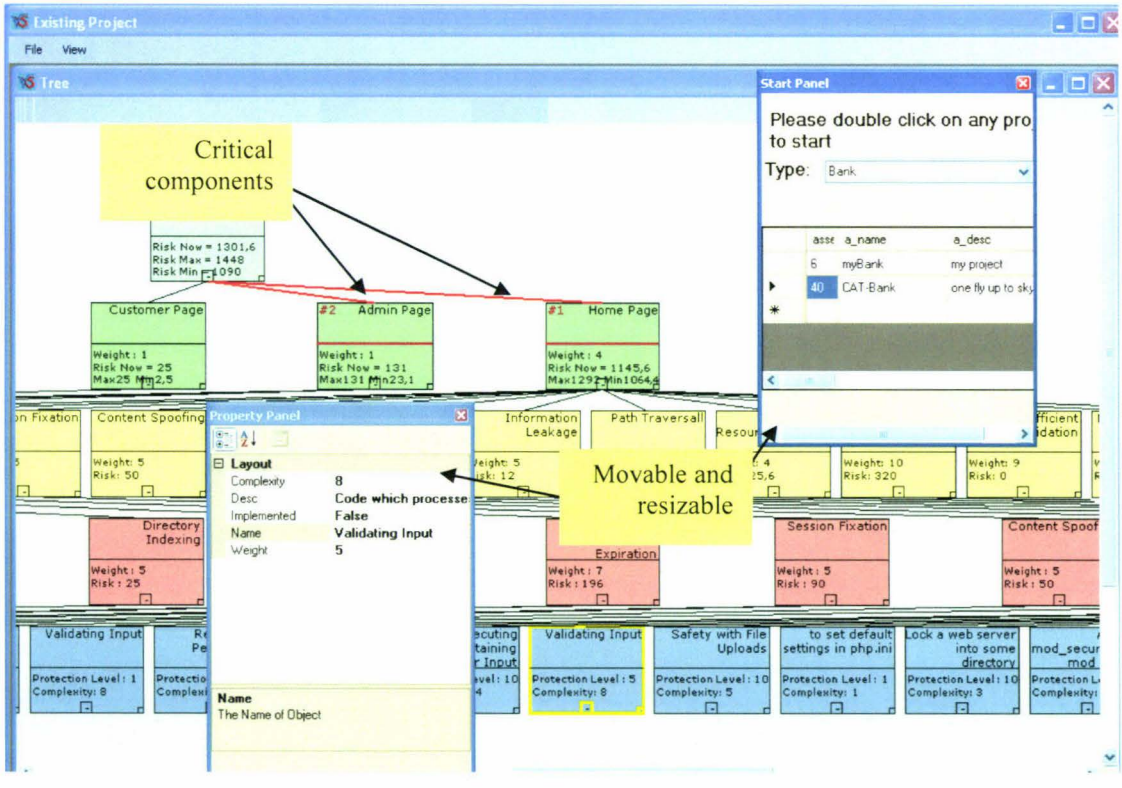


Figure 5.1 Screen shot of a tree diagram with a movable Start Panel and Property Panel. The two most critical components are marked with red lines and numbers.

The manager stated that, "the greatest benefits to be realised from the tool would be for big projects and to educate new developers". The developers are not very aware of security and risk issues related to their job roles, as demonstrated by the following comment from a junior developer: "Why do we need to worry about security, we have him (security specialist), and he needs to worry about all security issues". On the other hand, the developers were happy to see project security requirements modelled and presented as an interactive tree diagram.

The participating company is currently developing an online quotation system for their customers, which is based on viewing a web site project as a number of different components, each component having different functionality and potentially different security risks. For example, one of their components is called Online Payment, and, another one is called Administrative Login. A user can obtain an online quote for their required system, based on the cost of each component it will contain. Each new web site development project is constructed from user-selected, pre-defined components, in a sense similar to building a model from the existing parts of a LEGO set. The predominance of positive comments for the proposed security tool from this company may be due to the fit with their desire to build client web applications from reliable components which have been designed with security risks in mind. The manager indicated that they saw the proposed tool as an essential extension to their new component-based plan for building web applications. The tool was seen as helpful in performing risk calculations and in using existing security risk knowledge on possible vulnerabilities that can exist in standard web site components they deal with, such as online payments, logins and shopping carts.

Suggestions considered to be within the scope of the prototype were used to improve the design and implementation. Figure 5.2 shows the history graph added as a result of the first evaluation. Also, a suggestion to make movable windows for the Property window and for the Start window was accepted. Other small bugs were fixed.

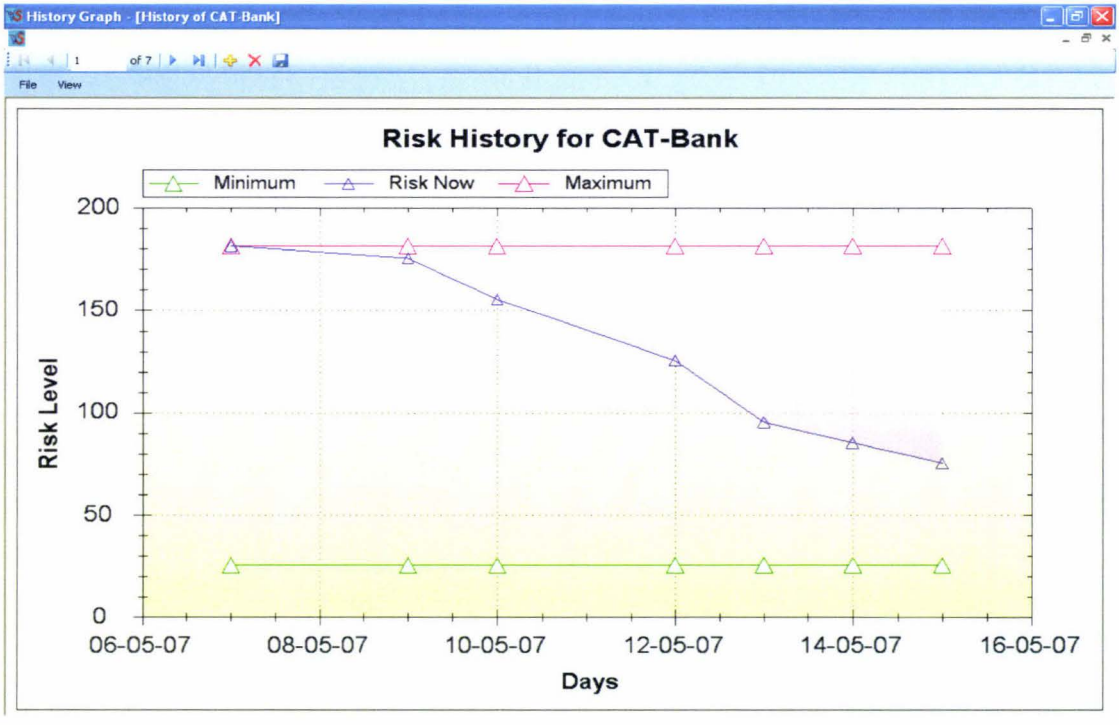


Figure 5.2 Example of a risk history for the hypothetical Cat-Bank Web Application.

5.3 Results from the Second evaluation

The second evaluation was carried out by one of New Zealand's largest financial services companies. The company employs over six-thousand people. A Software Infrastructure Specialist with eighteen years of experience in software development on Microsoft platforms participated in this evaluation. A PowerPoint presentation, which included an explanation of the main purpose of the tool and the risk calculation method used, was given at the start of the evaluation session. Next, a demonstration of the working prototype tool's functionalities was given. The participant decided to fill in the questionnaire afterwards, and emailed the completed questionnaire the next day. The length of the evaluation session was one hour.

The questions asked were similar to those used in the first evaluation, with additional questions added at the end of the questionnaire. The additional questions were designed to encourage the participant to think about additional functionality and/or security knowledge which would be desirable to have in a security risk assessment tool. One of the questions was designed to evaluate the use of relative ratings to weigh

different aspects of risk. Unfortunately, this participant didn't complete the questions on functionality. The same questions were used as in the first evaluation (see Table 5.1). He stated that he "didn't get a thorough enough look at the interface to make a judgement". Nonetheless, he did complete the open-ended questions, as shown in Table 5.4.

Table 5.4 *Questions and Answers from the Second Evaluation*

Questions	Comments
Overall idea of the tool?	The idea of the tool is great. To consolidate the security knowledge and show the effect of threats and attacks in a graphical tool in some measurable way is a big step forward from the "code it and see" approaches that is typical of almost all projects.
How useful do you think it would be for your company?	
Possibility of use by your company in the future?	I would think we would find the tool useful here, even as an educational tool to show developers what threats exist and how to code around them. Increasing awareness of security issues would be a large benefit.
What additional parameters do you think need to be kept in the system?	
This tool does a risk calculation based on weight (from 1-10). Do you think that it is a good idea? Is it easy to use?	Yes it is. The way the figures are displayed takes a little getting used to. The fact that it multiplies down the tree and then adds back up the tree (or is it the other way round) is a little confusing.
What kind of additional functionality would be useful for this tool?	Showing a list of threats/attacks and countermeasures in order of return on investment. I.e. which countermeasures give me the biggest benefit for the least amount of work?
Please state other suggestions here.	If the user interface updated as things were changed without requiring a save I think it would flow better. The user could then play around with different options and save or discard them more easily. I'd be interested to see how the interface copes with larger scale projects. The tree would get very wide very quickly I'd think and seeing it all may be an issue. Maybe an option to display as a table would be useful? Not sure.

The participant liked the idea of the tool: "The idea of the tool is great". He commented that the tool unites the security knowledge and presents this information in a graphical and measurable way. "this is a big step forward from the 'code it and see' approaches that [are] typical of almost all projects" (see Table 5.3). He also commented that the tool could be useful for the company as an educational tool. The participant stated that the greatest benefit the company could derive from this tool, would be to "increase the awareness of security issues". This feedback demonstrates

that the tool provides security knowledge to developers, as stated in its goal. The participant found the idea of a risk calculation based on weighting (from 1-10) to be good, however, he found that the risk calculation formula "...multiplies down the tree and then adds back up the tree ... a little confusing".

One suggestion from the participant was to have a list of threats/attacks and countermeasures, showing which countermeasures gave the greatest benefit and were the easiest to implement. This suggestion was implemented in the tool as an improvement. Other suggestions were made. One; the ability to work with the tree diagram and to have the file "updated as things were changed without requiring an explicit save"; was considered, but the functionality of "auto save" was not implemented due to its complexity. It will be recommended for the production version of the tool.

The participant was concerned as to how a very large tree diagram would fit onto a standard computer screen. This concern could be addressed through the use of a collapsing branches structure. This functionality was implemented previously in the first iteration. However, the participant did not raise any questions about scaling the tree diagram problem during the evaluation time. The participant only wrote down the suggestion on the next day, not knowing that the feature already exists in the system. He didn't notice that he can actually collapse the branches of the diagram with a click of a mouse button. Each security object in the tree diagram has a small square with a "-", which can turn into a "+", when a branch is collapsed. This small square is placed in the bottom of each tree leaf (See Figure 5 in Appendix A).

5.4 Results from the Third evaluation

The third evaluation was carried out by six people from the same large financial services company who participated in the second evaluation. There were three distinct user roles in this group: Security specialists (two people); technical specialists (three people); and developers who were not security specialists (one person). The evaluation began with a presentation by the researcher to the six participants on the research purpose and the problem the tool was intended to address. The presentation included a detailed explanation of the risk calculation performed by the tool. The

second part of the presentation was a demonstration of the functionality of the working prototype. At the end of the meeting each person filled out a questionnaire. The length of the meeting was one hour.

This questionnaire was different from the one used in the first evaluation, since the purpose of the final evaluation was different. First of all, it was shorter, as the basic functionality of the tool had already been assessed in the first evaluation and time constraints limited the final evaluation to one hour. This questionnaire was designed to detect additional functionality, or security knowledge, perceived to be necessary to improve the utility of the tool and to obtain suggestions for improvements. The median numerical response to each question, along with a number of comments, are summarised in Table 5.5. Overall, the participants were happy with the purpose of the tool, perceived it as useful for their company and saw the possibility of using it in the future.

A number of comments and recommendations were made by those evaluating the tool. A suggestion to make it multi-user was made. This is out of scope for this research project, but it would be recommended for the production stage of the tool. A report function, in order to allow the production of reports in different formats, was also suggested. This suggestion was implemented. Another suggestion made was that, "a quality assessment or proof of concept step would be [useful] to evaluate the software before the software goes into a full implementation". This statement suggested that there should be a quality assessment of the final version of the production tool by security specialists, before it is put into everyday use. This suggestion is out of scope for this research, due to time constraints for a one-year Masters degree. Future research, with a larger sample of participants, will be needed to validate the acceptability of the tool with Security and Software Development Professionals. The use of relative weights (from 1-10) for each component of risk was perceived to be practical in terms of users being able to provide this data as the required input for computing risk. In addition, participants were positive about the use of security templates as a way to bring security knowledge to non-security experts. Due to the restricted presentation time for the second evaluation, one of the participants did not gain a clear understanding of the template concept: "I am unclear of this. Can each web application template be exactly the same as another?" In

actuality, the tool has different types of templates. Web application projects of the same type (i.e. made up of the same types of components) could use the same security template as a starting point. For example, banks have web sites made up of similar components to provide an online banking service. In this case, a user would start with the basic bank web site security template, would save it under new name as a project, and would make modifications and additions as needed. Another participant stated, however, that: "After a brief overview it was easy to understand". Another participant suggested the addition of a help system with a tutorial, would make the tool's features clearer. A simple help system was developed and implemented. Another useful suggestion was to provide an interface to easily add new, or customised/personalised, reports on the existing data. Due to the limited time frame this suggestion was not implemented, but could be considered in future enhancements to the tool.

Table 5.5 *Results of the Third Evaluation*

Questions	**Satisfaction Median (Std. Dev) Range: 1 to 10**	**Comments**
Overall idea of the tool?	8.5(1.9)	Good if it was multi-user.
How useful do you think it would be for your company?	6.5(1.9)	Great if it can output reports in the various format e.g. word, PDF, HTML.
Is there any possibility that it would be used it in your company in future: All tool features? For security educational purpose for developers?	 5(2.5) 8.5(2.2)	A quality assessment, or proof of concept, would be implemented to evaluate it first.
This tool does a risk calculation based on weights (from 1-10). Do you think it is a good idea? Is it easy to use?	 7.5(2.1) 5.5(1.3)	
This tool has the ability to use security templates in order to bring security knowledge to non- security experts. Do you think it is good idea? Is it easy to use?	 10(1.9) 5(1.1)	I am unclear of this. Can each web application template be exactly similar to another? After a brief overview it is easy. A good help system with tutorials would be good. As others suggested there should be an interface to easily add new/custom-personalised data. Reports should be easy to generate.
There is a need to regularly update security knowledge which is kept in this tool database. A suggestion was made to put this into final development and support to open source. Do you think it is good idea?	 10(0.4)	Yes – suggest OWASP or CVE, MITRE or SANS. Keeping information updates is critical. Consider providing automated update service. A team will be required to assess validity of new data/information added.

Participants were strongly supportive of a suggestion to make the tool available as an open source product, with one participant suggesting the use of OWASP, CVE, MITRE, or SANS. They also underlined that keeping security information current was critical. An automated update process for the tool was strongly suggested. This suggestion is very important, however, due to the limited time available for this project it was not possible to implement, but will be put forward as a necessary feature for future development.

Table 5.6 *Additional Questions from Results of the Third Evaluation*

Questions	Comments
What would stop your company from using it?	Maintenance. High $$, Cost. Cost, usability.
What do you think about any additional parameters which need to be kept in the system?	Cost to companies if exploited.
What kind of additional functionality would be useful for this tool?	If tool could scan an application rather than give guidance, it will be helpful. Ability to sort threats and countermeasures by source e.g. OWASP top 10 PCI (VISA/Master card) Your own custom list. For example, we have an existing list of web technical security standards that web applications must comply with. (They are reviewed against these.) It would be fantastic if the user could import these standards into the tool, so that custom reference are displayed such as: "cross site scripting" ref: OWASAP =#3 PCI = Section 5.2.1 ABC-company web standard = section F This helps give context + greater relevance Plug-in to IDE.
Please state other suggestions and comments here	As mentioned, OWASP/PCI, or Custom, standards should be/can be included. Interesting concept. Security is important.

Participants also answered open-ended questions from the additional questions section of the survey. They stated that reasons that would prevent their company from using this tool would include: Maintenance; high cost; and usability. With respect to this research, maintenance and high cost are out of the scope of the project. Usability of the tool is one reason that the research project included carrying out external evaluations with practicing developers, managers and security experts. Most of the participants stated that this tool would be useful for their work and found it easy to use. Another suggestion that was out of scope of the research was that the tool should scan the application for vulnerabilities, instead of giving guidance on

modelling security requirements. Other suggestions were to provide the ability to sort threats and countermeasures by source and to make the tool a plug-in for an integrated development environment (IDE). Another valuable suggestion came from the participants, when they answered an additional set of questions (see Table 5.6). It was advised to relate existing web technical security standards (international standards that web applications should comply with) to the security data stored in the tool. Due to the time limitations of this research and the complexity of that implementation, this suggestion is left for future enhancements to the tool.

Chapter 6: Discussion

Creating secure web applications is a business process. As a business process, it needs to be analysed, measured and improved in order to achieve the desired security level for web applications (Cohen, 2005). The desired security level can be achieved through the risk management process. The risk management process contains the four phases of identification, quantification, controlling and monitoring (Faisst & Prokein, 2005). Current web application management practices focus on the last two phases. In order to improve the risk management process the first two stages should be embedded in the web application development. The importance of improving security in the development of web applications has prompted the recent appearance of other software packages which deal with web application security in the earlier stages of development. The other tools take a similar approach in the tackling of security problems at the beginning of their development. This chapter presents a comparison of the proposed security tool with two other existing security assessment tools: Microsoft's Application Security Threat Analysis & Modelling tool (Microsoft Corporation, 2006a); and Practical Threat Analysis (PTA) (PTA Technologies, 2006). These two tools were downloaded and evaluated. The difference and similarities are discussed from the perspective of the user. The first part of the chapter is a discussion of the risk management process and risk assessment, as a major part of that process. The current method for risk calculation, as part of the risk assessment process, will also be examined and discussed. The second part of this chapter is a comparison of the two security assessment tools with the proposed tool. At the end of this chapter is a discussion of how the positive, and negative, results from the final evaluations impact on the practical uses of the tool for each target user group.

6.1 Software security and risk management

The history and evolution of approaches to risk management in the field of software development can be viewed in terms of three generations (Fletcher et al., 1995). The first generation of risk management was based on concepts of confidentiality, integrity and availability (known as the CIA risk model). The same classification has, in the past, been commonly used for all systems. The main component of a software system was previously a mainframe computer. The mitigation strategies were

predefined, and mainly consisted of site-specific disaster recovery procedures. As all systems were characterised as being stand alone, a security problem with one system never posed any risk to other systems. Unfortunately, writers on the history of software security have not specified actual time frames for each of these generations. Based on their information that the first generation of risk management dealt with stand alone mainframes, it is, however, possible to estimate that the first generation occurred from 1950 to the late 1960s. The second generation began in the late 1960s when the first networks appeared, which were in the form of mainframes and multiple, networked terminals. At the time when networks first appeared, distributed systems appeared as well. Distributed systems development led to the need for the second generation of software security.

The second generation was characterised by the appearance of risk assessment tools for software systems. Most of the tools used some form of computerised checklist for risk assessment and provided guidance on the use of predefined countermeasures. The main focus of these tools was on the calculation of annualised loss expectancy using the monetary value of the business assets. First generation risk mitigation approaches and the CIA risk model were still in use, but it was hard to fit this approach to all situations and to all needs (Fletcher et al., 1995). Demand for system-specific risk assessment for software systems began to appear at this time. An analytical risk analysis approach which was developed in other fields, such as aviation, nuclear power, and munitions was used as a model of the development of new approaches to risk analysis in the software industry. The new approach to risk analysis was based on the components listed below (Fletcher et al., 1995):

- Vulnerabilities;
- threats (both active and passive);
- assets (data, hardware and software);
- impacts (disclosure, destruction, modification and unavailability);
- types of mitigation (avoidance, transfer and reduction);
- threats, reducing vulnerabilities and reducing impacts; and
- detection, response and recovery.

Risk evaluation became more general and was based on the concept of threats being

realised through vulnerability and the impact on assets (Fletcher et al., 1995).
Fletcher et al. (1995) indicated that the third generation of risk management will have the characteristics listed below:

- Provide total risk management;
- provide system operations based on the right level of security for access;
- ensure control, integrity, availability and safety;
- provide support for understanding of real threats, existing vulnerabilities and the practicality of countermeasures; and
- provide a solid basis for making system decisions.

Third generation tools (started from early 1990) were designed for the purpose of applying risk management to the whole lifecycle of the software system. The key concept is to undertake the appropriate development from the beginning of the development process, to do it well, and to provide the appropriate protection (Fletcher et al., 1995). The proposed tool, Vi-Secanto, provides all of the second generation functionality and some of the third generation functions using this classification. It helps users to deal with risk assessment (which is part of risk management) from the beginning of the development process and can be used to assess and track the impact of changes throughout the lifecycle of a web application. Vi-Secanto does not yet support user role modelling in terms of differential levels of access for components of a system.

An IT Risk Management survey conducted by Symantec reported that 60% of the 500 participants (from IT managers to top IT executives in international organisations) expect a major IT incident to occur at least once a year. A large number of companies still view IT risk management and IT controls as an unpleasant necessity. They consider activities to secure the IT environment as being limitations on their systems and irretrievable investments in time, money, or resources. IT professionals are, however, aware of the different security risks that an organisation typically faces. In the area of application development an understanding of the importance of security is growing. IT professionals recognise the effectiveness of preventative measures, such as making the source code resistant to application vulnerabilities by eliminating them at earlier stages of development. The necessary technology requires considerable early

investment in terms of tools and skills. For most organisations, secure application development practice is still in an early stage of adoption, even though it has been confirmed as being very cost-effective over time (Symantec Corporation, 2007).

An important aspect of such risk management is risk analysis. Risk analysis methodologies can be grouped into two primary categories: Commercial (including Microsoft's STRIDE, Sun's ACSM/SAR, Insight's CRAMM and Cigital's SQM); and standards based (the National Institute of Standards and Technology's ASSET, or the Software Engineering Institute's OCTAVE) (Verdon & McGraw, 2004). The different risk analysis methodologies use different risk calculation approaches. These approaches to risk calculation can be classified into three main categories: Qualitative; quantitative; and mixed approaches. The Australian and New Zealand national standard for risk analysis (Standards Australia and New Zealand, 1999) allows the use of both approaches (qualitative or quantitative) for risk assessment. The benefits and losses to taking either a solely quantitative, or a solely qualitative approach have given rise to mixed approaches. The greatest benefit of the quantitative analysis approach is the use of independent objective metrics and the provision of numeric results for comparison, such as presenting potential loss in monetary units. These types of outcomes can be easily understood by business managers (Farahmand, Navathe, Enslow, & Sharp, 2003). Improving the understandability of the risk analysis process for business managers will assist businesses in the realisation of their security goals (Hamdi & Boudriga, 2005). The results of quantitative risk analyses also help to clarify distinct goals. This type of analysis usually requires the following inputs (Microsoft Corporation, 2006b):

- Monetary value of each asset;
- a list of the main, or most important, threats;
- each threat's probability;
- the potential loss from each threat; and
- a list of countermeasures and their applicability.

One of the bad points of this type of approach, which is usually voiced by its opponents, is that a large amount of initial work is required and accurate estimates are often difficult to obtain. This work includes: assigning numeric values to security

risks; specifying countermeasures; estimating probabilities of occurrence; and doing complex calculations (Farahmand et al., 2003). As the numbers are based on estimated figures, it is hard to calculate the true value of any potential damage to the business. For example, the exact cost of publicly exposed data will be difficult to calculate. The process of applying quantitative risk management to all aspects of a business can be extremely costly for an organisation (Microsoft Corporation, 2006b). As a manual calculation can be difficult, and time and error prone, automatic tools are essential to make this process quicker, easier and less error prone (Hamdi & Boudriga, 2005).

Qualitative risk assessment, unlike quantitative risk assessment, doesn't require the assignment of exact values to the financial impact of a security problem. Instead, relative values are used to distinguish higher risk threats from lower risk threats (Microsoft Corporation, 2006b). The main advantage of the qualitative risk management and analysis approach is that only simple calculations are required and they do not involve any qualifications for threat frequency (Farahmand et al., 2003). There is no need to calculate exact values for the possible impact and cost of countermeasures; only relative values need to be determined. This saves time for businesses and makes risk analysis simpler (Microsoft Corporation, 2006b). As risk analysis becomes simpler, it can be more easily understood by high-level managers. This can give managers additional encouragement to apply the risk assessment process (Hamdi & Boudriga, 2005). A disadvantage of the qualitative approach is its more subjective nature (Farahmand et al., 2003). The expertise for qualitative analysis is based on the knowledge of the security team. This can be a disadvantage, due to the high cost of security specialists (Hamdi & Boudriga, 2005). In summary, it is difficult to determine which security protections are most important and, therefore, need to be considered as priorities for early implementation. Furthermore, it is impossible to justify a desired investment in security countermeasures when no figures are available for the necessary cost-benefit analyses (Microsoft Corporation, 2006b).

Vi-Secanto facilitates cost-benefit analysis by supporting risk assessment using an interactive tree diagram. This diagram shows minimum, maximum and current risks for the whole system and for each component. The Management team can easily see

which system components are a more desirable investment in terms of both the effort required to implement each countermeasure and the impact of doing so on risk reduction.

Different business organisations can adopt different approaches for risk management and analysis. Microsoft recently provided a summary table comparing the benefits and drawbacks of the quantitative and qualitative approaches to risk management (See Table 6.1). Microsoft's security risk management process claims to offer a combination of the best elements of these two approaches. Their approach is said to provide a faster risk management process than that of a typical quantitative approach and more detail for business decision making than is provided through a typical qualitative approach (Microsoft Corporation, 2006b).

Table 6.1 *Benefits and Drawbacks of Each Risk Management Approach*

	Quantitative	**Qualitative**
Benefits	• Risks are prioritised by financial impact; assets are prioritised by financial values. • Results facilitate management of risk by return on security investment. • Results can be expressed in management-specific terminology (for example, monetary values and probability expressed as a specific percentage). • Accuracy tends to increase over time as the organisation builds a historic record of data while gaining experience.	• Enables visibility and understanding of risk ranking. • Easier to reach consensus. • Not necessary to quantify threat frequency. • Not necessary to determine financial values of assets. • Easier to involve people who are not experts on security or computers.
Drawbacks	• Impact values assigned to risks are based on subjective opinions of participants. • Process to reach credible results and consensus is very time consuming. • Calculations can be complex and time consuming. • Results are presented in monetary terms only, and they may be difficult for non-technical people to interpret. • Process requires expertise, so participants cannot be easily coached through it.	• Insufficient differentiation between important risks. • Difficult to justify investing in control implementation, because there is no basis for a cost-benefit analysis. • Results are dependent upon the quality of the risk management team that is created.

Source: (Microsoft Corporation, 2006b).

The Proposed tool, Vi-Secanto, used relative weights (from 1 to 10, where 1 is the lowest and 10 is the highest rating) for the security objects (i.e. components, attacks, vulnerabilities, and countermeasures). This approach is a combination of qualitative risk assessment and quantitative risk assessment, providing benefits from both (see Table 6.2) while minimising some of the drawbacks of each (see Table 6.3).

Table 6.2 *Benefits of Using Relative Weights in the Vi-Secanto Tool*

Benefits of Relative Ratings
• Risks are prioritised. • Results facilitate management of risk; shows different options for risk minimisation. • Enables visibility and understanding of current risks and compares it with minimum and maximum risks. • Easier to reach a consensus for management. • Not necessary to determine precise financial values of assets; does relative values instead • Easier to involve people who are not security or computer experts. • Accuracy tends to increase over time as the organisation builds a historic record of data while gaining experience.

Table 6.3 *Minimisation of Drawbacks by Using a Hybrid Approach*

	Drawbacks Source: (Microsoft Corporation, 2006b).	**How Vi-Secanto Minimises drawbacks**
Drawbacks of Quantitative	• Numeric values (e.g. probabilities) assigned to risks are based on subjective opinions of participants. • Process to reach credible results and consensus is very time consuming. • Calculations can be complex and time consuming. • Results are presented in monetary terms only, and they may be difficult for non-technical people to interpret. • Process requires expertise, so participants cannot be easily coached through it.	• Impact values assigned to risks are based on opinions of security experts. • Process to reach credible results and consensus is quicker due to access to security information via the tool. • Calculations are performed quickly by the tool. • Results are presented as relative numbers so they are easy to compare and interpret. • Expertise is supplied with the tool so non-experts can use it.
Drawbacks of Qualitative	• Insufficient differentiation between important risks. • Difficult to justify investing in control implementation, because there is no basis for a cost-benefit analysis. • Results are dependent upon the quality of the risk management team	• Mediates this by providing security objects rated by experts for use by non-experts. • Tool ranks components by risk & effort providing a basis for cost-benefit analysis.

Paddon describes a qualitative, or non-quantitative, model as a model where risk evaluation for different levels of threat against asset value are assigned the values of low, medium, or high (Paddon, 2000). Table 6.2 shows how threat severity relates to asset value using these three values. The scale (low, medium, high) could be extended to provide more precise analysis results. When estimating the overall risk to an asset; calculated as a function of its value and its threat severity; no numeric value is assigned to the three different levels of risk. This model only provides a simple risk evaluation. For example, if asset value is high and threat severity is low, risk will be described as being medium, see Table 6.4.

Table 6.4 *A Non-quantitative Approach for Evaluating Risk*

		Threat severity		
		low	**medium**	**high**
Asset value	**low**	low	medium	medium
	medium	medium	medium	high
	high	medium	high	high

Source: (Paddon, 2000).

From this model it is hard to determine how much a business should invest to ensure a particular asset will have reasonable protection, as there is no measure of asset value in monetary terms. It is also difficult to make comparisons across organisations, since the terms low, medium and high are relative. A business is unlikely to spend more than an asset's monetary value to protect its security. Vi-Secanto displays relative numeric weights to represent an asset's current risk, maximum risk and minimum risk. This allows for making comparisons between different system components in terms of relative level of risk. Decision making on the set of countermeasures that will give maximum protection with minimum investment is also supported by showing relative effort to implement each countermeasure. Each countermeasure has a relative weight, which is an indicator of how good it is at reducing a risk. It also has a parameter which shows how easy this countermeasure is to implement. This parameter is a proxy for the investment required for this countermeasure.

A number of quantitative models and methods to assess risk can be found in the literature. The main advantage of these models and methods is that risk is represented

as a measurable numeric value. One of the risk assessment methods expresses risk as a financial loss (Verdon & McGraw, 2004), referred to as annualised loss expectancy (ALE):

$$ALE = SLE \times ARO; \quad (1)$$

where SLE is single loss expectancy; and ARO is the annualised rate of occurrence (or the predicted frequency of a loss event actually happening).

Another method is the Probability Risk Assessment methodology (D. P. Gilliam, 2003). This method calculates IT security risk using the following formula:

$$R = I * L * F, \text{ where } I=D*T \quad (2)$$

where R is risk; I is impact (i.e., how much impact the threat will have if it does succeed); L is the likelihood, or potential success of, an attack; F is frequency, where F = N/t; D is damage; T is recovery time; t is the specified time period; N is the number of events, where N=E* L* I; and E is how easy it is to originate an attack.

Risk is frequently treated in the literature as a function of the impact (loss incurred) of an undesirable event, the likelihood of this event occurring and how often this event occurs. The risk value represents the potential loss to the business from the occurrence of the unwanted event. Damage depends on the criticality of an IT resource, or data, and the degree of possible distraction, or loss, associated with the resource. The protections against attacks are a set of countermeasures that can reduce the damage if attacks succeed in circumventing the security measures. Recovery time can be defined as the time needed to obtain, or restore, lost data after it is compromised. Likelihood is the possibility of a successful attack, meaning an event which will affect confidentiality, integrity and/or the availability of a resource. Frequency is a function of the number of events occurring over a specified period of time (for example in one year). The impact and likelihood parameters may be represented as quadratic terms, implying a greater impact on risk than on the ease of attack. This method provides a clear basis for risk calculation, but may be difficult to use in terms of web application security; where there is a need to identify and evaluate

the potential for loss due to different threats, to identify the many vulnerabilities associated with each threat, and to identify the many attack methods that might exploit each of these vulnerabilities. There may also be a number of countermeasures, with different associated costs, that could be used to mitigate multiple threats. The tree diagram provided by Vi-Secanto reflects links between these different security objects. It demonstrates that a number of vulnerabilities can be associated with one component. It shows that many attacks might exploit a particular vulnerability. There may also be a number of countermeasures which can minimise each attack.

Figure 6.1 shows an example of a risk assessment and management process model for evaluating information systems security at the enterprise level. It is based on a five stage process (Farahmand et al., 2003).

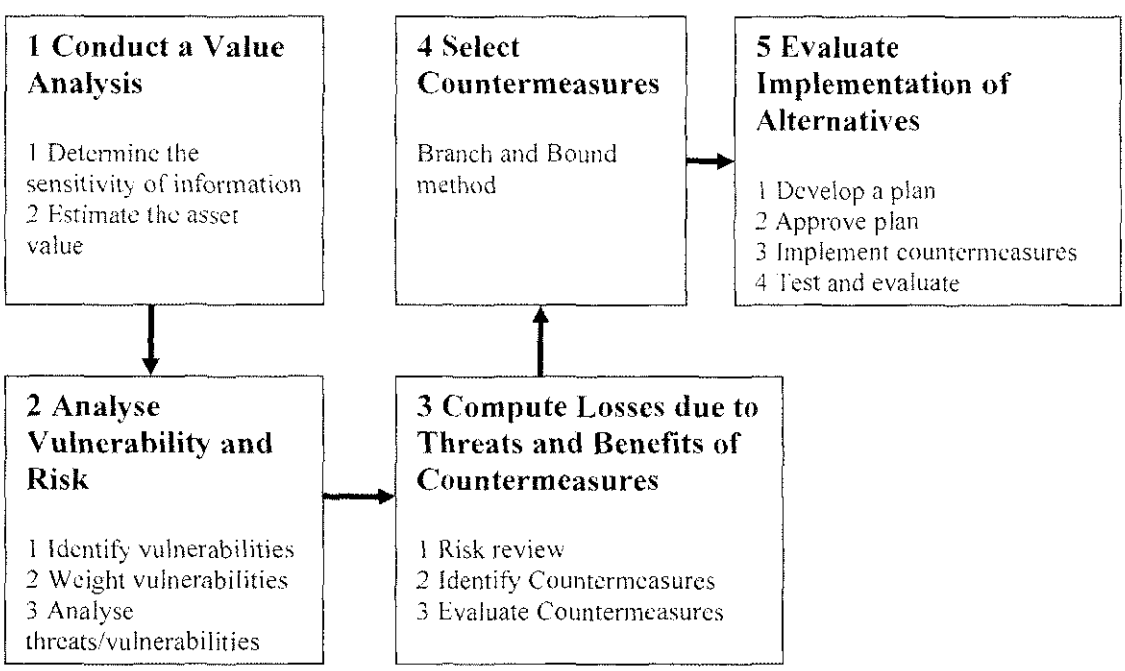


Figure 6.1 A risk assessment model for enterprise security improvement.
Source: (Farahmand et al., 2003)

Stage one, *Value Analysis*, includes an analysis of the system in terms of the sensitivity of the resources involved, the criteria to be used in evaluation, organisational goals and asset values. The outcome of stage one is a description of the system context and its components, with their estimated values being stated. During stage two, *Vulnerability and Risk Analysis*, each system component is

associated with potential threats and is examined for vulnerability to known attacks (unwanted incidents). Identifying risks includes defining the weighting of vulnerability, as well as threat relationships to vulnerabilities. Stage 3 (*Compute Losses due to Threats and Benefits of Countermeasures*) estimates the cost, or impact, of not securing each system component. It uses information on the frequency of attacks and threats, plus the expected benefit of countermeasures. Stage four, *Select Countermeasures*, requires that countermeasures need to be selected in order to achieve an acceptable, minimal level of cost. The Final stage, *Evaluating Implementation of Alternatives*, involves the planning and implementing of countermeasures, as well as testing (Farahmand et al., 2003). Farahmand et al. (2003) proposed using weighting scores for evaluating asset loss (from 1 to 10, where 10 is the most severe). They presented tables with examples of weighting scores for financial loss and intangible damage. One example of intangible damage (*Major stock price impact/bankruptcy)* was given the highest score of ten. The expected cost of an incident can be defined (Farahmand et al., 2003) as:

$$EC = \sum_i (AP_i * EC_i) \quad (3)$$

where EC is the total expected cost of the incidents; AP_i is the assessed probability of the occurrence of incident i; and EC_i is the expected cost for damage caused by incident i. This method may be difficult to apply in web application development due to its requirement for probability values as input data. It can be difficult to estimate these kinds of values for the many different threat types.

There is a similarity between this approach and Vi-Secanto's approach. All these steps are similar to steps which the user of the proposed tool needs to complete. Vi-Secanto also uses weighted scores and the scale in Vi-Secanto uses relative asset value (from 1 to 10, where 10 is the most valuable) to indicate potential loss where a higher value for an asset reflects a greater potential loss for the business. In contrast, Vi-Secanto does not require probability values as input data, it uses relative weights for all input terms.

Another quantitative method (called Security Meter) calculates risk using probability values specified for vulnerabilities, threats and countermeasures (Sahinoglu, 2005).

The Security Meter model is based on a probabilistic decision-tree diagram approach. It is a combined qualitative/quantitative approach, in the sense that relative values such as low, medium and high are converted to a numeric scale. This model can be used by a variety of users, from beginners to experienced security practitioners. It aims to provide a mathematical/statistical foundation for practical risk evaluation (Sahinoglu, 2005). In this model, risk is defined as the possibility of a certain threat being realised by exploiting a particular vulnerability, resulting in a harmful impact on the information system. One of the main components of the risk calculation is the presence of a countermeasure (CM), or the lack of a countermeasure (LCM). Every event, or object, which can reduce risk to the information system, can be defined as a countermeasure. For example, an action, a device, or a procedure can be represented as a CM. When the CM is applied, the risk is reduced, and the remaining risk is defined as residual risk. If a countermeasure can reduce risk completely, the value of the residual risk will be zero.

The Security Meter model takes two types of inputs: Deterministic (constant) values; and probabilistic (random) values. The outputs are the residual risk and the expected cost of the appropriate countermeasures. Probabilistic values are represented as values between 0 and 1 or as percentage values from 0% to 100%. Sums of probabilistic values should be 1, or 100. Probabilistic values (the likelihood of occurrence), are assigned to vulnerabilities and threats. The lack of countermeasures is defined from the existing level of countermeasures, with the sum of both values being equal to 1. For example, if a particular countermeasure provides 100% protection, then the lack of countermeasure will be 0. If a countermeasure has only 70% protection, then the lack of a countermeasure will be 0.3, or 30%. Constant inputs are required and include a measure of system criticality and capital (investment) costs. System criticality is defined as a measure of how critical a system is for the business if it were to be lost. The system criticality depends on residual risk. If residual risk is low, the criticality is also low, since the system can be protected using countermeasures which effectively mitigate risks. Capital cost is defined as the total expected loss to a business in dollar values if the system was to be completely destroyed and could no longer be used.

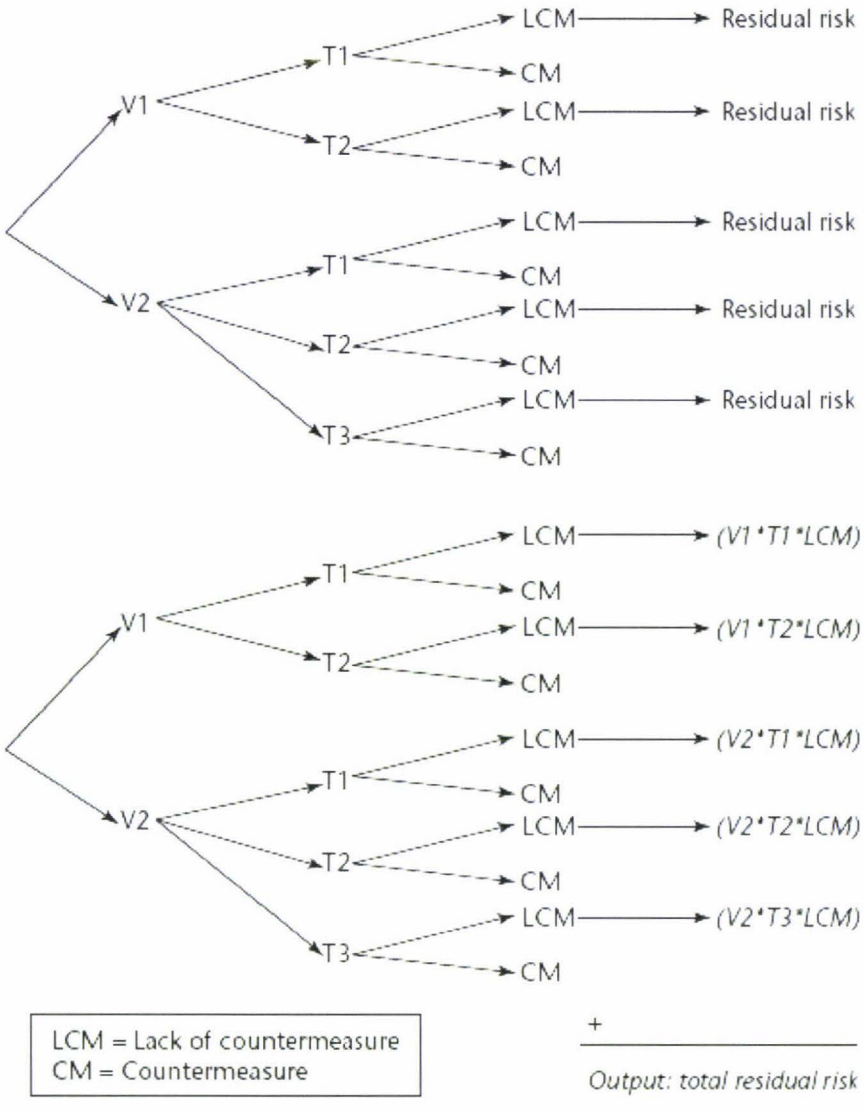


Figure 6.2 Security Meter decision tree diagram.
Source: (Sahinoglu, 2005, p.20.)

The Security Meter model uses a hierarchical tree structure, similar to the one used in Vi-Secanto, as shown in Figure 6.2, to represent the relationships between the information system to be protected (the root node); the weaknesses, or vulnerabilities, identified in the system; the threats associated with each vulnerability; and the use, or non-use, of countermeasures for each threat. The residual risk for each path in Figure 6.2 is calculated using probabilistic inputs and then summed over all vulnerabilities, in order for the system to arrive at the total residual risk (See Equation (4)).

Residual risk = Vulnerability * Threat * LCM

$$R = \sum_j (\sum_i v_i * t_{ij} * lcm_{ij}) \quad (4)$$

where R is the residual risk; v_i is vulnerability i, where i= 1..., n; n is the number of vulnerabilities in one system component; t_{ij} is the jth threat to vulnerability i, where j = 1.... m; m is the number of threats to vulnerability v_i; and lcm_{ij} is the probability of a lack of countermeasure for the threat t_{ij}.

The total residual risk for the example shown in Figure 6.2 is the sum of the five separate residual risk values. The final risk (shown as Equation (5) below) is the product of a criticality factor (a constant input) and the total residual risk. The expected cost of loss (ECL) is the product of the capital investment cost (a constant input) and the final risk (Sahinoglu, 2005).

Final risk = Residual risk * Criticality (5)

ECL = Final risk * Capital cost.

The expected cost of loss (ECL) is a monetary value, which is useful in determining the level of investment in countermeasures required in order to prevent loss, or to recover more quickly in the event of a security breach. Similarly, Vi-Secanto also uses a tree diagram to represent the relationships between security objects for an asset. Vi-Secanto also uses a component criticality parameter, to represent how much a particular component is valued by a business. Another similarity is that all risk calculations are carried along each branch of the tree. The main difference is that Sahinoglu's Security Meter approach requires the input of probability values while Vi-Secanto only requires the use of relative weights (from 1 to 10, where 10 is the highest). Sahinoglu's Security Meter model allows only one countermeasure per threat, but in reality more than one could be used. Vi-Secanto's approach takes into account the possibility of using several countermeasures for one attack/threat.

Sahinoglu's Security Meter approach for calculating a security risk is easy to understand, takes account of imperfect countermeasures, considers multiple threats for specific vulnerabilities and produces a cost estimate to facilitate decision making with respect to investment in countermeasures. The main advantage of this risk assessment technique is the ability to provide a mathematical/statistical foundation for security protection decisions, by basing these decisions on estimated costs. This technique can also incorporate qualitative levels, using terms such as P(high), or P(very low) and so

on, as long as the three rules of probability discussed above are retained when translating these probabilities into numbers. In this case Sahinoglu's Security Meter method can also be classified as a mixed approach between quantitative and qualitative approaches to risk management. The disadvantage of this method is that it is difficult to determine probability values for the security objects and, as a result, it is difficult to use in expressing web application security.

The Vi-Secanto tool has used an extension of Sima's formula (Sima, 2005), (the description of this formula is provided in Chapter 3). The choice to use this formula was based on weighted values being easy to use. These values can be provided by security experts. The known security information can be reused by, and in, other projects for evaluating web applications. The Sima formula was extended in two ways; first, to carry out the calculation based on the tree structure and, second, it is necessary to take countermeasures to account. The following section provides a comparison of the proposed tool with the two existing security assessment tools.

6.2 Comparison of Vi-Secanto against two existing security products

All three tools target existing problems within the area of web application security. The comparison of these tools shows that there are a number of commonalities, as well as some differences (please refer to Table 6.5). The intended common users for all three tools are developers who are not security experts. Computer security consultants are the other targeted user group for the Practical Threat Analysis (PTA) system. Vi-Secanto also targets managers, who may not be security experts, as the other intended users for the proposed tool.

The Microsoft Application Security Threat Analysis & Modelling tool has a top down menu and an easy-to-navigate tree view menu, to allow a presentation of the project components (see Figure 6.3). The Practical Threat Analysis tool has a top down level menu and a link menu (see Figure 6.4). Vi-Secanto has a top down menu and a context menu.

There are differences in the inputs for the three tools. For example, user roles are required for the Microsoft product, but are not required for the other tools. There are different formulas used for the risk calculation in the three different tools, as each of these formulas require different inputs. The differences in the risk calculation will be discussed further at a later point. Other differences for the Microsoft tool, in comparison to the other two tools, are that the project requirements are expressed as use cases.

Table 6.5 *A Comparison of the Proposed Tool with Two Existing Tools*

	Microsoft Threat Analysis & Modelling Tool (Microsoft Corporation, 2006a)	**Practical Threat Analysis (PTA) Tool (PTA Technologies, 2006)**	**Proposed Vi-Secanto Tool**
Intended Users	Developers, not security subject-matter experts	Developers, Computer security consultants	Developers, Managers, not security subject-matter experts
Navigation/ menu	Tree view navigation with visibility to all nodes at all times	Top menu, link menu on the side of main page and at bottom of some pages	Top menu, Context menu, Interactive tree diagram
Input	User roles, assets (text), components (text), external dependencies, threats, attacks, relevancies, use cases, their association	Assets ($), components ($), threats, vulnerabilities, countermeasures, their association	Assets, components, attacks, vulnerabilities, & countermeasures (all rated from 1 to 10), and their associations
Output	**Graphs**: Call flow, Data flow, Trust flow, Threat tree, System call flow, Attack surface	**Graphs**: Top current threats, risk history, analysis history	**Graphs**: risk history, analysis history, interactive tree can be saved as a graph . Each project is visually present with main parameters as an interactive tree with asset structure, including components, associated with vulnerabilities, attacks and countermeasures.
	A risk report with detailed characteristics for risk (accept, reduce, avoid, or transfer)	Max, min and current risk as % of total asset value. Can recalculate the value of the loss ($s) for each component and asset	Risk rating for each component and asset.
	Instruction for development team on what countermeasures need to be done for each component	**Information** can be obtained for a range of countermeasures sorted by cost-effectiveness, implementation cost,	**Information** for countermeasures, for attacks, vulnerabilities for particular project sorted in order of ease

	Microsoft Threat Analysis & Modelling Tool (Microsoft Corporation, 2006a)	Practical Threat Analysis (PTA) Tool (PTA Technologies, 2006)	Proposed Vi-Secanto Tool
	Information can be obtained for objectives, roles, components, external dependencies application use cases, threats and the associations between them **Instruction** for design team (CRUD matrix, components with defined roles, technology data storage, relevancies) **Instruction** for test team, how to test each component against particular attacks **Instruction** for operation team (components and use cases)	overall mitigation level **Information for** mitigation of threats sorted in descending order by their ROSI (Return On Security Investment) value and mitigation steps for target risk reduction **Detailed Information** for threats, assets, vulnerabilities, countermeasures **Information** on top threats ordered by their current risk level	of implementation and those providing better protection **Information** on priority of protection for components sorted by risk **Detailed Information** for attacks, assets vulnerabilities, countermeasures **Information** for top attacks ordered by their current risk level.
Risk calculation for asset components	A risk report with detailed characteristics for risk (accept, reduce, avoid, or transfer)	Threat's Maximal Risk = maximal potential financial damage: Max_Risk = Asset_value/100*Damage. Threat's Minimal Risk = the potential financial damage after all countermeasures are implemented: Min_Risk=Max_Risk-Max_Risk/100*Mitigation	The tool calculates risk using Sima formula 'security risk assessment equation' (Sima, 2005). Risk associated with the asset is calculated on the basis of a decision-tree. See section 3.5 for full description
Security knowledge provided by the tool	Project templates which include: attack library, threats, countermeasures and association between them. New project can be opened from template, templates are not provided but must be created by a user, by saving one of the projects as a template	No templates. Library downloads as text file with lists of assets, vulnerabilities, countermeasures, threats, attack types & tags. Existing project can be saved as a library or as a new project.	Project templates (web application) including components, vulnerability, attacks, countermeasures plus their dependences; Component templates with all dependent security objects; Vulnerability templates with all dependent security objects; Attack templates with all dependent security objects; Countermeasure templates. New project can be

	Microsoft Threat Analysis & Modelling Tool (Microsoft Corporation, 2006a)	Practical Threat Analysis (PTA) Tool (PTA Technologies, 2006)	Proposed Vi-Secanto Tool
			opened from template. Any project can be saved as new template. Any templates can be edited
Information stored	User input, keeps current state of each project	User input, keeps history of countermeasure implementation	User input, keeps current state of project, keeps history of risk reduction

Microsoft's tool has a predefined set of component Service types: *Thick Client*; *Website*; *Web Services*; *Database*; *File Service*; *Browser Active Object*; *Window Service*; *User Control*; *Batch Job*; and *Others*. A Service type represents a category; it can be used to classify the components in a system. For example, a user can define a component as with a service type of *Database*. In the example provided by Microsoft the other provided component has a service type of Website with the description: *This component provides the internet facing entry points.*

In the PTA tool the security entity, called a component in Vi-Secanto, is called an Asset and is defined in terms of its properties. To avoid confusion, it is also called a component in this discussion. For example, in the sample file provided by PTA Technology, one of the components was named the accuracy and integrity of the data in the system database. Components in Vi-Secanto are more similar to the Microsoft definition of components, which is based on their function. All three tools allow the splitting of web applications into finer-grained components, which are defined by the users. Vi-Secanto provides templates for a particular set of predefined components. Examples of pre-defined Vi-Secanto components are: Homepage; Standard Shopping Basket; Online Payment System; and Product Category Page.

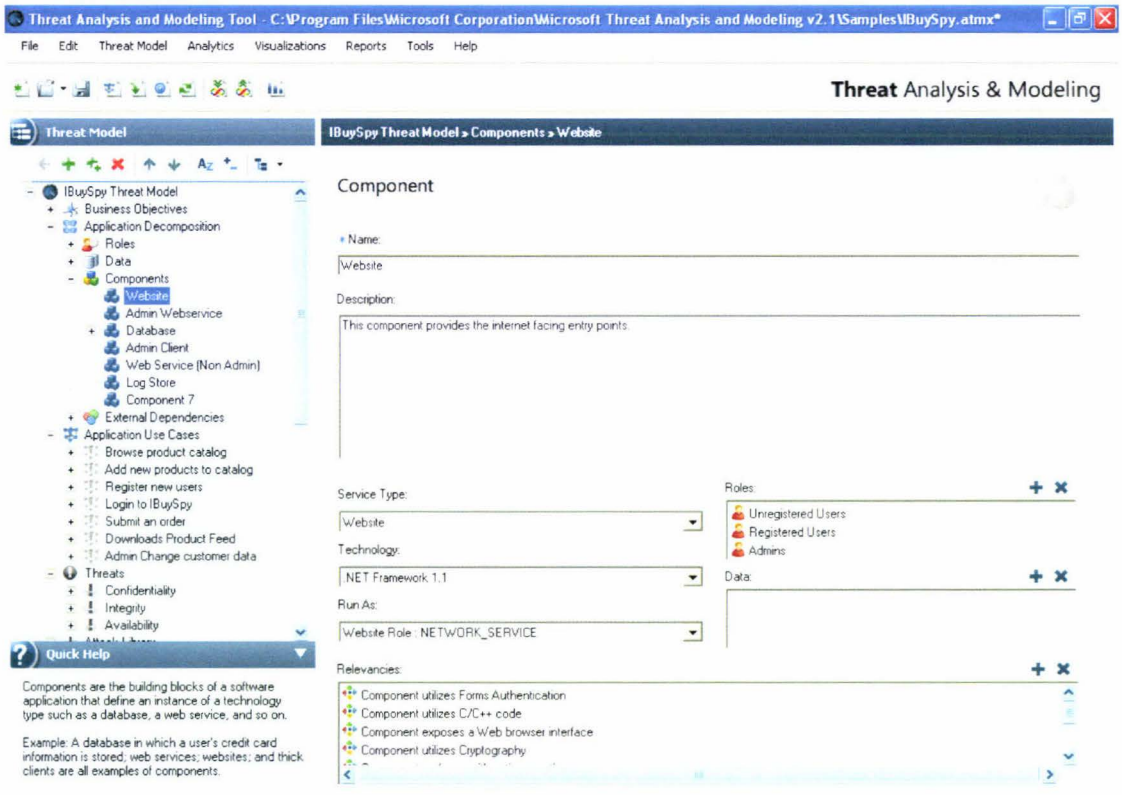


Figure 6.3 Microsoft's Application Security Threat Analysis & Modelling tool has an easy to navigate tree view menu.

Source: (Microsoft Corporation, 2006a)

The aim of Microsoft's Application Security Threat Analysis & Modelling tool (see Figure 6.3 for a screen-shot of the tool's interface) is to provide assistance to users with the creation of a threat model for a web application. The tool is able to automatically identify threats, after the user enters known data, including business requirements and application architecture. The tool requires the user to measure the risk associated with each threat. This helps in understanding how to deal with the risk associated with each threat. The options are, accept, reduce, avoid, or transfer. The Output for the Microsoft tool is a set of reports that provide instructions for each of the different teams; Designers, Developers, Testers, and the Operations team (see Table 6.5).

The Practical Threat Analysis (PTA) tool (see Figure 6.4) aims to assist users with threat modelling, risk assessment and the establishment of a risk reduction policy for the system (PTA Technologies, 2006). Threat modelling helps to analyse and identify threats, maps them to particular assets, and defines countermeasures. Risk assessment

helps to define a cost-effective risk mitigation policy for specific system configurations and functionality. The mitigation plan is used to lower system risk to a minimal, acceptable level. The risk reduction policy includes the mitigation plan with a number of countermeasures identified as being the most effective against a particular threat.

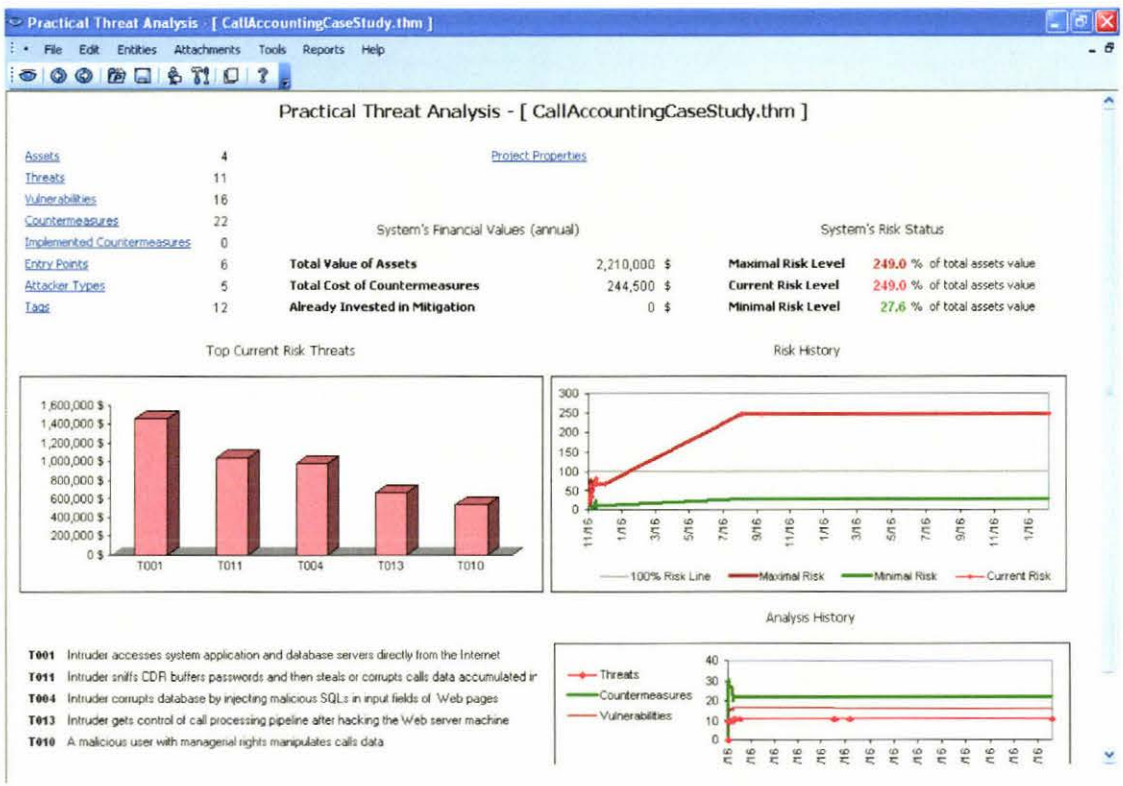


Figure 6.4 Practical Threat Analysis (PTA) shows the system's status.
Source: (PTA Technologies, 2006)

The Threat Analysis & Modelling tool is also able to use information provided by the users to build security artefacts such as access control matrices, data flow and trust flow diagrams. The Output for the PTA tool is a set of reports that provide comprehensive information for the user (for more details please refer to Table 6.5). Figure 6.4 shows the summary page (System status). This is the combined report for the project data. It shows the number of assets, threats, vulnerabilities, countermeasures, implemented countermeasures, entry points, attacker types (the various classes of attackers), and tags (classifications for the model entities) modelled for a project.

The main output of the Vi-Secanto tool is a visual presentation of project data as an interactive tree diagram, which can be saved as an image file. The interactive diagram is the way in which the program interacts with the user. This diagram performs calculations of the project risk. Any changes made by a user in any security object will trigger recalculation of the risk. The old proverb, A picture is worth a thousand words, reflects the ability of a user to more easily absorb images than written text. Vi-Secanto provides the opportunity to view the model of the project as one interactive image. This is advantageous because it allows the user to scan, recognise, and quickly recall, images. It also allows quick visual recognition of changes in size, colour, shape, movement, or texture. An interactive information visualisation system can take advantage of the human ability to compact large amounts of information into a manageable visual space. Text-based information processing can require a larger cognitive effort from humans, when the volume of information is large. Information visualisation, by its fundamental nature, provides relief from cognitive effort for the human perception system (Andrews, 2002; Heer, Card, & Landay, 2005; Johnson & Shneiderman, 1991).

In addition, Vi-Secanto provides output in the form of a list of countermeasures for attacks, or vulnerabilities, for a particular project, sorted in order of ease of implementation and greatest level of protection. The other information outputs for the tools are presented in Table 6.5.

A risk calculation is provided by all three tools. The difference is that the PTA tool expresses risk in terms of a dollar value, while Vi-Secanto expresses risk as a numeric, relative rating value. The Microsoft tool calculates risk from threats and provides a report with detailed characteristics of how to deal with this risk in terms of, acceptance, reduction, avoidance, or transference. There are advantages and disadvantages in both approaches. Risk expressed in dollar terms can be easily understood by a business manager. It is also possible to justify a desired investment for security countermeasures when figures are available for cost-benefit analyses. On the other hand, not all components can be measured in dollars. For example, if customer data is compromised by a malicious attack and the company reputation may suffer as a result, this would be difficult to quantify in dollar terms. The use of relative ratings proposed in Vi-Secanto overcomes some of these issues. Sima's (2005)

formula was adopted and adapted for the risk calculation (see Chapter 3) in Vi-Secanto. Security objects: such as components, attacks, vulnerabilities, and countermeasures: have weighted expressions (from 1 to 10, where 1 is the lowest and 10 is the highest rating). The use of weightings for the components provides simplicity for the manager. The manager needs to decide how important each component is to the business. In the case of a component having less importance for the business, then the value will be lower (e.g., three). In the case of a component being of greater importance the stated value could be nine, or ten. The weighted values for all of the security objects will be defined by security experts and stored in the tool database. Currently, the prototype tool does not contain expert defined values for security objects: but, it has the capability to store them. The advantage of using weighted values (from 1 to 10, with 10 highest) for the vulnerabilities, attacks and countermeasures is that this provides a combination of Quantitative and Qualitative analysis approaches. On the one hand, it is similar to a Qualitative risk assessment, with a finer degree of high, medium, and low, assessment. On the other hand, the final results of the risk calculation are provided as numerical values, which provide all the benefits of a Quantitative analysis approach. These benefits include: results which are easily understood by business managers, access to the independent objective metrics provided by security libraries, and numeric values which allow for comparisons and analysis of different stages of the system development. Vi-Secanto provides the manager with a numeric value of the risk attributed to each component and to the whole system, as a summary of all the components' risk. The numeric value for risk helps the manager to compare one state of the project with another, and provides an opportunity to compare different system components in order to determine which one poses more risk.

The Microsoft tool does not provide a risk calculation for the asset, or the components, probably because it is designed more for the use of developers, than managers. The manager is the user who may need to have final risk figures for the purposes of risk analysis. Instead, this tool provides risk assessment for each particular threat. The risk of a threat ranges from 1 to 9 (see Table 6.6) and is calculated by multiplying the Business impact of a particular threat (1 = Low, 2 = Medium, 3 = High) by the Probability of this threat's likely occurrence (1 = Low, 2 = Medium, 3 = High). The Risk Owner report presents the results for each threat in

terms of how to deal with it: accept; reduce; transfer; or avoid.

Table 6.6 *Risk Values for a Threat in Microsoft's Application Security Threat Analysis & Modelling Tool*

		Business Impact		
		Low	**Medium**	**High**
Probability	**Low**	1	2	3
	Medium	2	4	6
	High	3	6	9

Source: (Microsoft Corporation, 2006a)

The security knowledge provided by the Microsoft tool is presented in an example project file supplied with the tool. The example file consists of an attack library, threats, countermeasures and the association between them. The Microsoft tool provides the option to create a new project based on the supplied template. Currently, only one template is provided by Microsoft's tool, with no other templates yet available. If the user requires more templates, the templates need to be created by the user. The use of templates is also supported by the proposed Vi-Secanto tool. However, provision of templates differs in Vi-Secanto from the provision of a single template in the Microsoft tool in several ways. First of all, the Vi-Secanto tool provides different types of templates for different kinds of web applications. For example; if the user wants to develop Bank applications, they can choose the *Bank template*; and if the user decides to develop an E-Commerce site, it is possible to use the *E-Commerce template* or the *Simple Shop template*. Second, templates are designed at a much finer level below the level of the project. Templates for web applications provided by Vi-Secanto include templates for components, vulnerabilities, attacks and countermeasures, plus their dependencies. As a project is represented as an interactive tree diagram, all of the templates can be viewed as new branches for that tree. The length of the tree branch is dependant on the type of template it originates from. The templates can have different levels. For example, component templates have four levels, which include components, vulnerabilities, attacks and countermeasures. Other kinds of templates provided by the security libraries of Vi-Secanto are shorter. For example, attack templates have only two levels, which include attacks and countermeasures. The template data should be

verified by security experts. The weights used for the security data in the Vi-Secanto prototype have not been verified by security experts at this stage.

The evaluation of the proposed tool shows that users like the ability to use security knowledge in the form of security templates. The PTA tool doesn't have templates. It has libraries in the form of text files with lists of: Assets; vulnerabilities; countermeasures; threats; attacker types; and tags. The PTA tool has the capability to save an existing project as a library, or as a separate project. It is possible for the user to utilise the new library in their next project. These libraries, however, do not provide links between the security objects. When a user works with a project and downloads library information, all of the security objects will be downloaded as a list. The links between the security objects will be lost. This is not convenient for the user if the user wants to add a security object which is linked to other objects. The user needs to establish the links between security objects themselves. The only way to save links between security objects is to save the existing project as a new project, or save the library as a new project. Vi-Secanto provides the opportunity not only to start a project from a project template, but also to add other security information in the form of component templates, vulnerability templates, and attack templates (stored in the tool's security database) to the existing project. For example, the user can add an additional component and the tool will create it as a branch for an existing project tree. This branch will have four levels: The component; its vulnerabilities; attacks; and countermeasures. In this way, the user can save time. The user doesn't need to recreate all of the links. Another example is the situation where a user decides to add a new vulnerability for an existing component. The tool will add a branch with three levels.

Information storage for all three tools is similar, as they all maintain copies of the user's input and the current state of the project. Two tools (the PTA tool and Vi-Secanto) have an additional functionality. They both offer a history of risk reduction. The history of risk reductions is useful to the user, as it demonstrates how countermeasure implementation provides an overall security improvement and provides a comparison with the previous state of the project/s.

The PTA and Microsoft tools provide extensive functionality for the user. There are, however, two limitations with the PTA tool. First, there is no arrangement for security templates. Second, the data is presented in the form of table views. A visual, interactive representation may find greater acceptance. This expanded visual ability allows a person to consume the picture's content much faster than they could attain a similar understanding from a textual representation (Johnson & Shneiderman, 1991). If visual presentation of the security data and its dependencies, as a visual tree, were adopted by customers they would gain a better overall perception of the project's information, in the form of a picture.

The Microsoft tool has a lot of features, but is still focussed primarily on the developer. It misses the manager's needs for overall system risk calculation. It also presents the *overall web* application as one module, without splitting it into its different components. As the evaluation of Vi-Secanto by practitioners suggested, presentation of the web application in a more modular fashion as a number of components, is preferable. This corresponds with the fact that customers are often willing to develop applications in stages, whereby they can implement some features in the first stage of development, move this into production and add additional features in a phased approach.

Vi-Secanto has some advantages; such as a visual tree presentation, and the capability to use templates at a finer grained level than what is found in existing tools. A user of the tool can use templates for the different security objects, such as an overall project, components, vulnerabilities, attacks and countermeasures. As with all research, however, a number of improvements and extensions can be made to the proposed tool (see Chapter 7).

The research found that the web development industry has a requirement for a security evaluation tool, such as the tools discussed here. It has been demonstrated in this thesis that company specialists in a range of areas may gain benefits from using a security analysis tool in web application practice.

6.3 Positive and Negative Results from the Evaluations

Positive comments about Vi-Secanto were made by the participants. The participants understood the purpose of the tool, found it useful for their company and have a positive view on its viability in the future. All participants quickly became familiar with the tool. For all of those involved, it took less than an hour to understand the tool's capabilities. This demonstrated that the Usability requirements were met in terms of understandability and learnability of the software. Another usability characteristic, likeability (Kotonya & Sommerville, 1998), a measure of user satisfaction with the tool, was also met. In addition, the proposed method of risk calculation was found to be understandable and easy to use. The risk calculation formula; using relative weights (from 1-10) for each component and for the other security entities (vulnerability, attacks and countermeasures); was seen to be a practical input for computing risk. The use of security templates; as a method to represent security knowledge to non-security experts; was seen as beneficial by the survey respondents.

Chapter 7: Conclusion and Future Work

This thesis has designed and prototyped a software support tool, Vi-Secanto, to assess web application security. Current web application development practice does not generally undertake consistent steps to ensure that security factors are included in the development process. Instead, security is often treated as a standalone activity. Most web application developers carry out development without properly understanding security issues associated with existing vulnerabilities and the possibility of attacks. They need security information to be easily accessible. Managers also need to have the opportunity to perform risk assessment for their web applications. This research was driven by the motivation to make security information accessible for non-security experts, such as managers and developers.

This research contributes to the improvement of web application security in several ways. First, it designed and developed a computer-based tool that makes it easier for web application developers to build more secure software, and for managers to assess the risk and potential loss of not reducing these vulnerabilities. Second, this research proposed a modification of Sima's (2005) risk calculation formula in order to include a relative countermeasure rating (from 1 to 10; where 1 = least protective and 10 = full protection) to represent the level of protection provided by implementing a particular set of countermeasures.

The proposed tool was designed, developed and evaluated in the form of a highly functional prototype. The prototype was tested and evaluated by web application development companies with expertise in security problems which are specific to web applications. These evaluations provide support for the utility of the proposed tool in terms of meeting its stated objectives. Specifically, the tool was perceived as being helpful to developers in determining security requirements, by providing a database of security information, and by providing templates and proposed countermeasures making this information more accessible. The stored security information identifies the components, vulnerabilities, attacks and countermeasures, and the associations between them. These field evaluations also demonstrate the value of the proposed tool in supporting managers in risk assessment, by helping to calculate the residual risk

resulting from not investing in the implementation of countermeasures. This tool can also be useful to managers in providing them with a view of the history of the risk reduction process. Lastly, the feedback from the field evaluations was used to improve the design of the tool and to suggest future avenues of research to extend the tool, the subject of the next part of this chapter.

There are a number of limitations to this research, which may be addressed by future research. As it was a prototype version of the tool and the security database was not complete, it was not possible to test its use with real projects. New research can be carried out as case studies in real world companies, in order to determine how companies might use this tool in reality. It will be good to determine how the tool might be used to support different stages in the life cycle of a web application. One of the limitations was that the tool didn't undergo hands-on task-based evaluation. Future research is needed to discover what advantages companies can gain by using this tool in practice. Other research could focus on how a developer's own security knowledge improves through using this tool, compared with the security knowledge of developers who do not use it. New suggestions for design changes will come from users who interact with the tool over a long time period, and this can also become the basis of future research. Other design changes will be needed in order to link the tool's abilities to acceptable security standards (CESG, 2007; OWASP Foundation, 2005) which may improve the tool's ability to support standards-based modern secure systems design and development practice. Future research could be based on improvements to the design, such as adopting the capability to express project requirements as use cases.

Improvements and extensions which can be added to the security tool include the following.

- Implementation of import/export functions to keep the security information up to date.
- The stored security knowledge in the production version will need to be reviewed periodically by a security specialist team.
- Saving of the last user's project tree view so it loads on start up.
- The functionality of "auto save" for the user project is recommended for the

production version of the tool.

- Improvements in the property panel.
- The facility to have ad hoc, customised reports.
- A horizontal layout for the project tree presentation.
- Making it a multi-user application.
- Relate existing international web application security standards to the security data stored in the tool (i.e. compliance function).

The above limitations in the prototype suggest a number of avenues for further research. Due to the restricted time frame of this research, the import/export functions to automate security information updates were not developed. It would be a significant benefit for practitioners to have the ability to extract data from the security database and transform it into different formats such as XML, Excel and comma delimited text files. A data import function is also an important future addition, as it would allow for downloading new security information on vulnerabilities, attacks and countermeasures from other available sources in order to update the tool's existing database. As a number of different sources of security data are available, it is important to provide an export function for each of the sources, due to the existence of different formats and a different number of fields for similar entities. For example, a vulnerability record held by OWASP may have a different number of fields compared with a vulnerability recorded by SANs.

There is also a need to have highly qualified security specialists review the security information. Security specialists' advice is desirable when deciding on how to weight the properties of vulnerabilities, attacks and countermeasures. It will be a significant advantage in the future to have a security specialist team involved in the process of auditing and verifying the stored security data. If this tool were to be supported by a security specialist team, a large number of small companies could benefit from having reliable security analysis carried out on the systems they are having developed, without physically having to employ a permanent security specialist. In this case web application development communities would pay for one centralised security specialist team, which would spread the cost over a large number of tool users. A security specialist's advice regarding the relative weightings is desirable. Each

company will rely on this for the base values. The tool will, however, provide the possibility to carry out changes in these values, according to the company needs, and based on how critical the affected asset is.

A collapsing branches method can be extended to retain the last user's view in memory. This would show how the same user previously viewed the tree and allow for better use of the screen space. For example, a user can work with one branch *Customer Page,* while other branches are kept collapsed. Then, the next time the same user opens that project, the program will show the user's last screen view. The tool will show one branch *Customer Page,* with the other branches remaining collapsed.

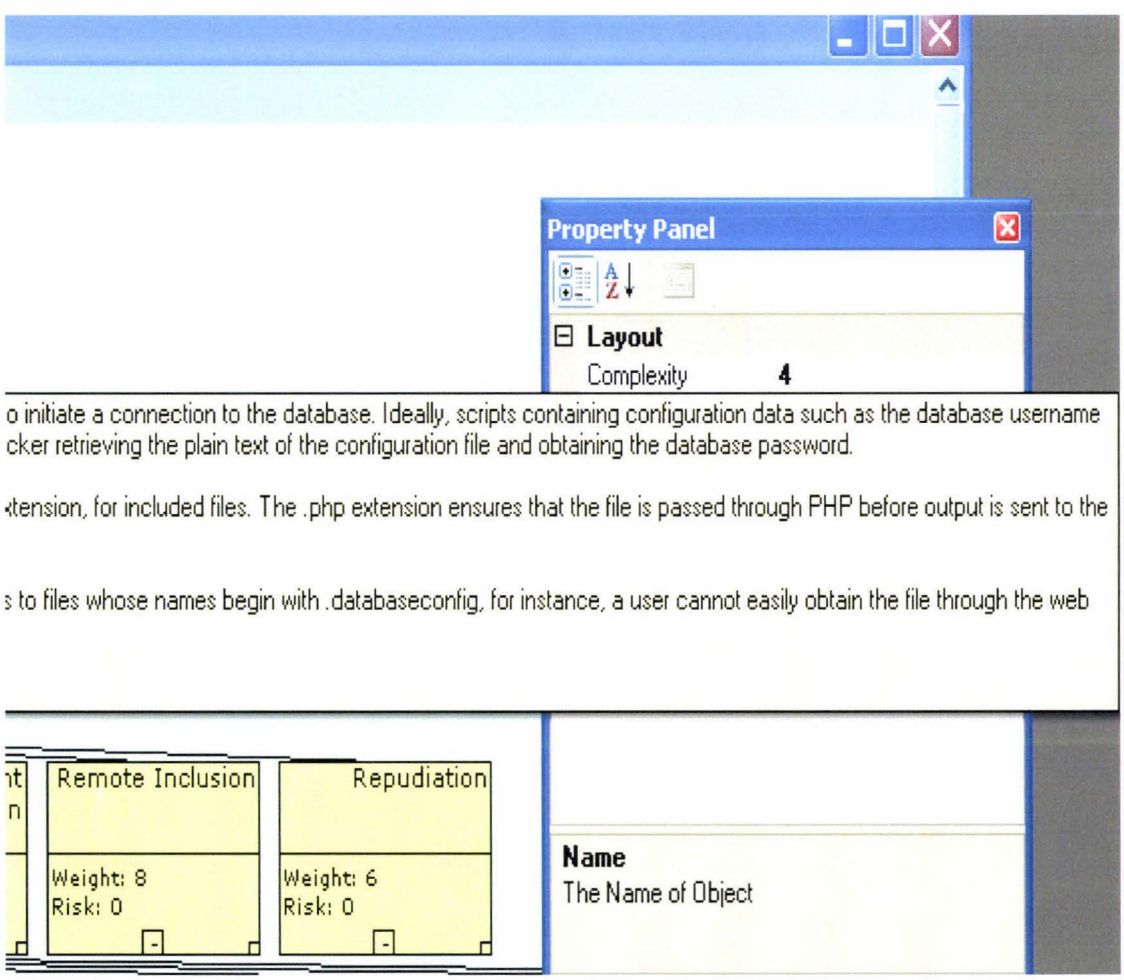


Figure 7.1 Vi-Secanto Property Window with bid description text.

There is a need to improve the property panel. If a description is very large, it is hard

to read and change (See Figure 7.1). Additional functionality can be implemented for the property window. The user should have the ability to set the width and height for the fields being presented in the property window of the data presentation.

There is a need to produce ad hoc, customised reports, which would vary depending on the user's specifications. A user should be able to query, or search for, specific security data and customise presentation details. The application should be able to produce reports in different, commonly used formats, such as HTML and PDF.

The project tree currently only has a vertical layout. A horizontal layout would also be useful, as different users can have different preferences for tree viewing. A horizontal layout may be preferred by some users due to the fact that users usually work with windows applications and all files and directories are presented in a horizontal layout.

The production version will need to be a multi-user application. This is important for companies in allowing the same project to be viewed by different people, who have different roles in the company. For example, a manager can view the project from a risk calculation perspective and determine which component is the most critical for their system. Developers can view the same project to obtain the advice needed regarding countermeasures for a particular system component.

Technical web security standards (e.g. WASS – Web Application Security Standards project) are important and companies may have requirements from some clients to comply with various standards. In order to make the Vi-Secanto tool more attractive for use by these practitioners, there is a need to support checking for compliance with standards by adding this functionality. However, at the time of this study, major efforts such as WASS, part of the OWASP project, were still in progress, making full tool support somewhat premature. The WASS Security Frame contains thirteen principles considered to be minimum security requirements (OWASP Foundation, 2007).

During the implementation, testing and evaluation of this prototype certain questions were raised which were not vital to this particular study, but would certainly need to be addressed in the future. One of these was concerning the future tool development

and security database maintenance. Advice was offered that the production stage of this security tool be put on open source. This recommendation was heavily supported by a number of security experts from the financial corporation who participated in two of the evaluations. Advantages of making the tool available on open source would include: every web application developer will have access to the security knowledge database which would be kept current by using updates provided by the tool's support organisation.

The evaluation of the tool prototype shows that the idea of a security evaluation tool is perceived as useful. The feedback from security specialists demonstrated that there is a strong interest in such a tool within the Web Application development industry. In addition, the managers and developers who evaluated the tool were willing to adopt such a tool in the future. Furthermore, Microsoft, a recognised leading software company, developed a similar tool and released it during the timeframe of this study. This provides additional support for further work on the design and development of this type of software-based decision-support tool.

The main concern uncovered in the feedback was the maintenance of up-to-date security information in the database. This could be achieved by placing this product in the open source domain, or passing it over to a company with the resources to provide future development, maintenance and security database updates. This research has focussed on a proactive approach to web application security and can be extended to provide a proactive approach to software development in any field. There is a possibility to use Vi-Secanto in other types of non-web software development. All that is needed is to organise a security database for other uses, which means that the basic framework can be populated with different content (i.e., types of vulnerabilities, attacks, etc.), in order to suit different contexts.

Improving security for web sites is a very important issue today; the proposed security tool will be a useful first step for any web application development team or company wanting to improve web application security by improving their ability to assess risk and identify appropriate countermeasures during the early stages of web application development.

Glossary

Artefact	An artefact is a human-made object (Hevner et al., 2004).
Asset	An asset is a valuable entity for business. It can be information, customer data, or even a company's reputation (Microsoft Corporation, 2006c).
Attack (or exploit)	An attack is an intelligent action or exploit with a deliberate attempt taken that uses one or more vulnerabilities to realize a threat (Microsoft Corporation, 2006c; Network Working Group, 2000).
Countermeasure	Countermeasures address vulnerabilities to reduce the impact of threats. Improving application design is an example (Microsoft Corporation, 2006c).
GUI	Graphical User Interface (GUI) - A GUI can be defined as the visual interface area, which can display the output of and allow input for the simultaneous running of computer processes (Microsoft Corporation, 2007).
Requirement	A requirement is defined as what a system must do; it describes a necessary attribute, capability, characteristic, or quality of it in order to provide value for the user (Sommerville, 2007).
Risk	Risk can be viewed as a function of the likelihood that a threat will materialise, the level of vulnerability and the potential for loss of resources.
Threat	A threat is an undesired incident. An occurrence of a threat can damage a business asset. It may or may not be malicious in nature (Microsoft Corporation, 2006c). A threat is defined as "a potential occurrence, malicious or otherwise, that can have an undesirable effect on the assets or resources associated with a computer system" (Amoroso, 1994, p. 2).
Vulnerability	Vulnerability is a flaw in a system that makes an exploit possible. Vulnerabilities can occur in part of or in some feature of a system. Vulnerabilities can exist in the network, host, or application levels of a system (Microsoft Corporation, 2006c). Amoroso (1994) defines "vulnerability" as a characteristic of the software that makes it possible for a threat to occur. A web application vulnerability can be exploited by actions known as attacks which allow threats to be realised.

References

Alberts, C., & Dorofee, A. (2001). *An Introduction to the OCTAVE Method.* Retrieved 20 May, 2007, from http://www.cert.org/octave/methodintro.html

Amoroso, E. G. (1994). *Fundamentals of Computer Security Technology*. Englewood Cliffs, N.J.: Prentice-Hall PTR.

Andrews, K. (2002). *Visualising Information Structures: Aspects of Information Visualisation.* Graz University of Technology, Graz.

Balzarotti, D., Monga, M., & Sicari, S. (2005, September). *Assessing the Risk of Using Vulnerable Components.* Paper presented at the First Workshop on Quality of Protection, Milan, Italy.

Barnum, S., & McGraw, G. (2005). Knowledge for Software Security. *IEEE Security & Privacy Magazine, 3*(2), 74-78.

Bernstein, A. (2005). *So what is a (Diploma) Thesis? A few thoughts for first-timers.* Retrieved 20 May, 2007, from http://www.ifi.uzh.ch/ddis/fileadmin/theses/general/WHATISATHESIS_V1_0.pdf

Boehm, B. W. (1991). Software Risk Management: Principles and Practices. *IEEE Software, 8*(1), 32-41.

Buehrer, G. T., Weide, B. W., & Sivilotti, P. A. G. (2005). *Using Parse Tree Validation to Prevent SQL Injection Attacks.* Paper presented at the 5th International Workshop on Software Engineering and Middleware, Lisbon, Portugal.

Burns, S. F. (2005). *Threat Modelling: A Process to Ensure Application Security.* Retrieved 10 November, 2005, from http://www.sans.org/rr/whitepapers/securecode/1646.php

CERT Coordination Center. (2005). *CERT/CC Statistics 1988-2005.* Retrieved 14 May, 2005, from http://www.cert.org/

CESG. (2007). *Common Criteria and ITSEC.* Retrieved 1 March, 2007, from http://www.cesg.gov.uk/site/iacs/index.cfm?menuSelected=1&displayPage=1

Chinotec Technologies. (2007). *Paros.* Retrieved 11 June, 2007, from http://www.parosproxy.org/index.shtml

Cohen, M. (2005). *Web Application Security: More Budget Needed.* Retrieved 21

August, 2006, from http://www.ebcvg.com/articles.php?id=952

Deloitte Touche Tohmatsu. (2006). *2006 Global Security Survey*. Retrieved 20 March, 2007, from http://www.deloitte.com/dtt/cda/doc/content/us_fsi_150606globalsecuritysurvey(1).pdf#search=%222006%20Deloitte%20Global%20Security%20Survey%22

Faisst, U., & Prokein, O. (2005). *An Optimization Model for the Management of Security Risks in Banking Companies.* Paper presented at the Seventh IEEE International Conference on E-Commerce Technology (CEC'05).

Farahmand, F., Navathe, S. B., Enslow, P. H., & Sharp, G. P. (2003). *Managing Vulnerabilities of Information Systems to Security Incidents.* Paper presented at the 5th International Conference on Electronic Commerce, Pittsburgh, Pennsylvania.

Feather, M. S., Sigal, B., Cornford, S. L., & Hutchinson, P. (2001). *Incorporating Cost-Benefit Analyses into Software Assurance Planning.* Paper presented at the 26th Annual NASA Goddard Software Engineering Workshop.

Fletcher, S. K., Halbgewachs, R., Jansma, R. M., Murphy, M. D., Lim, J. J., & Wyss, G. D. (1995). *Software System Risk Management and Assurance.* Paper presented at the New Security Paradigms Workshop.

Foundstone. (2007). *Web Application Penetration Assessment*. Retrieved 10 June, 2007, from http://www.foundstone.com/us/services-web-appl-penetration.asp

Geer, D., Jr., Hoo, K. S., & Jaquith, A. (2003). Information Security: Why the Future Belongs to the Quants. *IEEE Security & Privacy Magazine, 1*(4), 24-32.

Gegick, M., & Williams, L. (2005). *Matching Attack Patterns to Security Vulnerabilities in Software-Intensive System Designs.* Paper presented at the 2005 Workshop on Software Engineering for Secure Systems—Building Trustworthy Applications, St. Louis, Missouri.

Gilliam, D. P. (2003, November). *Managing Information Technology Security Risk.* Paper presented at the International Symposium On Software Security, Tokyo, Japan.

Gilliam, D. P. (2004). *Security Risks: Management and Mitigation in the Software Life Cycle.* Paper presented at the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2004).

Gregg, D., Kulkarni, U., & Vinze, A. (2001). Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems. *Information Systems Frontiers, 3*(2), 169-183.

Hamdi, M., & Boudriga, N. (2005). Computer and Network Security Risk Management: Theory, Challenges, and Countermeasures. *International Journal of Communication Systems, 18*(8), 763-793.

Heer, J., Card, S. K., & Landay, J. A. (2005). *prefuse: A Toolkit for Interactive Information Visualization.* Paper presented at the SIGCHI conference on Human factors in computing systems, Portland, Oregon, USA.

Hevner, A. R., & March, S. T. (2003). The Information Systems Research Cycle. *Computer, 36*(11), 111-113.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information System Research. *MIS Quarterly, 28*(1), 75-105.

Holz, T., Marechal, S., & Raynal, F. (2006). New Threats and Attacks on the World Wide Web. *IEEE Security and Privacy Magazine, 4*(2), 72-75.

Howard, M. (2004). Building More Secure Software with Improved Development Processes. *IEEE Security & Privacy Magazine, 2*(6), 63-65.

Huang, Y. W., Tsai, C. H., Lee, D. T., & Kuo, S. Y. (2004, November 02-05). *Non-Detrimental Web Application Security Scanning.* Paper presented at the 15th International Symposium on Software Reliability Engineering (ISSRE'04).

Huang, Y. W., Yu, F., Hang, C., Tsai, C. H., Lee, D. T., & Kuo, S. Y. (2004, May 17-22). *Securing Web Application Code by Static Analysis and Runtime Protection.* Paper presented at the Thirteenth International World Wide Web Conference (WWW2004), New York.

International Organisation for Standardization. (2001). *ISO/IEC 9126-1:2001.* Retrieved 10 March, 2007, from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749

Internet Storm Center. (2005). *Country Report.* Retrieved 10 April, 2005, from http://isc.sans.org/reports.php

Johnson, B., & Shneiderman, B. (1991). *Tree-maps: a Space-filling Approach to the Visualization of Hierarchical Information Structures.* Paper presented at the IEEE Conference on Visualization (Visualization '91).

Kavado Inc. (2005). *ScanDo Web Application Scanner.* Retrieved 10 October, 2005,

from http://www.kavado.com/products/scando.asp

Kotonya, G., & Sommerville, I. (1998). *Requirements Engineering Processes and Techniques*. Chichester, New York, Weinheim, Brisbane, Singapore, Toronto: John Wiley & Sons.

Kotulica, A. G., & Clark, J. G. (2004). Why There aren't More Information Security Research Studies. *Information & Management, 41*, 597-607.

Leffingwell, D., & Wigrig, D. (2003). *Managing Software Requirements: A Use Case Approach* (Second ed.). Boston, San Francisco, New York, Toronto, Montreal, London: Addison-Wesley.

Lin, H. X., Choong, Y.-Y., & Salvendy, G. (1997). A Proposed Index of Usability: a Method for Comparing the Relative Usability of Different Software Systems. *Behaviour & Information Technology, 16*(4), 267-277.

March, S. T., & Smith, G. F. (1995). Design and Natural Science Research on Information Technology. *Decision Support Systems, 15*(4), 251-266.

McClure, R. A., & Kruger, I. H. (2005). *SQL DOM: Compile Time Checking of Dynamic SQL Statements*. Paper presented at the 27th international Conference on Software Engineering, St. Louis, MO, USA.

McGraw, G. (2004). Software Security. *IEEE Security & Privacy Magazine, 2*(2), 80-83.

Meier, J. D. (2006). Web Application Security Engineering. *IEEE Security & Privacy Magazine, 4*(4), 16-24.

Melbourne, J., & Jorm, D. (2003). *Penetration Testing for Web Applications*. Retrieved 2 March, 2007, from http://www.securityfocus.com/infocus/1704

Microsoft Corporation. (2007). *Glossary*. Retrieved 11 May, 2007, from http://support.microsoft.com/default.aspx?scid=%2Fsupport%2Fglossary%2Fg.asp

Microsoft Corporation. (2006a). Microsoft Threat Analysis and Modeling Tool (Version v2.1.1): Microsoft Download Center.

Microsoft Corporation. (2006b). *The Security Risk Management Guide*. Retrieved 1 September, 2006, from http://www.microsoft.com/downloads/details.aspx?familyid=C782B6D3-28C5-4DDA-A168-3E4422645459&displaylang=en

Microsoft Corporation. (2006c). *Threat Modeling Web Applications*. Retrieved 2 September 2006, 2006, from http://msdn2.microsoft.com/en-

us/library/ms978516.aspx

Moore, A. P., Ellison, R. J., & Linger, R. C. (2001, March). *Attack Modeling for Information Security and Survivability*. Retrieved 2 April, 2006, from http://www.bauernberger.com/docs/01tn001.pdf.

Netcraft Ltd. (2007). *May 2007 Web Server Survey*. Retrieved 20 May, 2007, from http://news.netcraft.com/archives/2007/05/01/may_2007_web_server_survey.html

Network Working Group. (2000). *RFC 2828 - Internet Security Glossary*. Retrieved 5 November, 2007, from http://www.ietf.org/rfc/rfc2828.txt

OWASP Foundation. (2005a). *Open Web Application Security Project (OWASP)*. Retrieved 11 May, 2005, from http://www.owasp.org/documentation/topten/commentary.html

OWASP Foundation. (2005b). *OWASP- Open Web Application Security Project. Top Ten Most Critical Web Application Security Vulnerabilities*.

OWASP Foundation. (2005c). *OWASP- Open Web Application Security Project. A Guide to Building Secure Web Applications and Web Services*.

OWASP Foundation. (2007). *OWASP- Open Web Application Security Project. WASS Project*. Retrieved 20 August 2007, 2007, from http://www.owasp.org/index.php/Category:OWASP_WASS_Project

Paddon, M. (2000). *The Art of Keeping Secrets or Aspects of Good Information Security Policy*. Retrieved 29 August, 2006, from http://webct-ce.massey.ac.nz/SCRIPT/157738_0602_ALBN_I/scripts/serve_home

PTA Technologies. (2006). Practical Threat Analisys (Version 1.53). Tel-Aviv, Israel: PTA Technologies Ltd.

Sahinoglu, M. (2005). Security Meter: a Practical Decision-Tree Model to Quantify Risk. *IEEE Security & Privacy Magazine, 3*(3), 18-24.

Salter, C., Saydjari, O. S., Schneier, B., & Wallner, J. (1998). *Toward a Secure System Engineering Methodology.* Paper presented at the Workshop on New Security Paradigms, Charlottesville, Virginia, United States.

Schneier, B. (1999). Attack Trees. *Dr. Dobb's Journal Software Tools, 24*(12), 21-29.

Schneier, B. (2004). Security and Compliance. *IEEE Security & Privacy, 2*(3), 96.

Scott, D., & Sharp, R. (2003). Specifying and Enforcing Application-Level Web Security Policies. *IEEE Transactions on Knowledge and Data Engineering, 15*(4), 771-783.

Security-Assessment.com. (2007). *Secure Web Applications*. Retrieved 10 March. 2007. from http://www.security-assessment.com/

Sima. C. (2005). *Security Risk Assessment and Management in Web Application Security*. Retrieved 7 February. 2007. from http://www.devcity.net/Articles/187/1/article.aspx

SoftLogica. (2007). *WAPT 5.0*. Retrieved 10 June. 2007. from http://www.loadtestingtool.com/

Sommerville. I. (2007). *Software engineering* (Eight ed.). Harlow. England. London. New York. Boston. San Francisco. Tokyo: Addison-Wesley.

SPI Dynamics Ltd. (2007). *WebInspect 7*. Retrieved 10 June. 2007. from http://www.spidynamics.com/products/webinspect/

Standards Australia and New Zealand. (1999). *AS/NZS 4360: Risk Management*: Standards Australia and New Zealand.

Steven. J. (2006). Adopting an Enterprise Software Security Framework. *IEEE Security & Privacy Magazine. 4*(2). 84-87.

Symantec. (2007). *Internet Security Threat Report*. Retrieved 10 March. 2007. from http://www.symantec.com/enterprise/theme.jsp?themeid=threatreport

Symantec Corporation. (2007). *IT Risk Management Report*. Retrieved 15 July. 2007. from http://www.symantec.com/enterprise/theme.jsp?themeid=itrisk_report

Thompson. H. H. (2005). Application Penetration Testing. *IEEE Security & Privacy Magazine. 3*(1). 66-69.

Urrego-Giraldo. G. (2004). *Reasoning Nonfunctional Goals and Features in Web Systems*. Paper presented at the International Conference on Information and Communication Technologies: From Theory to Applications..

Vaishnavi. V.. & Kuechler. B. (2005). *Design Research in Information Systems*. Retrieved 14 March. 2006. from http://www.isworld.org/Researchdesign/drisISworld.htm

Verdon. D.. & McGraw. G. (2004). Risk Analysis in Software Design. *IEEE Security & Privacy Magazine. 2*(4). 79-84.

Verton. D. (2002. February 4). Airline Web Sites Seen as Riddled with Security Holes. *Computerworld*.

Watchfire-Corporation. (2006). *AppScan 6.0 Technical Overview*. Retrieved 25 January. 2006. from http://www.watchfire.com/securityzone/default.aspx

Web Application Security Consortium. (2005). *Web Security Threat Classification* -

Classes of Attack. Retrieved 10 March, 2007, from http://www.webappsec.org/projects/threat/

Zelkowitz, M., & Wallace, D. (1998). Experimental Models for Validating Technology. *IEEE Computer, 31*(5), 23-31.

Zhou, B. (2002). *Security Analysis and the DSM Model.* Paper presented at the 13th International Workshop on Database and Expert Systems Applications.

Appendix A: Screen shots

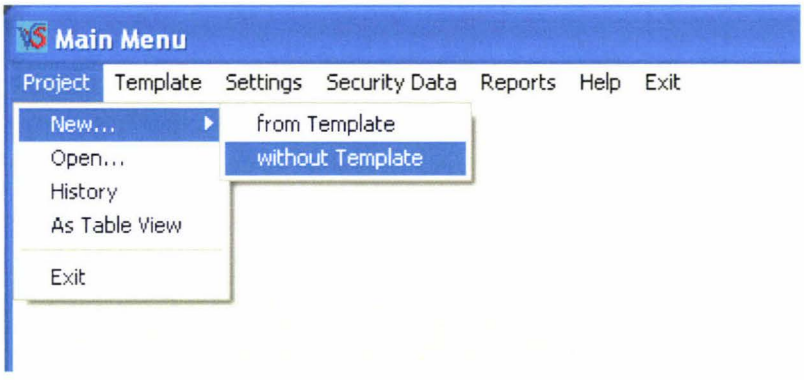


Figure 1. Screen shot of menu navigation for the functionality "Project/ New/ without Template".

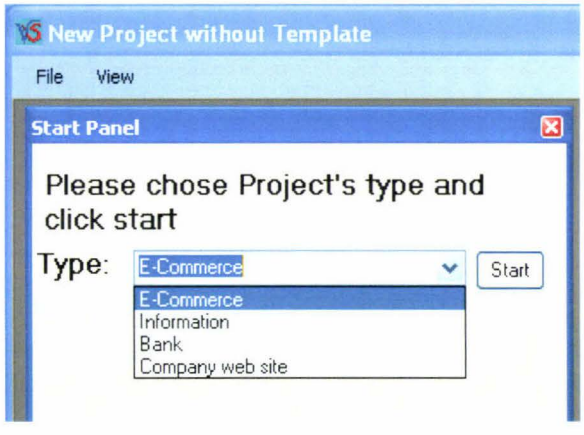


Figure 2. Screen shot of the functionality "Project/ New/ without Template"; a user selects a type of project from a drop down list.

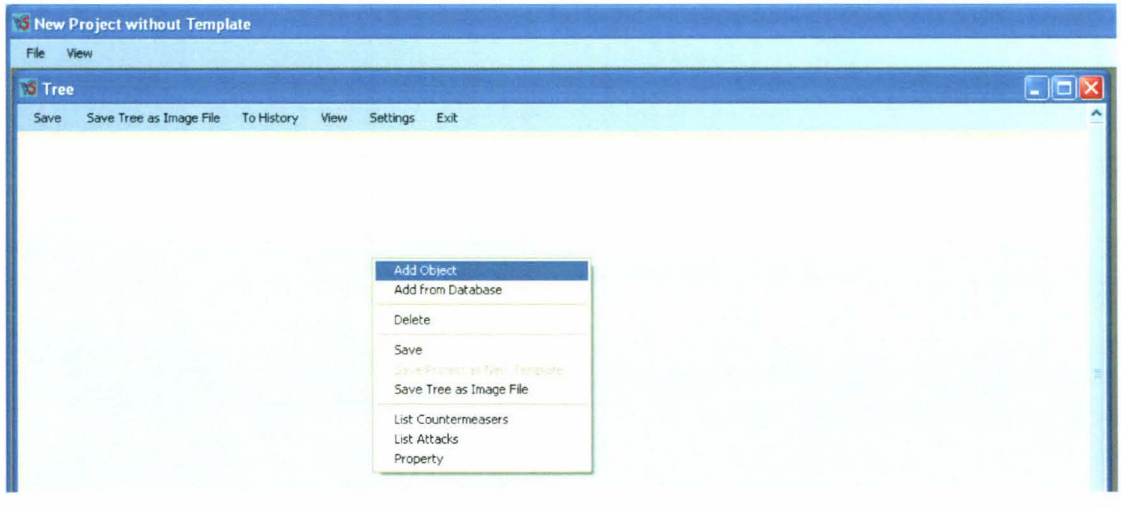


Figure 3. Screen shot of the functionality "Project/ New/ without Template" where a

user is choosing to add a new object to start building a tree for an asset.

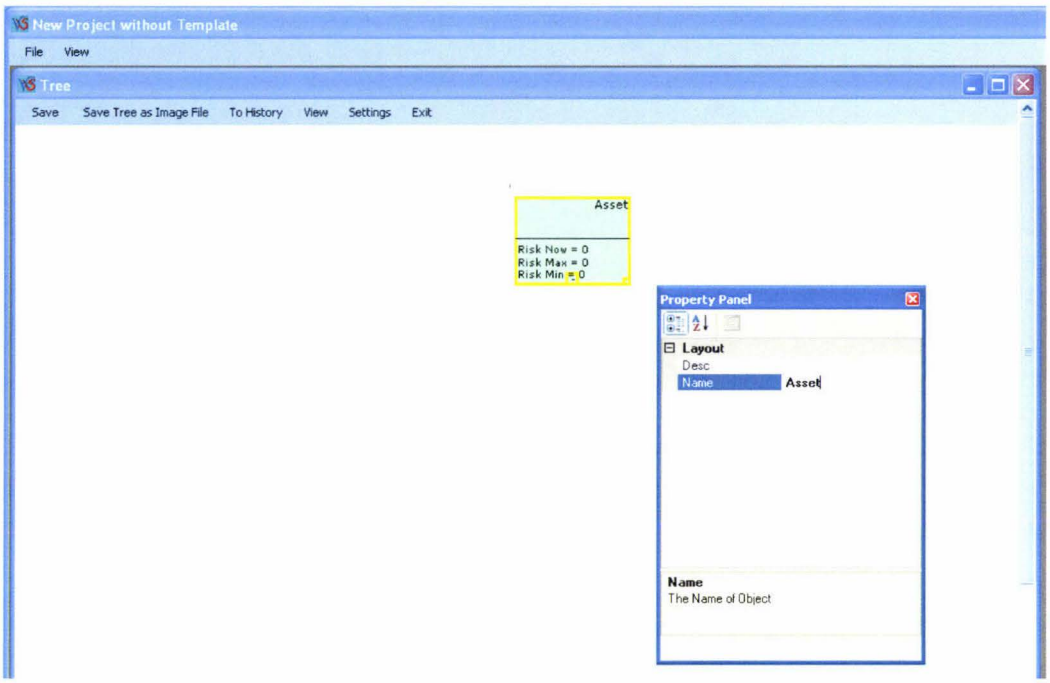


Figure 4. Screen shot of the functionality "Project/ New/ without Template" showing how a user starts with an asset and completes a *Property Panel* to describe it.

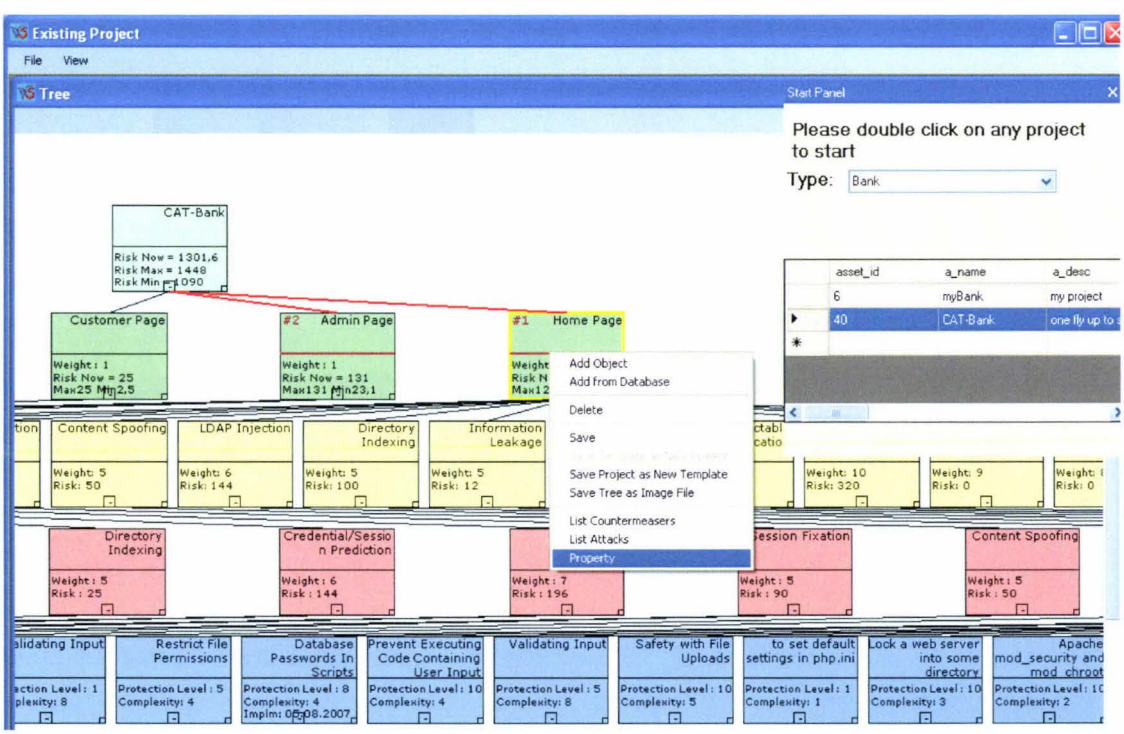


Figure 5. Screen shot of the functionality "Project/ Open" where a user has opened an existing project and then chooses *Property* to view and edit a selected object's

properties.

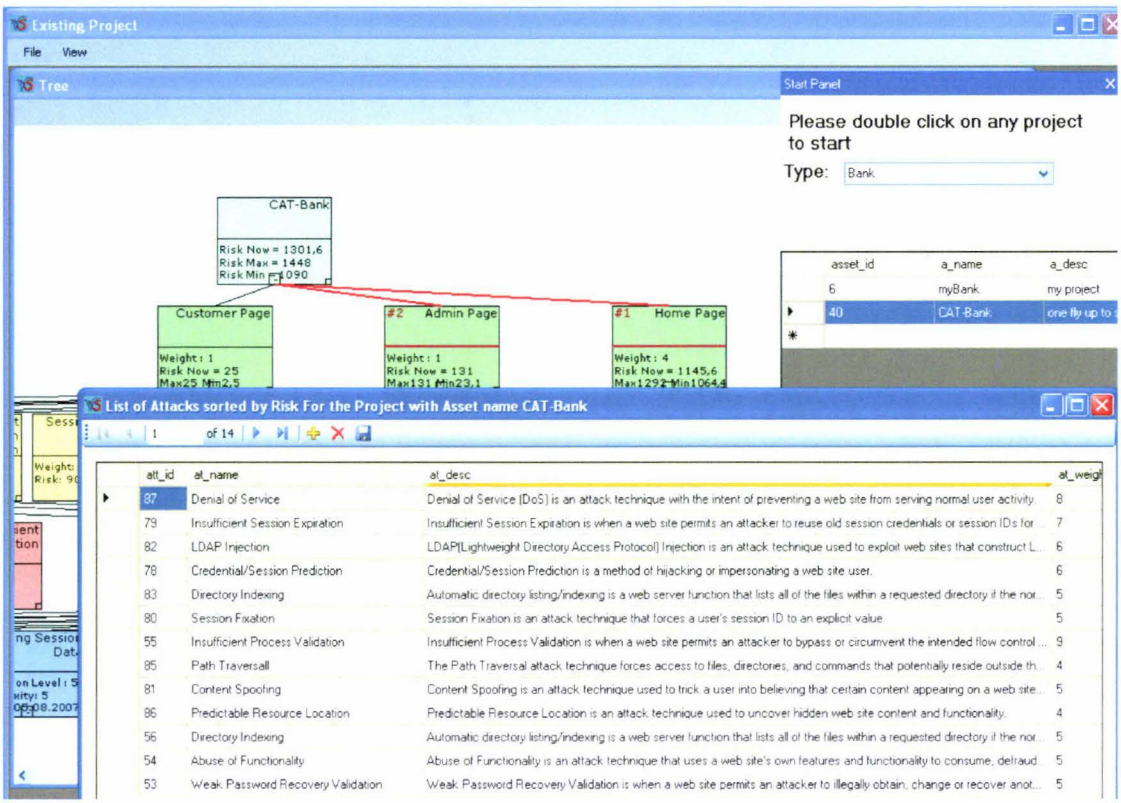


Figure 6. Screen shot of a "List of Attacks sorted by Risk" for the open project.

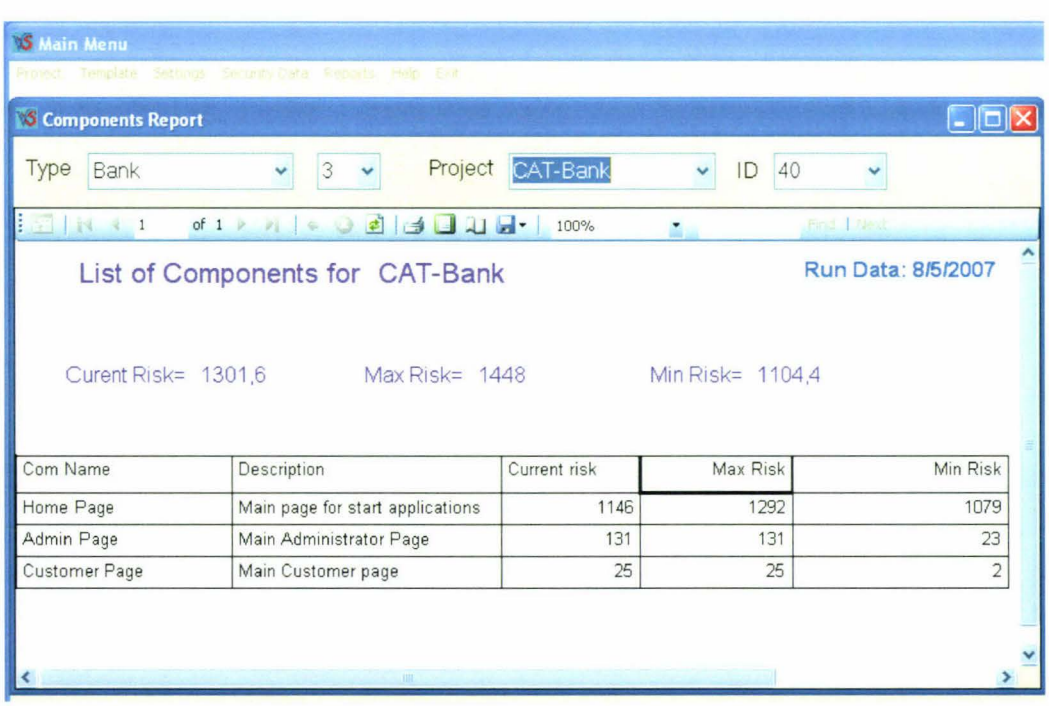


Figure 7. Screen shot of a Component Report for the open project.

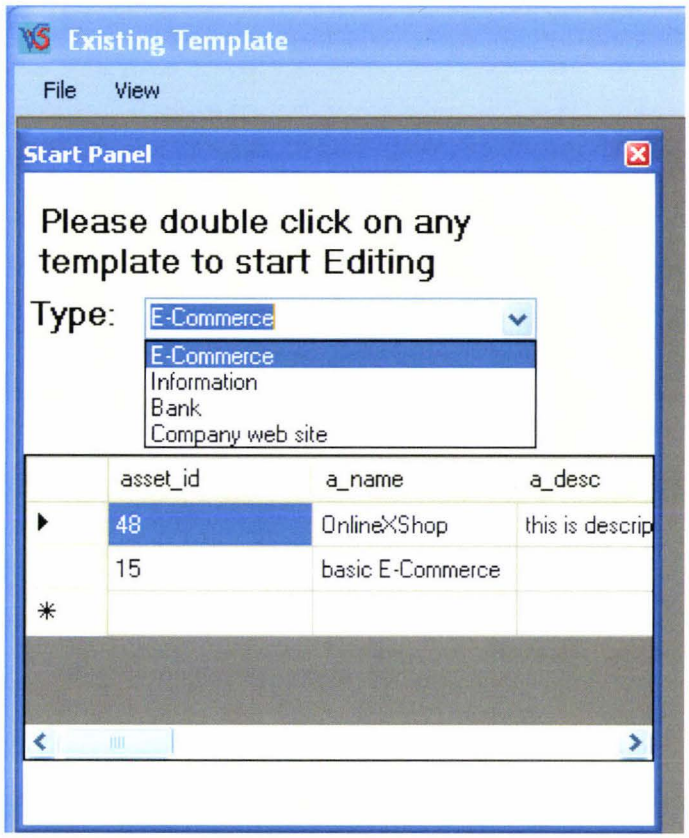


Figure 8. Screen shot of the drop down list of "types of projects" and their templates.

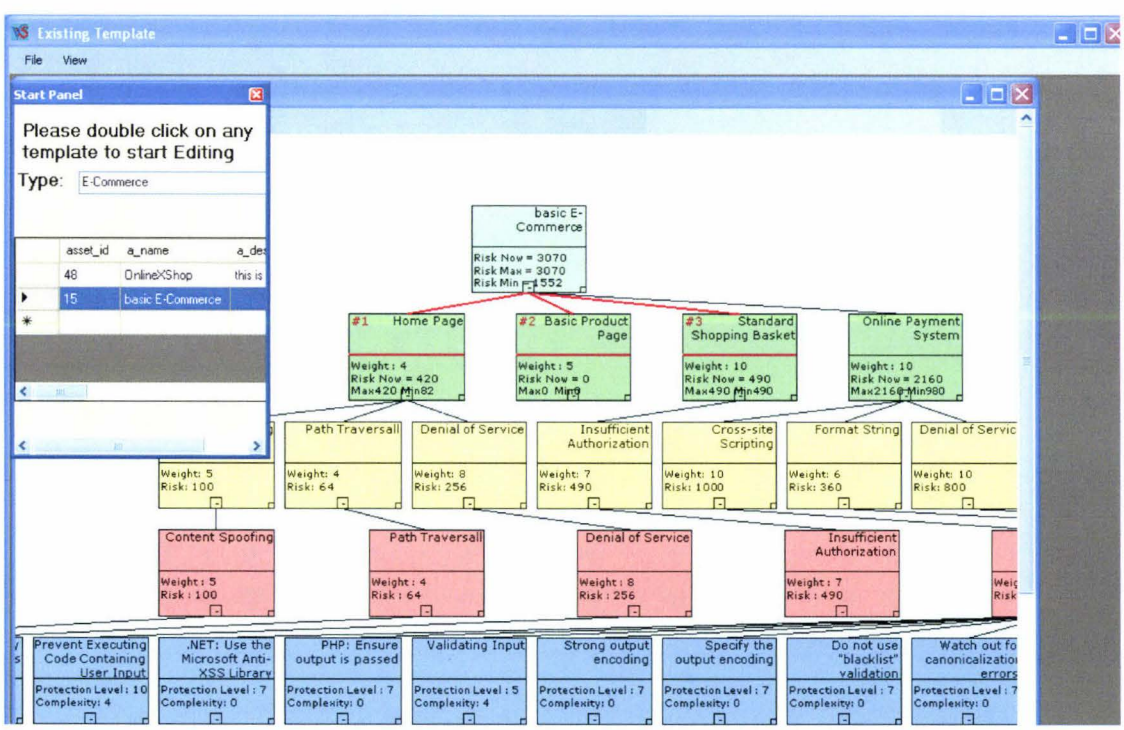


Figure 9. Screen shot of a list of Existing Templates (i.e. Start Panel) of type E-

Commerce and the tree that opens after double-clicking on the second item in the list.

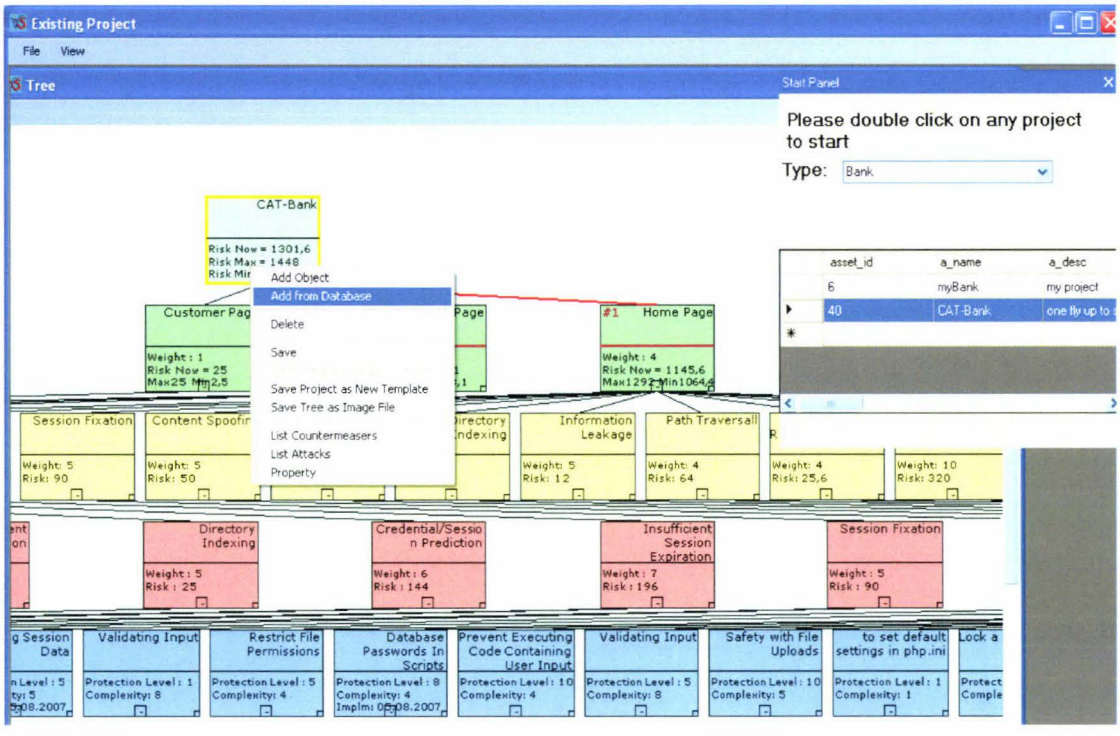


Figure 10. Screen shot of the function "Add from Database"; allows you to add a component to this project using an existing component template from the database.

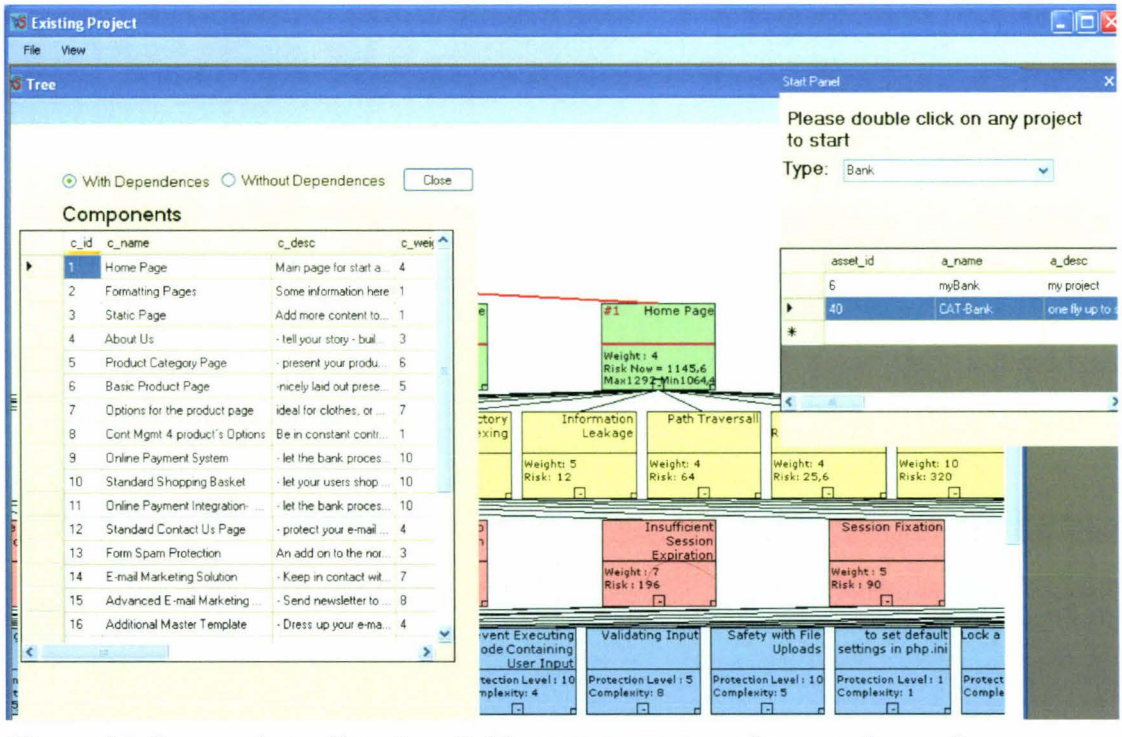


Figure 11. Screen shot a list of available component templates to choose from.

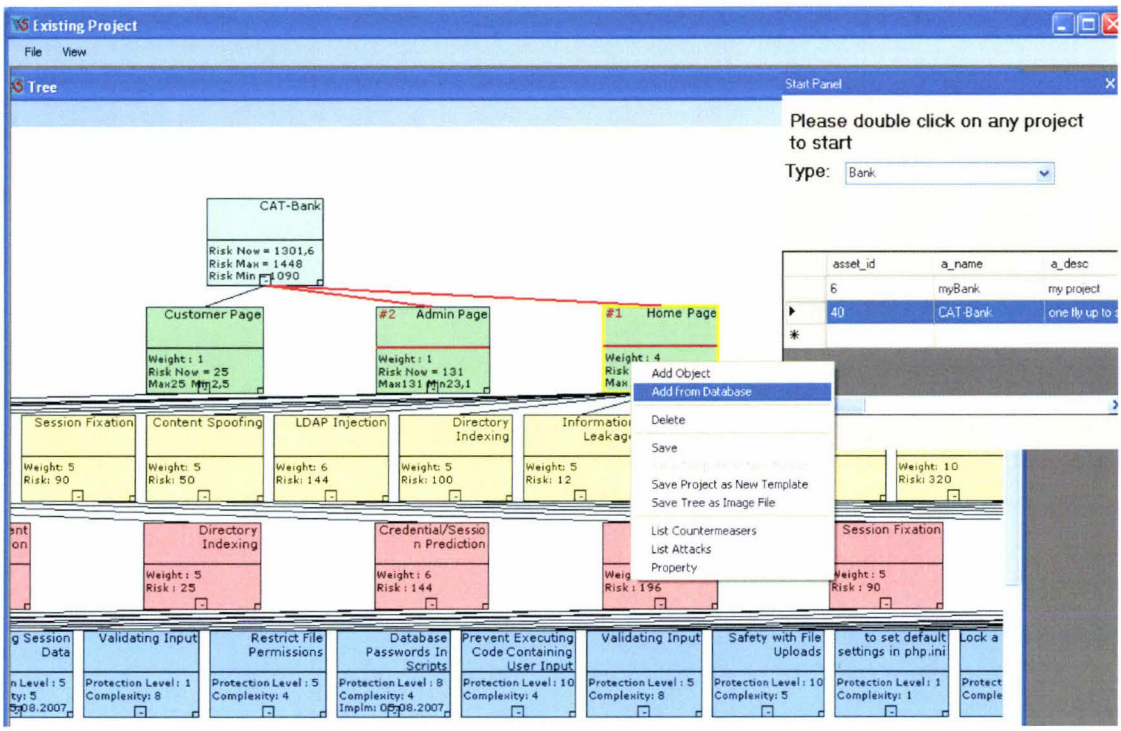


Figure 12. Screen shot of the function "Add from Database"; allows you to add a vulnerability using existing vulnerability templates.

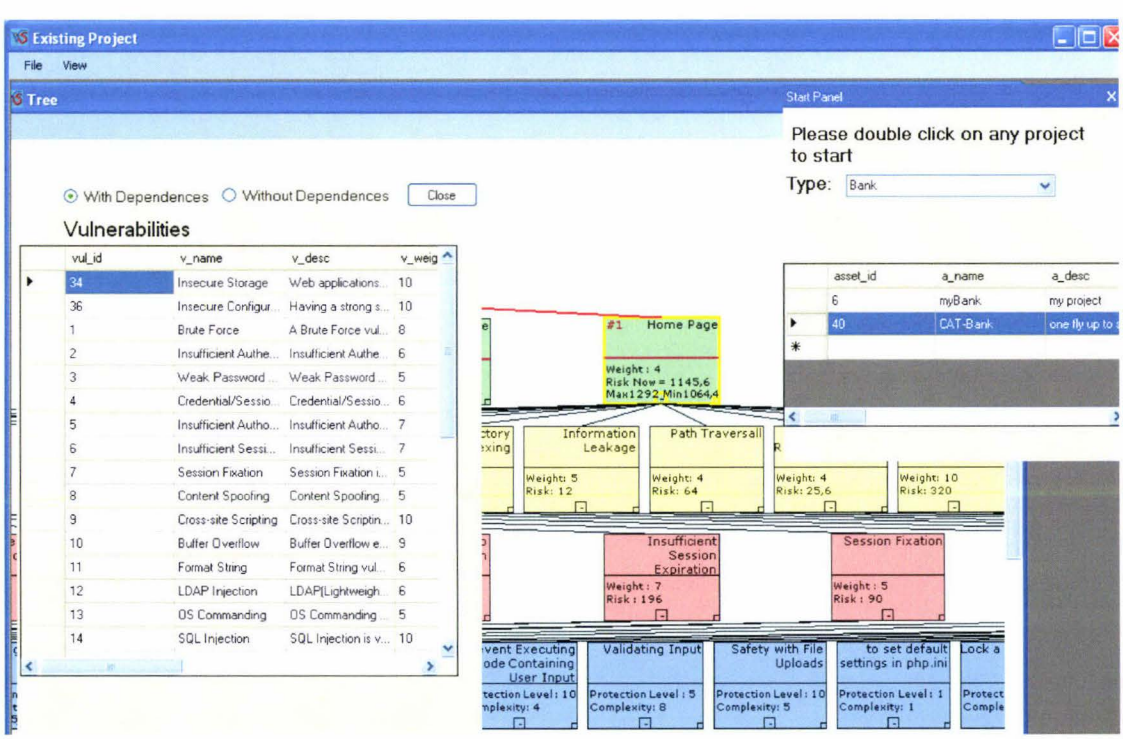


Figure 13. Screen shot of the list of Vulnerability Templates to choose from.

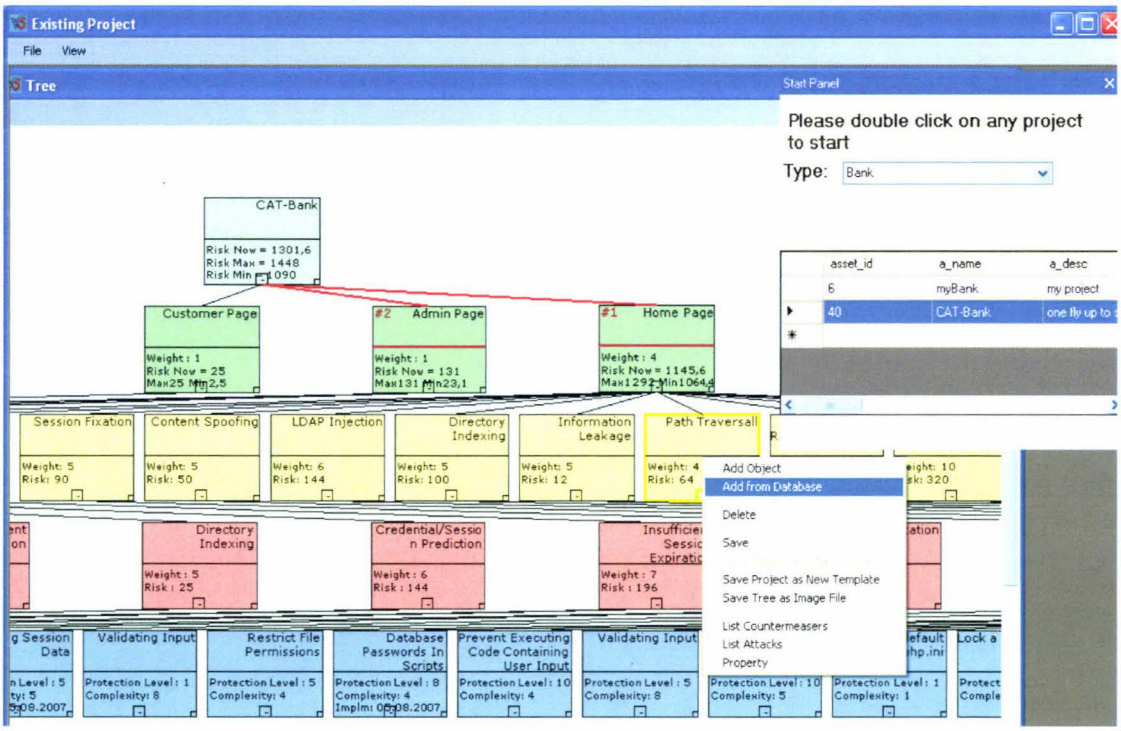


Figure 14. Screen shot of the function "Add from Database"; allows you to add an attack using an existing attack template.

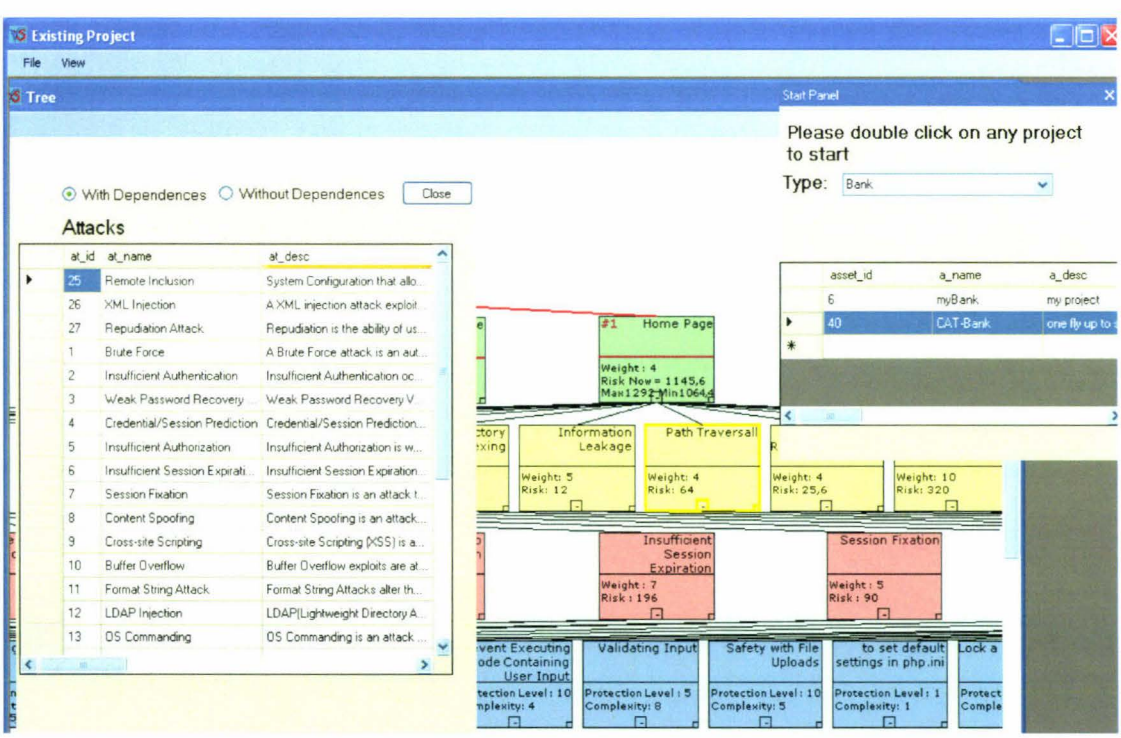


Figure 15. Screen shot of the list of Attack Templates to choose from.

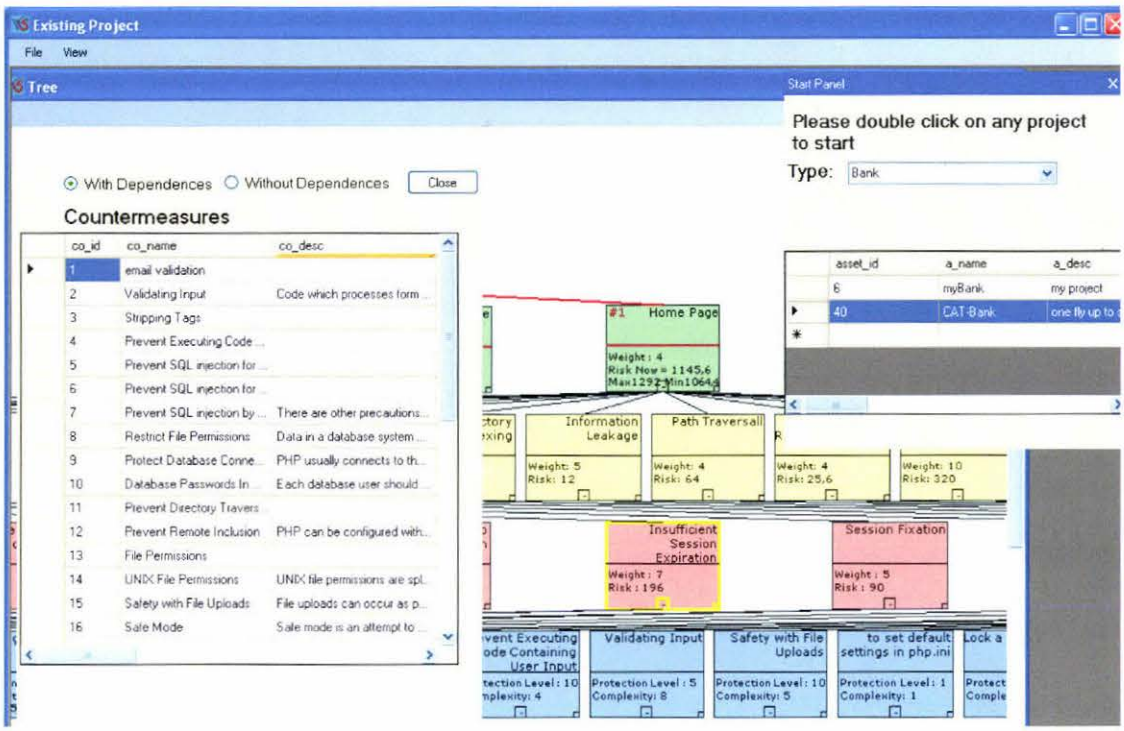


Figure 16. Screen shot of the function "Add from Database" where a user can add a countermeasure to the tree by choosing a countermeasure template from a list.

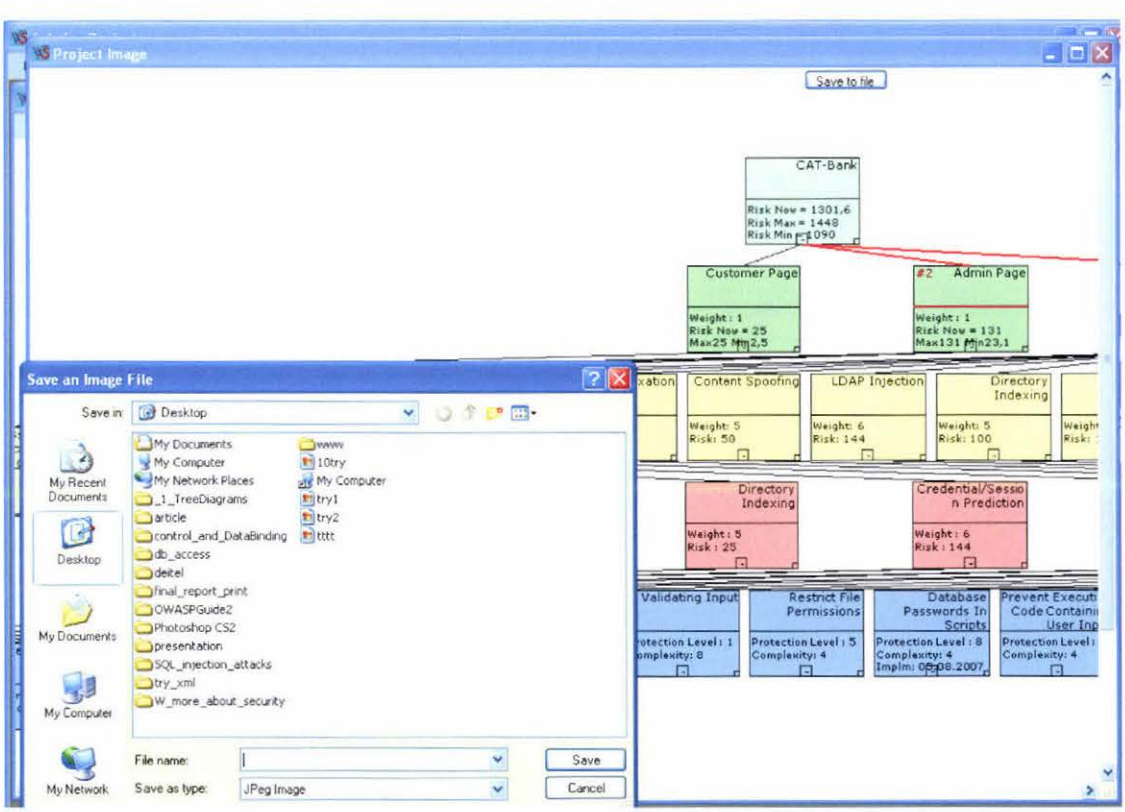


Figure 17. Screen shot of the functionality "Save to Image file" used to save the model for use in other documents or presentations.

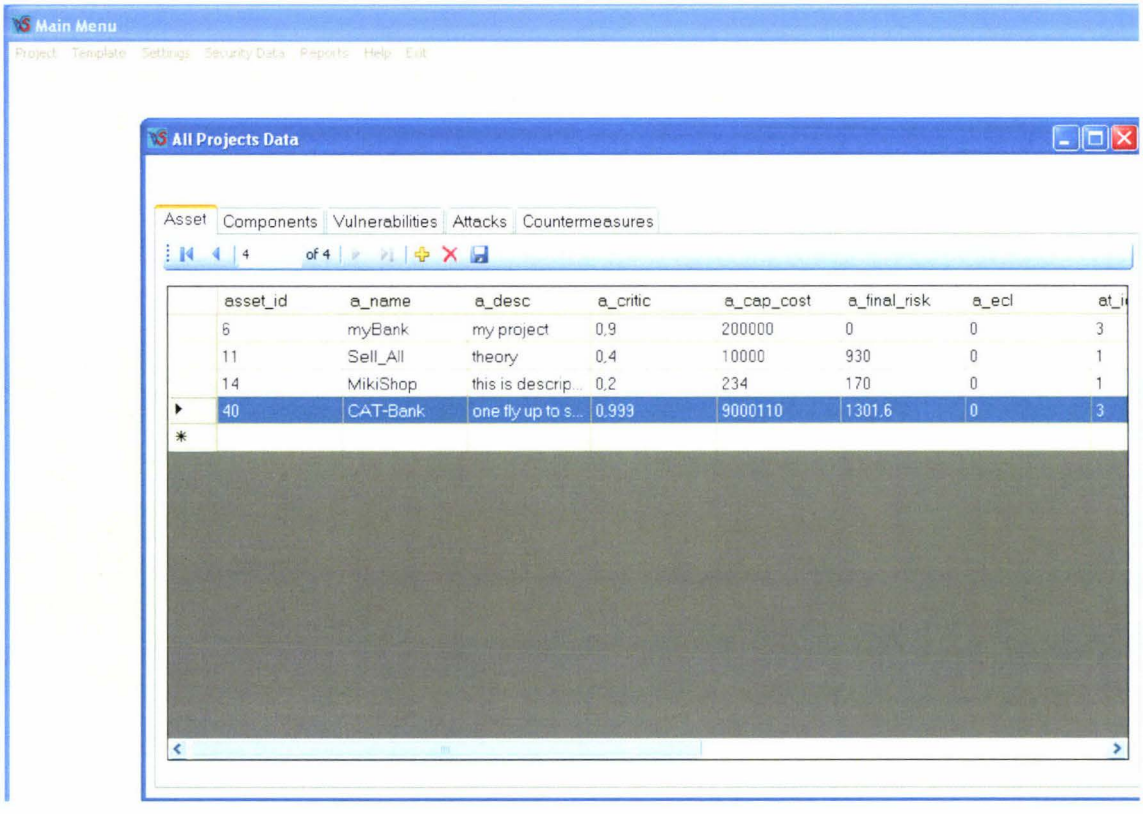


Figure 18. Screen shot of the functionality "Project view as Table". Shows a user choosing a project corresponding to the asset name "CAT-Bank". The tool will populate the other tabs (e.g. Components, etc.).

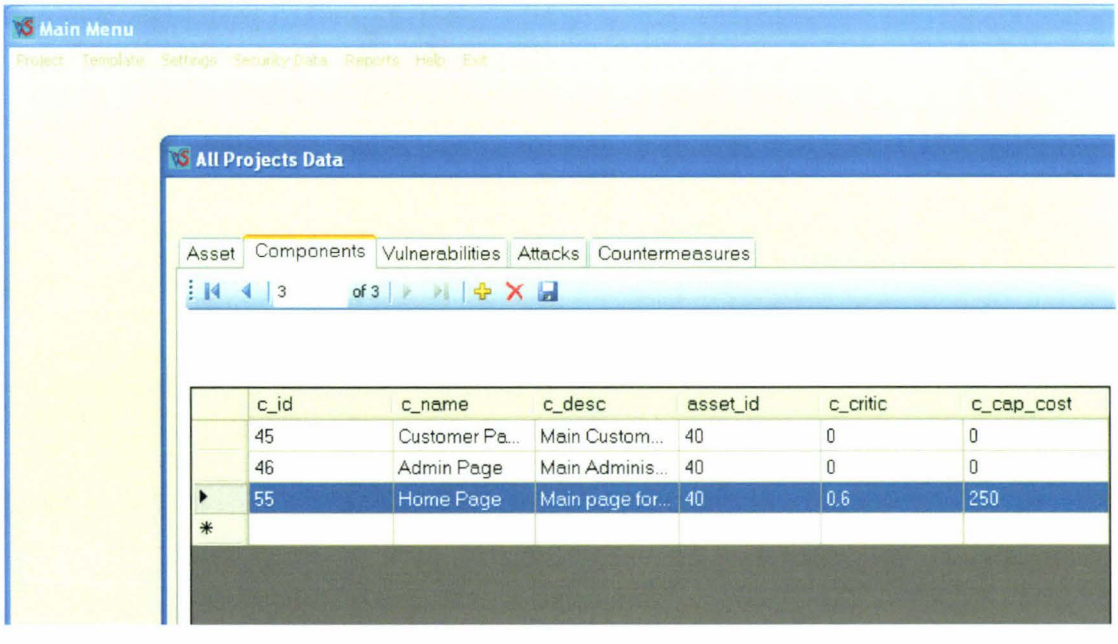
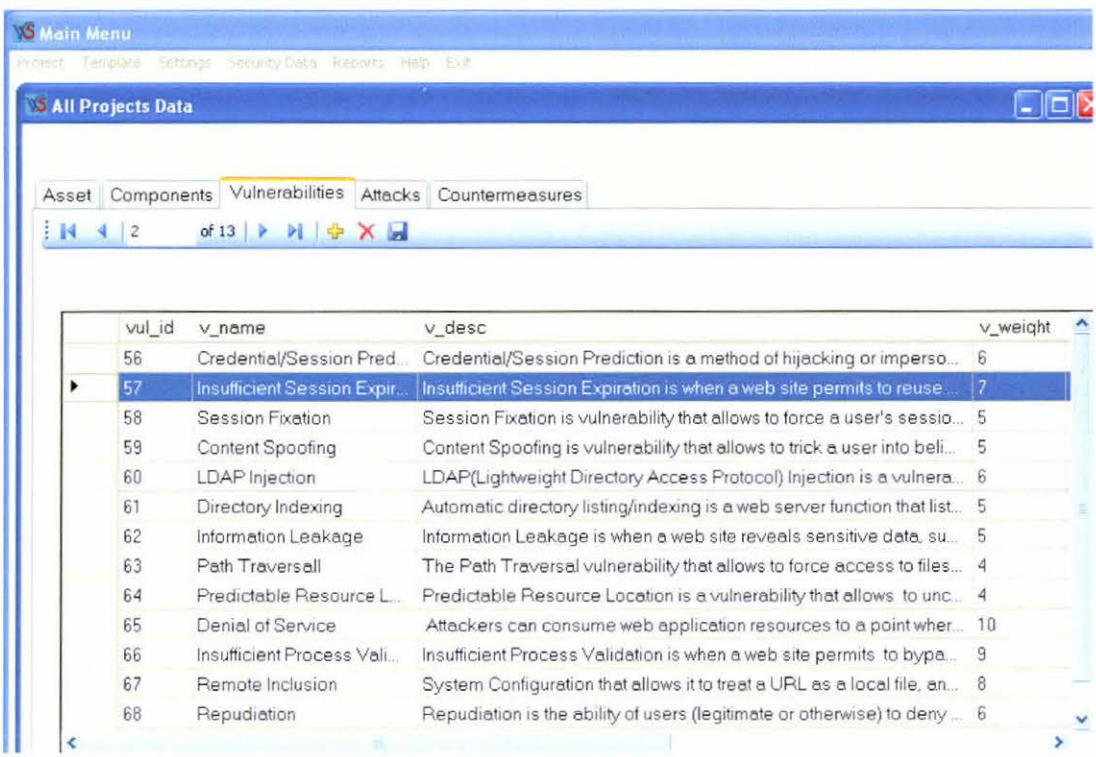


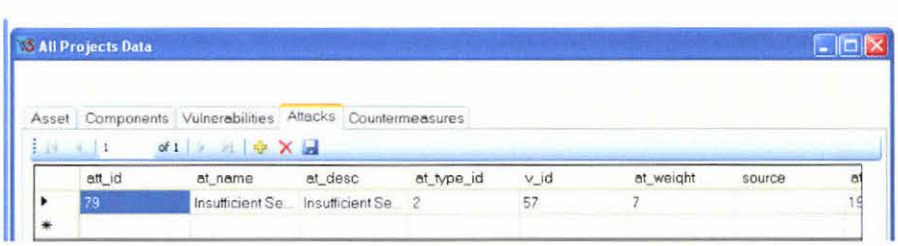
Figure 19. Screen shot after choosing the Components tab for the asset (i.e. project)

chosen in Figure 18.



vul_id	v_name	v_desc	v_weight
56	Credential/Session Pred...	Credential/Session Prediction is a method of hijacking or imperso...	6
57	Insufficient Session Expir...	Insufficient Session Expiration is when a web site permits to reuse ...	7
58	Session Fixation	Session Fixation is vulnerability that allows to force a user's sessio...	5
59	Content Spoofing	Content Spoofing is vulnerability that allows to trick a user into beli...	5
60	LDAP Injection	LDAP(Lightweight Directory Access Protocol) Injection is a vulnera...	6
61	Directory Indexing	Automatic directory listing/indexing is a web server function that list...	5
62	Information Leakage	Information Leakage is when a web site reveals sensitive data, su...	5
63	Path Traversall	The Path Traversal vulnerability that allows to force access to files...	4
64	Predictable Resource L...	Predictable Resource Location is a vulnerability that allows to unc...	4
65	Denial of Service	Attackers can consume web application resources to a point wher...	10
66	Insufficient Process Vali...	Insufficient Process Validation is when a web site permits to bypa...	9
67	Remote Inclusion	System Configuration that allows it to treat a URL as a local file, an...	8
68	Repudiation	Repudiation is the ability of users (legitimate or otherwise) to deny ...	6

Figure 20. Screen shot after choosing a component from Figure 19 and then choosing the Vulnerabilities tab. Shows the vulnerabilities belonging to the component "Home Page", within the asset "CAT-Bank".



att_id	at_name	at_desc	at_type_id	v_id	at_weight	source	at
79	Insufficient Se...	Insufficient Se...	2	57	7		19

Figure 21. Screen shot of the Attacks which use the particular vulnerability selected from Figure 20.

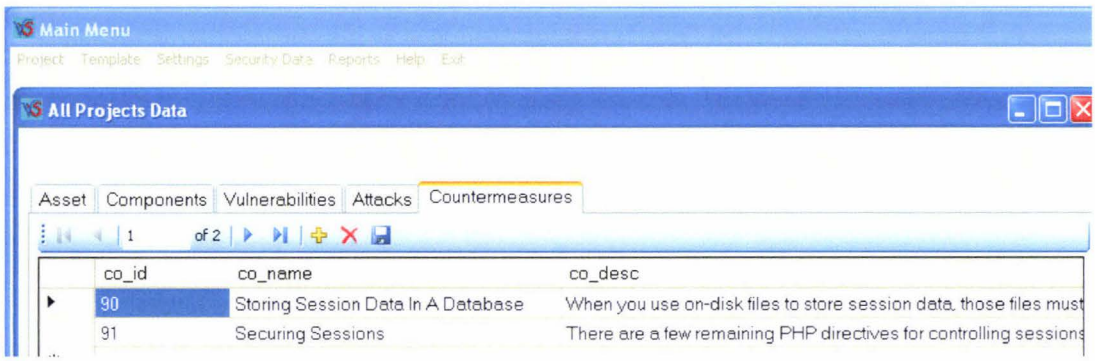


Figure 22. Screen shot of Countermeasures for preventing or reducing the effects of a particular attack chosen in Figure 21.

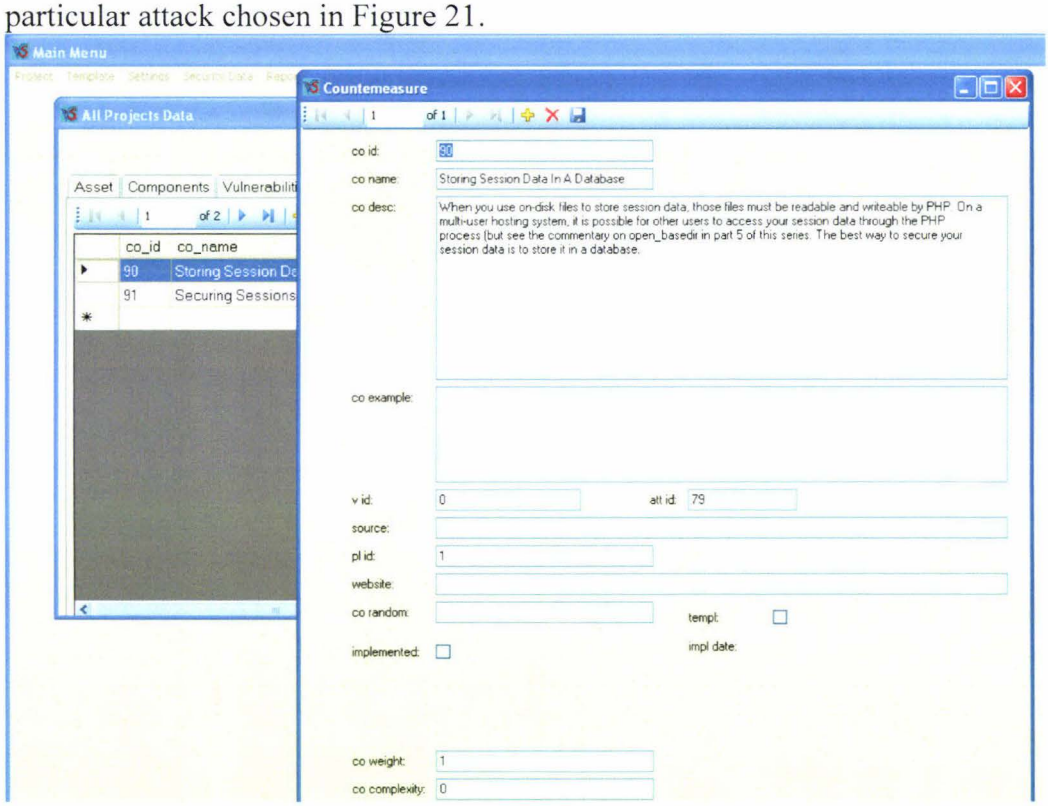


Figure 23. Screen shot of the functionality "Project view as Table" showing the information stored on a particular countermeasure.

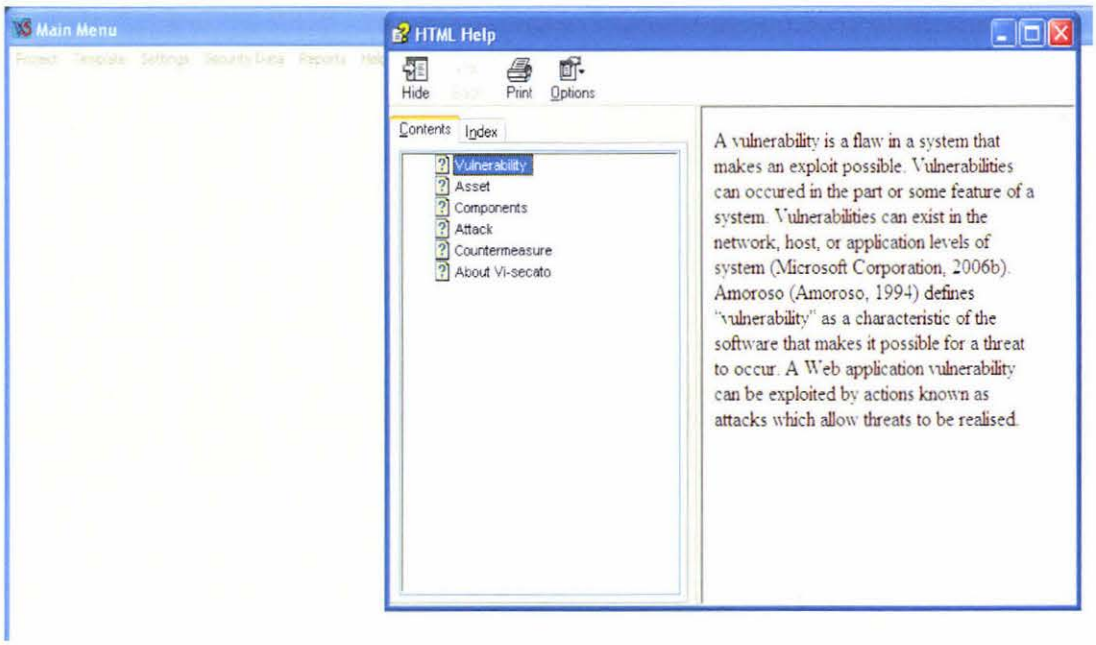


Figure 24. Screen shot showing a sample of online "Help".

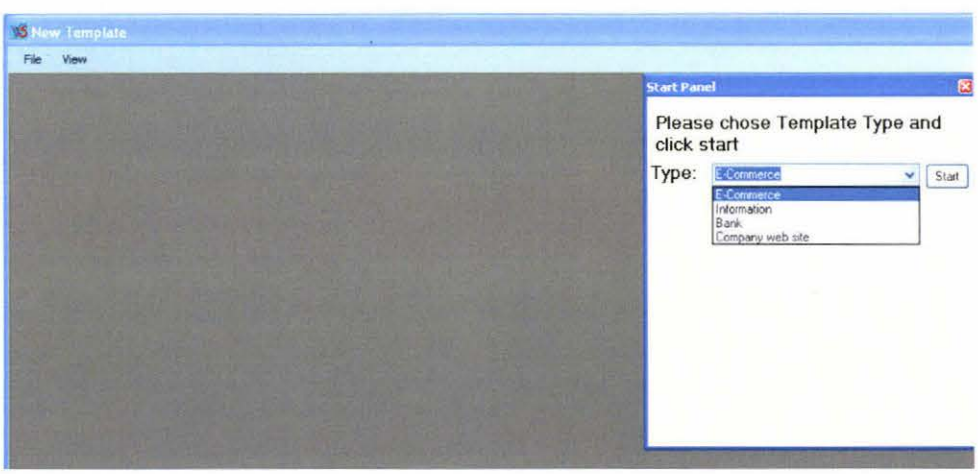


Figure 25. Screen shot of the functionality "Template/ New"; a user selects a type of template from a drop down list.

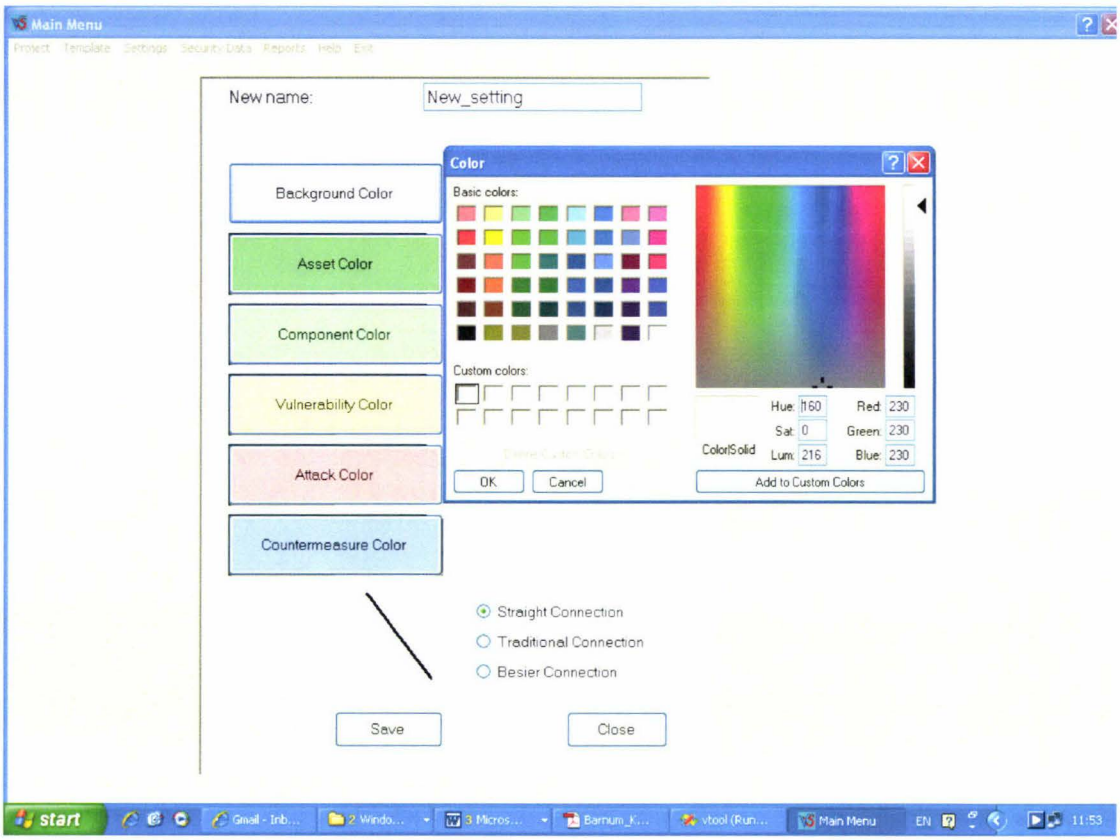


Figure 26. Vi-Secanto's Main Menu Form with New Settings Panel, each button defines colour settings for a particular tree level.

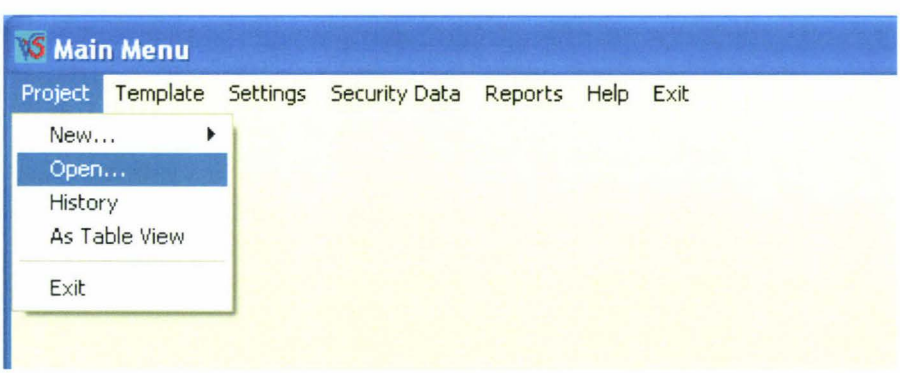


Figure 27. Screen shot of menu navigation for the functionality "Project/ Open".

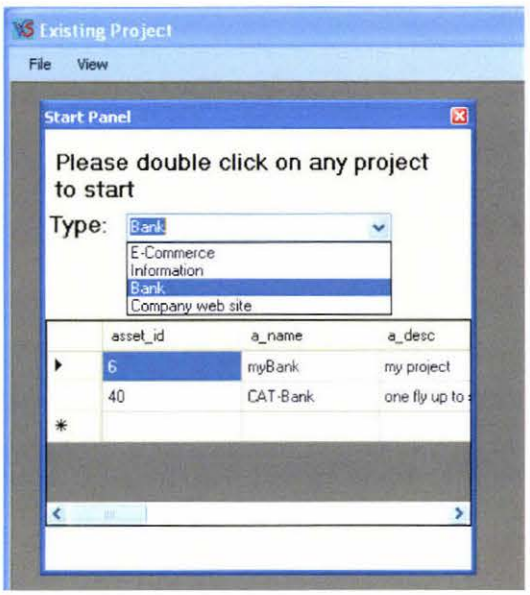


Figure 28. Screen shot of the functionality "Project/ Open"; a user selects a type of project from a drop down list.

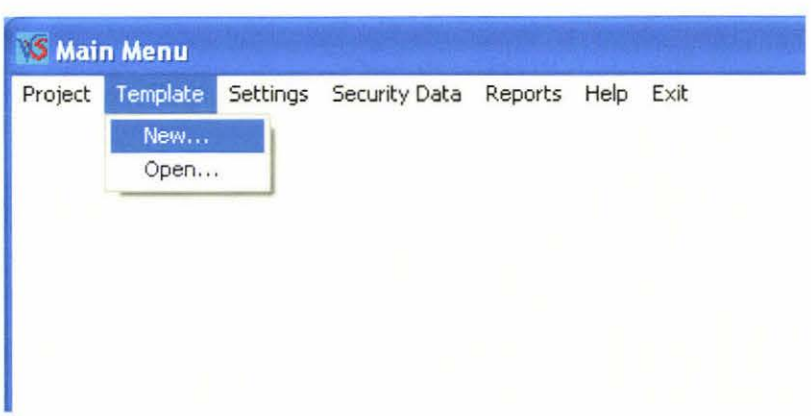


Figure 29. Screen shot of menu navigation for the functionality "Template/ New".

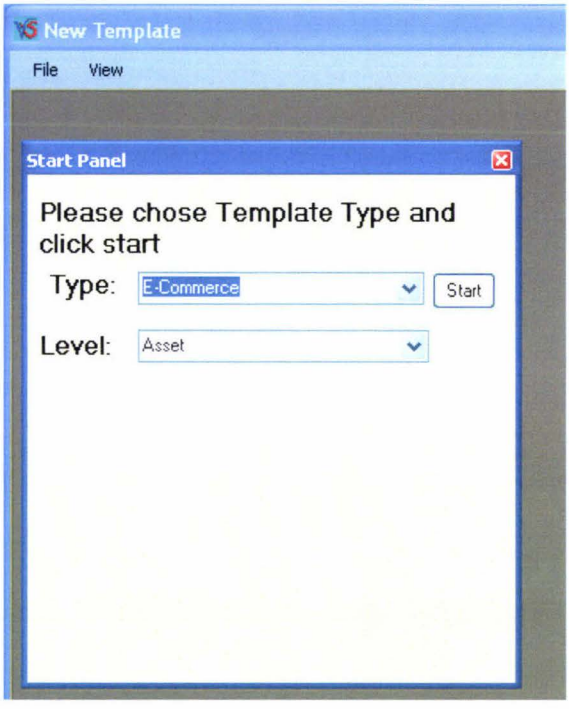


Figure 30. Screen shot of the functionality "Template/ New"; a user selects a type of template and the level of security entity (e.g. Asset) from a drop down list.

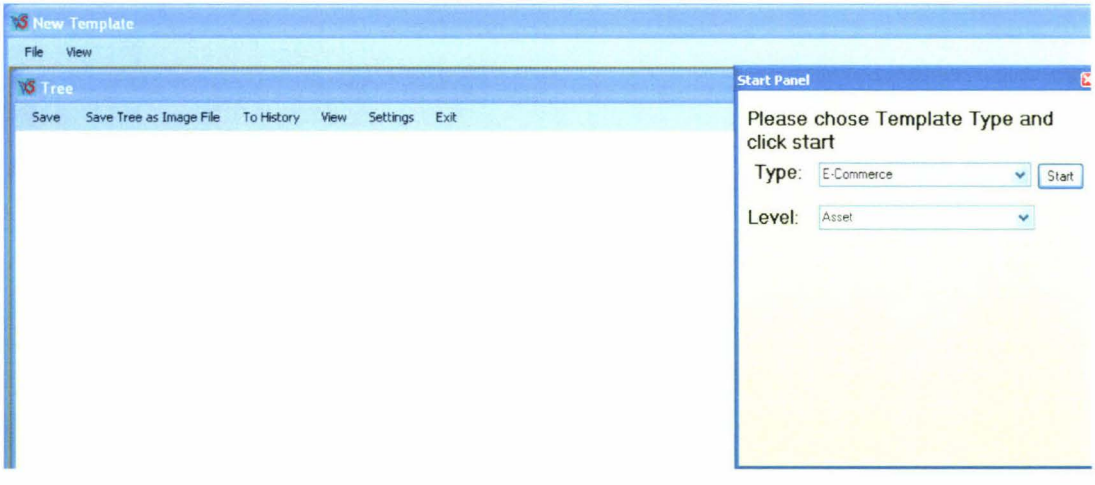


Figure 31. Screen shot of the functionality "Template / New" showing how a user starts with an empty form to start building a tree for an Asset for the chosen template type.

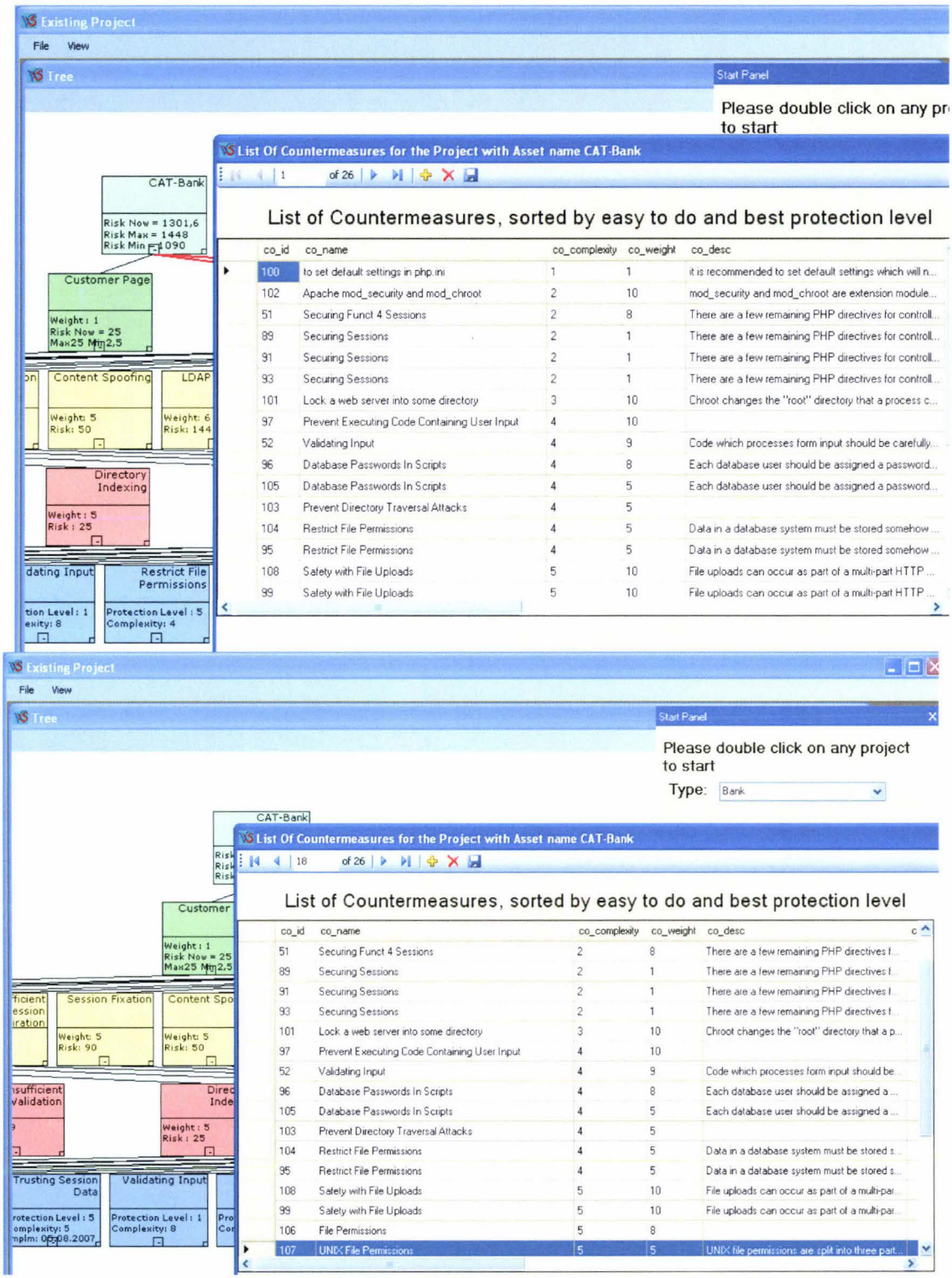


Figure 32. Screen shot of a "List of Countermeasures sorted by easy to do and best protection level" for the open project.

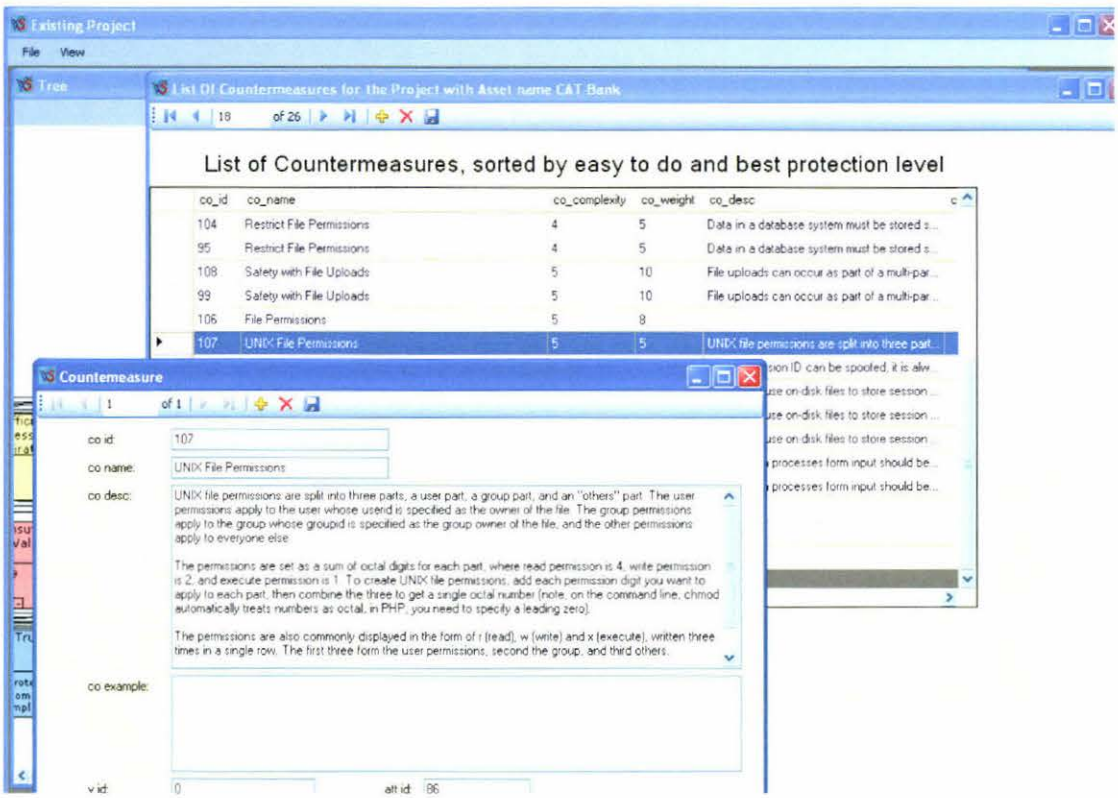


Figure 33. Screen shot of a "List of Countermeasures sorted by easy to do and best protection level" with open details for one countermeasure.

Appendix B: Evaluation Questions

Table 1. *Results of the First Evaluation in the Web Development Company*

Use Case **(Main Menu Options)**	Alternative sub-flows **(Items under each main menu item)**	Description	Satisfaction	Comments
Create a project (Project/New)	New from template	Choose the appropriate template from a table. Save it as a project with a new name.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	New without template	Create a new project from scratch and save it.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
Open (Project/Open)	Open existing project	Open a saved project by choosing it from a list	1-2-3-4-5-6-7-8-9-10 poor--------very good	
Edit tree diagram	Edit asset	Change the selected asset's properties.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Edit component	Change the selected component's properties. The tool recalculates residual risk for the component.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Edit attack	Change the selected attack's properties. The tool recalculates residual risk for the component it belongs to.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Edit counter-measure	Changes the selected countermeasure's properties. The tool recalculates residual risk for the component it belongs to.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Create a component -add new from template.	Choose a component template from a list to add to the current project.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Create a component - add new without template	Add a new component to the project.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Delete asset	When user delete asset it is a deletion of the whole project.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Delete component	Delete a component and all nodes under it.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Delete attack	Delete an attack and all the nodes under it.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Delete counter-measures	Delete countermeasures	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Save new project	Save the current project as a new project.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Save project as template	Save current project as a new template.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Save project template as new project	Save the current project as a new project under a new name.	1-2-3-4-5-6-7-8-9-10 poor--------very good	

Use Case **(Main Menu Options)**	Alternative sub-flows **(Items under each main menu item)**	Description	Satisfaction	Comments
View project data as a table (Project/As Table)	Information about project presented in tables	The user shall be able to view project information in the table view.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
Access security database	Project templates	Project templates (e.g. Shop, information site, bank) - includes components, vulnerabilities, attacks, countermeasures and associations between them.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Component templates	Component templates have components, vulnerabilities, attacks, countermeasures and associations between them.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Vulnerability templates.	Vulnerability templates have vulnerabilities, attacks, countermeasures and associations between them.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Attack templates.	Attack templates have attacks and countermeasures and associations between them.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	Countermeasure templates.	Countermeasure templates contain details on countermeasures.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
Information on Project (project tree open)	From context menu choose "Save tree as Image file"	Information on the current state of the selected project. Shows max Risk, min Risk, current Risk.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
	From context menu choose "List Attacks" .	List of top attack ordered by residual risk level. (see Figure 5.1)	1-2-3-4-5-6-7-8-9-10 poor--------very good	
Report	(Report/ Components Report) (See Figure 7 from Appendices A).	Reports on components sorted by risk to show which components should be given high priority for protection via countermeasures.	1-2-3-4-5-6-7-8-9-10 poor--------very good	

Table 2. *Questions used for the First Evaluation in the Web Development Company.*

Questions	Aim of the question	Satisfaction	Comments
Overall Idea of the tool	To find user opinion about the tool. To find out if the user likes it or not.	1-2-3-4-5-6-7-8-9-10 poor--------very good	
How useful you find it to be for your company	To find about the usability of tool	1-2-3-4-5-6-7-8-9-10 poor--------very useful	
Possibility to use it in your company in the future	To find out if the user would like to make use of the tool in the future.	1-2-3-4-5-6-7-8-9-10 not use----------use	

Table 3. *Questions used for the Second Evaluation*

Questions	**Comments**
Overall idea of the tool?	
How useful do you think it would be for your company?	
Possibility of use by your company in the future?	
What additional parameters do you think need to be kept in the system?	
This tool does a risk calculation based on weights (from 1-10). Do you think that it is a good idea? Is it easy to use?	
What kind of additional functionality would be useful for this tool?	
Please state other suggestions here.	

Table 4 *Questions from the Third Evaluation*

	Satisfaction	Comments
Overall idea of the tool?	1-2-3-4-5-6-7-8-9-10 poor--------very good	
How useful do you think it would be for your company?	1-2-3-4-5-6-7-8-9-10 poor--------very good	
Is there any possibility that it would be used it in your company in future: All tool features? For security educational purpose for developers?	 1-2-3-4-5-6-7-8-9-10 poor--------very good 1-2-3-4-5-6-7-8-9-10 poor--------very good	
This tool does a risk calculation based on weights (from 1-10). Do you think it is a good idea? Is it easy to use?	1-2-3-4-5-6-7-8-9-10 poor--------very good 1-2-3-4-5-6-7-8-9-10 poor--------very good	
This tool has the ability to use security templates in order to bring security knowledge to non- security experts. Do you think it is good idea? Is it easy to use?	1-2-3-4-5-6-7-8-9-10 poor--------very good 1-2-3-4-5-6-7-8-9-10 poor--------very good	
There is a need to regularly update security knowledge which is kept in this tool database. A suggestion was made to put this into final development and support it on open source. Do you think it is good idea?	1-2-3-4-5-6-7-8-9-10 poor--------very good	

Appendix C: Program code for class: "Tree"

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Data;
using System.Drawing.Imaging;
using ControlExtenders; // controls to make panels movable from Codeprojects

namespace vtool
{
        /// <summary>
        /// Tree graphical representation of models for vulnerabilities/attacks trees
        /// </summary>

  public class Tree : System.Windows.Forms.Form
  {

    public ShapeCollection shcol= new ShapeCollection(); // collections of all security objects (like
asset,components, vulner-s, attacks, counterm-s, for building tree
                private ConnectionCollection concol;
    public double[,] array_sec_comp= new double[10, 2]; //for component going to keep gid, risk. start from most
critical one. Used for shows most critical components.
    public double[][] array_sec_vuln = new double[10][ ]; //jagged array, for vulnerab going to keep gid, risk

    public int tot_critic_num = 2; // how many most criticall components need to show
    public int tot_critic_num_v = 2; // how many most criticall vulnerab-s need to show
    public form_Pict frmS;
                private Point[] point = new Point[5];
                private int[] UnitWigth;
                private int indexSelect = -100;
                private int intMode    = -100;
                private Rectangle RecShape;
    private int mVul_now;
    private int mComponent_now;
    // current security object now:
    public Asset asset_now=new Asset();
    public Component comp_now = new Component();
    public Vulnerability vuln_now =new Vulnerability();
    public Attack attack_now=new Attack();
    public Countermeasure countm_now=new Countermeasure();
    public int con_type;
    private double ecl_sum = 0;
    private bool my_templ = true; // if current user document template , if not=false ( we have real project)
    public bool have_asset = false; // if user start new empty project, program allows have only one asset per
project
    public int temp_level = 0; // if user selected  level of template from comboBox_Level ( in parent form) 0-asset
template; 1- component templ; 2- vuln; 3- attack; 4 -counterm;
    // our graphic dimentions
    public int  SchWidth =1000;
    public int  SchHeight = 800;
    public int  Sch_shSize =  115;
    public int  Sch_distBt = 10;
    //distance for diffrent level of graph
    public int dist = 115; //univer.
```

```
  public int dist1 = 115; //will be recalculated later
public int dist2 = 115; //will be recalculated
public int dist3 = 115; //will be recalculated
public int dist4 = 115; //will be recalculated

// summary  width  for levels
public int sumWidLevel1 = 0;
public int sumWidLevel2 = 0;
public int sumWidLevel3 = 0;
public int sumWidLevel4 = 0;
public int g_id = 0; // index for graphic place of our object

public int a_calcul_type = 1;    // in future, it will be possible to implemente  diffrent calculations formulas
public forms_Data.form_ListCount frmList;
public forms_Data.form_ListAttacks frmAttacks;

public System.Windows.Forms.PictureBox  TreePicture;

private System.Windows.Forms.ContextMenu OperationMenu;
                private System.Windows.Forms.MenuItem menuItem10;
                private System.Windows.Forms.MenuItem menuItem11;
                private System.Windows.Forms.MenuItem menuItemProperty;
private System.Windows.Forms.MenuItem menuAddAsset;
private ColorDialog colorD_backGr;
public MenuItem menuSaveTemAsPr;
private MenuItem menuItemSaveAll;
private Panel panel_wait;
private Label label1;
private ProgressBar progressBar1;
private MenuItem menuItem2;
private MenuItem menuItemDel;
private Panel paGrProperty;
private Label laSettings_Now;
private Label label2;
public ComboBox comBox_Settings;
private Button bnPan_close;
private Button bnSaveSettings;
private MenuStrip menuStrip1;
private ToolStripMenuItem saveToolStripMenuItem;
private ToolStripMenuItem saveGraphicToFileToolStripMenuItem;
private ToolStripMenuItem toolStripMenuItemSettings;
private ToolStripMenuItem exitToolStripMenuItem;
private db_vt_1DataSet db_vt_1DataSet;
private BindingSource settingsBindingSource;
private vtool.db_vt_1DataSetTableAdapters.settingsTableAdapter settingsTableAdapter;
private DataGridView settingsDataGridView;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn1;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn2;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn3;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn4;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn5;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn6;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn9;
private ToolStripMenuItem saveAllToolStripMenuItem;
private ToolStripMenuItem saveAsNewTemplateToolStripMenuItem;
private ToolStripMenuItem saveAsNewProjectToolStripMenuItem;
private MenuItem menuItemAdd_frLib;
private BindingSource library_vulnsBindingSource;
private vtool.db_vt_1DataSetTableAdapters.library_vulnsTableAdapter library_vulnsTableAdapter;
private Panel panel_Lib;
```

```
private DataGridView library_vulnsDataGridView;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn10;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn11;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn12;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn13;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn14;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn15;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn16;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn17;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn18;
private BindingSource library_attacksBindingSource;
private vtool.db_vt_1DataSetTableAdapters.library_attacksTableAdapter library_attacksTableAdapter;
private DataGridView library_attacksDataGridView;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn19;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn20;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn21;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn22;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn23;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn24;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn25;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn26;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn27;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn28;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn29;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn30;
private DataGridViewCheckBoxColumn dataGridViewCheckBoxColumn1;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn31;
private BindingSource library_countsBindingSource;
private vtool.db_vt_1DataSetTableAdapters.library_countsTableAdapter library_countsTableAdapter;
private DataGridView library_countsDataGridView;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn32;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn33;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn34;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn35;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn36;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn37;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn38;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn39;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn40;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn41;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn42;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn43;
private DataGridViewCheckBoxColumn dataGridViewCheckBoxColumn2;
private Button bnClosePan;
private BindingSource library_compsBindingSource;
private vtool.db_vt_1DataSetTableAdapters.library_compsTableAdapter library_compsTableAdapter;
private DataGridView library_compsDataGridView;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn44;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn45;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn46;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn47;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn48;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn49;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn50;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn51;
private DataGridViewCheckBoxColumn dataGridViewCheckBoxColumn3;
private DataGridViewTextBoxColumn dataGridViewTextBoxColumn52;
private BindingSource library_comp_vulnBindingSource;
private vtool.db_vt_1DataSetTableAdapters.library_comp_vulnTableAdapter library_comp_vulnTableAdapter;
private DataGridView library_comp_vulnDataGridView;
```

```
        private DataGridViewTextBoxColumn dataGridViewTextBoxColumn53;
    private DataGridViewTextBoxColumn dataGridViewTextBoxColumn54;
    private BindingSource library_v_aBindingSource;
    private vtool.db_vt_1DataSetTableAdapters.library_v_aTableAdapter library_v_aTableAdapter;
    private DataGridView library_v_aDataGridView;
    private DataGridViewTextBoxColumn dataGridViewTextBoxColumn55;
    private DataGridViewTextBoxColumn dataGridViewTextBoxColumn56;
    private DataGridViewTextBoxColumn dataGridViewTextBoxColumn57;
    private BindingSource library_coun_4a_vBindingSource;
    private vtool.db_vt_1DataSetTableAdapters.library_coun_4a_vTableAdapter
library_coun_4a_vTableAdapter;
    private DataGridView library_coun_4a_vDataGridView;
    private DataGridViewTextBoxColumn dataGridViewTextBoxColumn58;
    private DataGridViewTextBoxColumn dataGridViewTextBoxColumn59;
    private DataGridViewTextBoxColumn dataGridViewTextBoxColumn60;
    private DataGridViewTextBoxColumn dataGridViewTextBoxColumn61;
    private RadioButton rBn_without_dep;
    private RadioButton rBn_with_dep;
    private Label lbTableName;
    private ToolStripMenuItem menuView;
    private ToolStripMenuItem toolStripMenuItemToHistory;
    private MenuItem menuItem_as_NewTempl;
    private MenuItem menuItemListCountm;
    private MenuItem menuItemListAttacks;
    private MenuItem menuItemSaveTreeIm;
    private IContainer components;


                public Tree()
                {
                        //
                        // Required for Windows Form Designer support
                        //
                        InitializeComponent();
                }

                /// <summary>
                /// Clean up any resources being used.
                /// </summary>
                protected override void Dispose( bool disposing )
                {
                        if( disposing )
                        {
                                if(components != null)
                                {
                                        components.Dispose();
                                }
                        }
                        base.Dispose( disposing );
                }

                #region Windows Form Designer generated code
                /// <summary>
                /// Required method for Designer support - do not modify
                /// the contents of this method with the code editor.
                /// </summary>
                private void InitializeComponent()
                {
        this.components = new System.ComponentModel.Container();
```

```
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Tree));
        this.TreePicture = new System.Windows.Forms.PictureBox();
        this.OperationMenu = new System.Windows.Forms.ContextMenu();
        this.menuAddAsset = new System.Windows.Forms.MenuItem();
        this.menuItemAdd_frLib = new System.Windows.Forms.MenuItem();
        this.menuItem10 = new System.Windows.Forms.MenuItem();
        this.menuItemDel = new System.Windows.Forms.MenuItem();
        this.menuItem11 = new System.Windows.Forms.MenuItem();
        this.menuItemSaveAll = new System.Windows.Forms.MenuItem();
        this.menuSaveTemAsPr = new System.Windows.Forms.MenuItem();
        this.menuItem_as_NewTempl = new System.Windows.Forms.MenuItem();
        this.menuItemSaveTreeIm = new System.Windows.Forms.MenuItem();
        this.menuItem2 = new System.Windows.Forms.MenuItem();
        this.menuItemListCountm = new System.Windows.Forms.MenuItem();
        this.menuItemListAttacks = new System.Windows.Forms.MenuItem();
        this.menuItemProperty = new System.Windows.Forms.MenuItem();
        this.colorD_backGr = new System.Windows.Forms.ColorDialog();
        this.panel_wait = new System.Windows.Forms.Panel();
        this.label1 = new System.Windows.Forms.Label();
        this.progressBar1 = new System.Windows.Forms.ProgressBar();
        this.paGrProperty = new System.Windows.Forms.Panel();
        this.settingsDataGridView = new System.Windows.Forms.DataGridView();
        this.dataGridViewTextBoxColumn1 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn2 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn3 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn4 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn5 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn6 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn9 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.settingsBindingSource = new System.Windows.Forms.BindingSource(this.components);
        this.db_vt_1DataSet = new vtool.db_vt_1DataSet();
        this.laSettings_Now = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.comBox_Settings = new System.Windows.Forms.ComboBox();
        this.bnPan_close = new System.Windows.Forms.Button();
        this.bnSaveSettings = new System.Windows.Forms.Button();
        this.menuStrip1 = new System.Windows.Forms.MenuStrip();
        this.saveToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.saveAllToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.saveAsNewTemplateToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.saveAsNewProjectToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.saveGraphicToFileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.toolStripMenuItemToHistory = new System.Windows.Forms.ToolStripMenuItem();
        this.menuView = new System.Windows.Forms.ToolStripMenuItem();
        this.toolStripMenuItemSettings = new System.Windows.Forms.ToolStripMenuItem();
        this.exitToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
        this.panel_Lib = new System.Windows.Forms.Panel();
        this.lbTableName = new System.Windows.Forms.Label();
        this.rBn_without_dep = new System.Windows.Forms.RadioButton();
        this.rBn_with_dep = new System.Windows.Forms.RadioButton();
        this.library_coun_4a_vDataGridView = new System.Windows.Forms.DataGridView();
        this.dataGridViewTextBoxColumn58 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn59 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn60 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn61 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.library_coun_4a_vBindingSource = new System.Windows.Forms.BindingSource(this.components);
        this.library_v_aDataGridView = new System.Windows.Forms.DataGridView();
        this.dataGridViewTextBoxColumn55 = new System.Windows.Forms.DataGridViewTextBoxColumn();
        this.dataGridViewTextBoxColumn56 = new System.Windows.Forms.DataGridViewTextBoxColumn();
```

```
  this.dataGridViewTextBoxColumn57 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.library_v_aBindingSource = new System.Windows.Forms.BindingSource(this.components);
this.library_comp_vulnDataGridView = new System.Windows.Forms.DataGridView();
this.dataGridViewTextBoxColumn53 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn54 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.library_comp_vulnBindingSource = new System.Windows.Forms.BindingSource(this.components);
this.library_compsDataGridView = new System.Windows.Forms.DataGridView();
this.dataGridViewTextBoxColumn44 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn45 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn46 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn47 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn48 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn49 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn50 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn51 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewCheckBoxColumn3 = new System.Windows.Forms.DataGridViewCheckBoxColumn();
this.dataGridViewTextBoxColumn52 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.library_compsBindingSource = new System.Windows.Forms.BindingSource(this.components);
this.bnClosePan = new System.Windows.Forms.Button();
this.library_countsDataGridView = new System.Windows.Forms.DataGridView();
this.dataGridViewTextBoxColumn32 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn33 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn34 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn35 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn36 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn37 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn38 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn39 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn40 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn41 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn42 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn43 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewCheckBoxColumn2 = new System.Windows.Forms.DataGridViewCheckBoxColumn();
this.library_countsBindingSource = new System.Windows.Forms.BindingSource(this.components);
this.library_attacksDataGridView = new System.Windows.Forms.DataGridView();
this.dataGridViewTextBoxColumn19 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn20 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn21 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn22 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn23 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn24 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn25 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn26 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn27 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn28 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn29 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn30 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewCheckBoxColumn1 = new System.Windows.Forms.DataGridViewCheckBoxColumn();
this.dataGridViewTextBoxColumn31 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.library_attacksBindingSource = new System.Windows.Forms.BindingSource(this.components);
this.library_vulnsDataGridView = new System.Windows.Forms.DataGridView();
this.dataGridViewTextBoxColumn10 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn11 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn12 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn13 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn14 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn15 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn16 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn17 = new System.Windows.Forms.DataGridViewTextBoxColumn();
this.dataGridViewTextBoxColumn18 = new System.Windows.Forms.DataGridViewTextBoxColumn();
```

```
        this.library_vulnsBindingSource = new System.Windows.Forms.BindingSource(this.components);
        this.settingsTableAdapter = new vtool.db_vt_1DataSetTableAdapters.settingsTableAdapter();
        this.library_vulnsTableAdapter = new vtool.db_vt_1DataSetTableAdapters.library_vulnsTableAdapter();
        this.library_attacksTableAdapter = new vtool.db_vt_1DataSetTableAdapters.library_attacksTableAdapter();
        this.library_countsTableAdapter = new vtool.db_vt_1DataSetTableAdapters.library_countsTableAdapter();
        this.library_compsTableAdapter = new vtool.db_vt_1DataSetTableAdapters.library_compsTableAdapter();
        this.library_comp_vulnTableAdapter = new
vtool.db_vt_1DataSetTableAdapters.library_comp_vulnTableAdapter();
        this.library_v_aTableAdapter = new vtool.db_vt_1DataSetTableAdapters.library_v_aTableAdapter();
        this.library_coun_4a_vTableAdapter = new
vtool.db_vt_1DataSetTableAdapters.library_coun_4a_vTableAdapter();
        ((System.ComponentModel.ISupportInitialize)(this.TreePicture)).BeginInit();
        this.panel_wait.SuspendLayout();
        this.paGrProperty.SuspendLayout();
        ((System.ComponentModel.ISupportInitialize)(this.settingsDataGridView)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.settingsBindingSource)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.db_vt_1DataSet)).BeginInit();
        this.menuStrip1.SuspendLayout();
        this.panel_Lib.SuspendLayout();
        ((System.ComponentModel.ISupportInitialize)(this.library_coun_4a_vDataGridView)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_coun_4a_vBindingSource)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_v_aDataGridView)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_v_aBindingSource)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_comp_vulnDataGridView)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_comp_vulnBindingSource)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_compsDataGridView)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_compsBindingSource)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_countsDataGridView)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_countsBindingSource)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_attacksDataGridView)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_attacksBindingSource)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_vulnsDataGridView)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.library_vulnsBindingSource)).BeginInit();
        this.SuspendLayout();
        //
        // TreePicture
        //
        this.TreePicture.BackColor = System.Drawing.SystemColors.Info;
        this.TreePicture.ContextMenu = this.OperationMenu;
        this.TreePicture.Location = new System.Drawing.Point(5, 30);
        this.TreePicture.Name = "TreePicture";
        this.TreePicture.Size = new System.Drawing.Size(650, 488);
        this.TreePicture.TabIndex = 0;
        this.TreePicture.TabStop = false;
        this.TreePicture.WaitOnLoad = true;
        this.TreePicture.DoubleClick += new System.EventHandler(this.TreePicture_DoubleClick);
        this.TreePicture.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.TreePicture_MouseDown);
        this.TreePicture.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.TreePicture_MouseMove);
        this.TreePicture.Paint += new System.Windows.Forms.PaintEventHandler(this.TreePicture_Paint);
        this.TreePicture.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.TreePicture_MouseUp);
        //
        // OperationMenu
        //
        this.OperationMenu.MenuItems.AddRange(new System.Windows.Forms.MenuItem[] {
        this.menuAddAsset,
        this.menuItemAdd_frLib,
        this.menuItem10,
```

```
  this.menuItemDel,
this.menuItem11,
this.menuItemSaveAll,
this.menuSaveTemAsPr,
this.menuItem_as_NewTempl,
this.menuItemSaveTreeIm,
this.menuItem2,
this.menuItemListCountm,
this.menuItemListAttacks,
this.menuItemProperty});
//
// menuAddAsset
//
this.menuAddAsset.Index = 0;
this.menuAddAsset.Text = "Add Object";
this.menuAddAsset.Click += new System.EventHandler(this.menuAddAsset_Click);
//
// menuItemAdd_frLib
//
this.menuItemAdd_frLib.Index = 1;
this.menuItemAdd_frLib.Text = "Add from Database";
this.menuItemAdd_frLib.Click += new System.EventHandler(this.menuItemAdd_frLib_Click);
//
// menuItem10
//
this.menuItem10.Index = 2;
this.menuItem10.Text = "-";
//
// menuItemDel
//
this.menuItemDel.Index = 3;
this.menuItemDel.Text = "Delete";
this.menuItemDel.Click += new System.EventHandler(this.menuItemDel_Click);
//
// menuItem11
//
this.menuItem11.Index = 4;
this.menuItem11.Text = "-";
//
// menuItemSaveAll
//
this.menuItemSaveAll.Index = 5;
this.menuItemSaveAll.Text = "Save ";
this.menuItemSaveAll.Click += new System.EventHandler(this.menuItemSaveAll_Click);
//
// menuSaveTemAsPr
//
this.menuSaveTemAsPr.Index = 6;
this.menuSaveTemAsPr.Text = "Save Template as New Project";
this.menuSaveTemAsPr.Click += new System.EventHandler(this.menuSaveTemAsPr_Click);
//
// menuItem_as_NewTempl
//
this.menuItem_as_NewTempl.Index = 7;
this.menuItem_as_NewTempl.Text = "Save Project as New Template";
this.menuItem_as_NewTempl.Click += new System.EventHandler(this.menuItem_as_NewTempl_Click);
//
// menuItemSaveTreeIm
//
this.menuItemSaveTreeIm.Index = 8;
```

```
        this.menuItemSaveTreeIm.Text = "Save Tree as Image File";
        this.menuItemSaveTreeIm.Click += new System.EventHandler(this.menuItemSaveTreeIm_Click);
        //
        // menuItem2
        //
        this.menuItem2.Index = 9;
        this.menuItem2.Text = "-";
        //
        // menuItemListCountm
        //
        this.menuItemListCountm.Index = 10;
        this.menuItemListCountm.Text = "List Countermeasers";
        this.menuItemListCountm.Click += new System.EventHandler(this.menuItemListCountm_Click);
        //
        // menuItemListAttacks
        //
        this.menuItemListAttacks.Index = 11;
        this.menuItemListAttacks.Text = "List Attacks";
        this.menuItemListAttacks.Click += new System.EventHandler(this.menuItemListAttacks_Click);
        //
        // menuItemProperty
        //
        this.menuItemProperty.Index = 12;
        this.menuItemProperty.Text = "Property";
        this.menuItemProperty.Click += new System.EventHandler(this.menuItem12_Click);
        //
        // panel_wait
        //
        this.panel_wait.BackColor = System.Drawing.SystemColors.ActiveCaption;
        this.panel_wait.Controls.Add(this.label1);
        this.panel_wait.Controls.Add(this.progressBar1);
        this.panel_wait.Location = new System.Drawing.Point(226, 190);
        this.panel_wait.Name = "panel_wait";
        this.panel_wait.Size = new System.Drawing.Size(429, 53);
        this.panel_wait.TabIndex = 53;
        this.panel_wait.Visible = false;
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.label1.ForeColor = System.Drawing.SystemColors.HighlightText;
        this.label1.Location = new System.Drawing.Point(3, 16);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(84, 13);
        this.label1.TabIndex = 54;
        this.label1.Text = "Please wait...";
        //
        // progressBar1
        //
        this.progressBar1.AccessibleDescription = "Please";
        this.progressBar1.Location = new System.Drawing.Point(96, 16);
        this.progressBar1.Name = "progressBar1";
        this.progressBar1.Size = new System.Drawing.Size(320, 18);
        this.progressBar1.TabIndex = 53;
        this.progressBar1.UseWaitCursor = true;
        //
        // paGrProperty
        //
```

```
  this.paGrProperty.BackColor = System.Drawing.SystemColors.InactiveCaptionText;
this.paGrProperty.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
this.paGrProperty.Controls.Add(this.settingsDataGridView);
this.paGrProperty.Controls.Add(this.laSettings_Now);
this.paGrProperty.Controls.Add(this.label2);
this.paGrProperty.Controls.Add(this.comBox_Settings);
this.paGrProperty.Controls.Add(this.bnPan_close);
this.paGrProperty.Controls.Add(this.bnSaveSettings);
this.paGrProperty.Location = new System.Drawing.Point(443, 30);
this.paGrProperty.Name = "paGrProperty";
this.paGrProperty.Size = new System.Drawing.Size(526, 131);
this.paGrProperty.TabIndex = 55;
this.paGrProperty.Visible = false;
//
// settingsDataGridView
//
this.settingsDataGridView.AutoGenerateColumns = false;
this.settingsDataGridView.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
this.dataGridViewTextBoxColumn1,
this.dataGridViewTextBoxColumn2,
this.dataGridViewTextBoxColumn3,
this.dataGridViewTextBoxColumn4,
this.dataGridViewTextBoxColumn5,
this.dataGridViewTextBoxColumn6,
this.dataGridViewTextBoxColumn9});
this.settingsDataGridView.DataSource = this.settingsBindingSource;
this.settingsDataGridView.Location = new System.Drawing.Point(242, 23);
this.settingsDataGridView.Name = "settingsDataGridView";
this.settingsDataGridView.Size = new System.Drawing.Size(140, 81);
this.settingsDataGridView.TabIndex = 50;
this.settingsDataGridView.Visible = false;
//
// dataGridViewTextBoxColumn1
//
this.dataGridViewTextBoxColumn1.DataPropertyName = "s_id";
this.dataGridViewTextBoxColumn1.HeaderText = "s_id";
this.dataGridViewTextBoxColumn1.Name = "dataGridViewTextBoxColumn1";
//
// dataGridViewTextBoxColumn2
//
this.dataGridViewTextBoxColumn2.DataPropertyName = "s_name";
this.dataGridViewTextBoxColumn2.HeaderText = "s_name";
this.dataGridViewTextBoxColumn2.Name = "dataGridViewTextBoxColumn2";
//
// dataGridViewTextBoxColumn3
//
this.dataGridViewTextBoxColumn3.DataPropertyName = "s_connect_type";
this.dataGridViewTextBoxColumn3.HeaderText = "s_connect_type";
this.dataGridViewTextBoxColumn3.Name = "dataGridViewTextBoxColumn3";
//
// dataGridViewTextBoxColumn4
//
this.dataGridViewTextBoxColumn4.DataPropertyName = "s_color_asset";
this.dataGridViewTextBoxColumn4.HeaderText = "s_color_asset";
this.dataGridViewTextBoxColumn4.Name = "dataGridViewTextBoxColumn4";
//
// dataGridViewTextBoxColumn5
//
this.dataGridViewTextBoxColumn5.DataPropertyName = "s_color_com";
this.dataGridViewTextBoxColumn5.HeaderText = "s_color_com";
```

```
        this.dataGridViewTextBoxColumn5.Name = "dataGridViewTextBoxColumn5";
        //
        // dataGridViewTextBoxColumn6
        //
        this.dataGridViewTextBoxColumn6.DataPropertyName = "s_color_vuln";
        this.dataGridViewTextBoxColumn6.HeaderText = "s_color_vuln";
        this.dataGridViewTextBoxColumn6.Name = "dataGridViewTextBoxColumn6";
        //
        // dataGridViewTextBoxColumn9
        //
        this.dataGridViewTextBoxColumn9.DataPropertyName = "s_color_back";
        this.dataGridViewTextBoxColumn9.HeaderText = "s_color_back";
        this.dataGridViewTextBoxColumn9.Name = "dataGridViewTextBoxColumn9";
        //
        // settingsBindingSource
        //
        this.settingsBindingSource.DataMember = "settings";
        this.settingsBindingSource.DataSource = this.db_vt_1DataSet;
        //
        // db_vt_1DataSet
        //
        this.db_vt_1DataSet.DataSetName = "db_vt_1DataSet";
        this.db_vt_1DataSet.SchemaSerializationMode = System.Data.SchemaSerializationMode.IncludeSchema;
        //
        // laSettings_Now
        //
        this.laSettings_Now.AutoSize = true;
        this.laSettings_Now.Font = new System.Drawing.Font("Microsoft Sans Serif", 14F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.laSettings_Now.Location = new System.Drawing.Point(129, 8);
        this.laSettings_Now.Name = "laSettings_Now";
        this.laSettings_Now.Size = new System.Drawing.Size(70, 24);
        this.laSettings_Now.TabIndex = 50;
        this.laSettings_Now.Text = "_____";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 14F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.label2.Location = new System.Drawing.Point(10, 8);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(90, 24);
        this.label2.TabIndex = 49;
        this.label2.Text = "Currently:";
        //
        // comBox_Settings
        //
        this.comBox_Settings.Font = new System.Drawing.Font("Microsoft Sans Serif", 14F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.comBox_Settings.FormattingEnabled = true;
        this.comBox_Settings.Location = new System.Drawing.Point(3, 39);
        this.comBox_Settings.Name = "comBox_Settings";
        this.comBox_Settings.Size = new System.Drawing.Size(206, 32);
        this.comBox_Settings.TabIndex = 48;
        //
        // bnPan_close
        //
        this.bnPan_close.Location = new System.Drawing.Point(125, 77);
        this.bnPan_close.Name = "bnPan_close";
```

```
  this.bnPan_close.Size = new System.Drawing.Size(84, 29);
this.bnPan_close.TabIndex = 2;
this.bnPan_close.Text = "Close";
this.bnPan_close.UseVisualStyleBackColor = true;
this.bnPan_close.Click += new System.EventHandler(this.bnPan_close_Click_1);
//
// bnSaveSettings
//
this.bnSaveSettings.Location = new System.Drawing.Point(5, 77);
this.bnSaveSettings.Name = "bnSaveSettings";
this.bnSaveSettings.Size = new System.Drawing.Size(84, 27);
this.bnSaveSettings.TabIndex = 0;
this.bnSaveSettings.Text = "Save";
this.bnSaveSettings.UseVisualStyleBackColor = true;
this.bnSaveSettings.Click += new System.EventHandler(this.bnSaveSettings_Click);
//
// menuStrip1
//
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.saveToolStripMenuItem,
this.saveGraphicToFileToolStripMenuItem,
this.toolStripMenuItemToHistory,
this.menuView,
this.toolStripMenuItemSettings,
this.exitToolStripMenuItem});
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Size = new System.Drawing.Size(969, 24);
this.menuStrip1.TabIndex = 56;
this.menuStrip1.Text = "menuStrip1";
//
// saveToolStripMenuItem
//
this.saveToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.saveAllToolStripMenuItem,
this.saveAsNewTemplateToolStripMenuItem,
this.saveAsNewProjectToolStripMenuItem});
this.saveToolStripMenuItem.Name = "saveToolStripMenuItem";
this.saveToolStripMenuItem.Size = new System.Drawing.Size(46, 20);
this.saveToolStripMenuItem.Text = "Save ";
this.saveToolStripMenuItem.Click += new System.EventHandler(this.saveAllToolStripMenuItem_Click);
//
// saveAllToolStripMenuItem
//
this.saveAllToolStripMenuItem.Name = "saveAllToolStripMenuItem";
this.saveAllToolStripMenuItem.Size = new System.Drawing.Size(183, 22);
this.saveAllToolStripMenuItem.Text = "Save All";
//
// saveAsNewTemplateToolStripMenuItem
//
this.saveAsNewTemplateToolStripMenuItem.Name = "saveAsNewTemplateToolStripMenuItem";
this.saveAsNewTemplateToolStripMenuItem.Size = new System.Drawing.Size(183, 22);
this.saveAsNewTemplateToolStripMenuItem.Text = "Save as New Template";
//
// saveAsNewProjectToolStripMenuItem
//
this.saveAsNewProjectToolStripMenuItem.Name = "saveAsNewProjectToolStripMenuItem";
this.saveAsNewProjectToolStripMenuItem.Size = new System.Drawing.Size(183, 22);
this.saveAsNewProjectToolStripMenuItem.Text = "Save as New Project";
//
```

```
        // saveGraphicToFileToolStripMenuItem
        //
        this.saveGraphicToFileToolStripMenuItem.Name = "saveGraphicToFileToolStripMenuItem";
        this.saveGraphicToFileToolStripMenuItem.Size = new System.Drawing.Size(134, 20);
        this.saveGraphicToFileToolStripMenuItem.Text = "Save Tree as Image File";
        this.saveGraphicToFileToolStripMenuItem.Click += new
System.EventHandler(this.saveGraphicToFileToolStripMenuItem_Click);
        //
        // toolStripMenuItemToHistory
        //
        this.toolStripMenuItemToHistory.Name = "toolStripMenuItemToHistory";
        this.toolStripMenuItemToHistory.Size = new System.Drawing.Size(68, 20);
        this.toolStripMenuItemToHistory.Text = "To History";
        this.toolStripMenuItemToHistory.Click += new
System.EventHandler(this.toolStripMenuItemToHistory_Click);
        //
        // menuView
        //
        this.menuView.Name = "menuView";
        this.menuView.Size = new System.Drawing.Size(41, 20);
        this.menuView.Text = "View";
        //
        // toolStripMenuItemSettings
        //
        this.toolStripMenuItemSettings.Name = "toolStripMenuItemSettings";
        this.toolStripMenuItemSettings.Size = new System.Drawing.Size(58, 20);
        this.toolStripMenuItemSettings.Text = "Settings";
        this.toolStripMenuItemSettings.Click += new System.EventHandler(this.toolStripMenuItemSettings_Click);
        //
        // exitToolStripMenuItem
        //
        this.exitToolStripMenuItem.Name = "exitToolStripMenuItem";
        this.exitToolStripMenuItem.Size = new System.Drawing.Size(37, 20);
        this.exitToolStripMenuItem.Text = "Exit";
        this.exitToolStripMenuItem.Click += new System.EventHandler(this.exitToolStripMenuItem_Click);
        //
        // panel_Lib
        //
        this.panel_Lib.Controls.Add(this.lbTableName);
        this.panel_Lib.Controls.Add(this.rBn_without_dep);
        this.panel_Lib.Controls.Add(this.rBn_with_dep);
        this.panel_Lib.Controls.Add(this.library_coun_4a_vDataGridView);
        this.panel_Lib.Controls.Add(this.library_v_aDataGridView);
        this.panel_Lib.Controls.Add(this.library_comp_vulnDataGridView);
        this.panel_Lib.Controls.Add(this.library_compsDataGridView);
        this.panel_Lib.Controls.Add(this.bnClosePan);
        this.panel_Lib.Controls.Add(this.library_countsDataGridView);
        this.panel_Lib.Controls.Add(this.library_attacksDataGridView);
        this.panel_Lib.Controls.Add(this.library_vulnsDataGridView);
        this.panel_Lib.Location = new System.Drawing.Point(12, 62);
        this.panel_Lib.Name = "panel_Lib";
        this.panel_Lib.Size = new System.Drawing.Size(425, 841);
        this.panel_Lib.TabIndex = 57;
        this.panel_Lib.Visible = false;
        //
        // lbTableName
        //
        this.lbTableName.AutoSize = true;
        this.lbTableName.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
```

```
            this.lbTableName.Location = new System.Drawing.Point(39, 33);
            this.lbTableName.Name = "lbTableName";
            this.lbTableName.Size = new System.Drawing.Size(118, 24);
            this.lbTableName.TabIndex = 61;
            this.lbTableName.Text = "Components";
            //
            // rBn_without_dep
            //
            this.rBn_without_dep.AutoSize = true;
            this.rBn_without_dep.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.rBn_without_dep.Location = new System.Drawing.Point(188, 4);
            this.rBn_without_dep.Name = "rBn_without_dep";
            this.rBn_without_dep.Size = new System.Drawing.Size(162, 20);
            this.rBn_without_dep.TabIndex = 60;
            this.rBn_without_dep.Text = "Without Dependences ";
            this.rBn_without_dep.UseVisualStyleBackColor = true;
            //
            // rBn_with_dep
            //
            this.rBn_with_dep.AutoSize = true;
            this.rBn_with_dep.Checked = true;
            this.rBn_with_dep.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.rBn_with_dep.Location = new System.Drawing.Point(43, 5);
            this.rBn_with_dep.Name = "rBn_with_dep";
            this.rBn_with_dep.Size = new System.Drawing.Size(144, 20);
            this.rBn_with_dep.TabIndex = 59;
            this.rBn_with_dep.TabStop = true;
            this.rBn_with_dep.Text = "With Dependences ";
            this.rBn_with_dep.UseVisualStyleBackColor = true;
            //
            // library_coun_4a_vDataGridView
            //
            this.library_coun_4a_vDataGridView.AutoGenerateColumns = false;
            this.library_coun_4a_vDataGridView.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
            this.dataGridViewTextBoxColumn58,
            this.dataGridViewTextBoxColumn59,
            this.dataGridViewTextBoxColumn60,
            this.dataGridViewTextBoxColumn61});
            this.library_coun_4a_vDataGridView.DataSource = this.library_coun_4a_vBindingSource;
            this.library_coun_4a_vDataGridView.Location = new System.Drawing.Point(258, 340);
            this.library_coun_4a_vDataGridView.Name = "library_coun_4a_vDataGridView";
            this.library_coun_4a_vDataGridView.Size = new System.Drawing.Size(167, 78);
            this.library_coun_4a_vDataGridView.TabIndex = 58;
            //
            // dataGridViewTextBoxColumn58
            //
            this.dataGridViewTextBoxColumn58.DataPropertyName = "co_id";
            this.dataGridViewTextBoxColumn58.HeaderText = "co_id";
            this.dataGridViewTextBoxColumn58.Name = "dataGridViewTextBoxColumn58";
            //
            // dataGridViewTextBoxColumn59
            //
            this.dataGridViewTextBoxColumn59.DataPropertyName = "v_id";
            this.dataGridViewTextBoxColumn59.HeaderText = "v_id";
            this.dataGridViewTextBoxColumn59.Name = "dataGridViewTextBoxColumn59";
            //
            // dataGridViewTextBoxColumn60
```

```
        //
        this.dataGridViewTextBoxColumn60.DataPropertyName = "at_id";
        this.dataGridViewTextBoxColumn60.HeaderText = "at_id";
        this.dataGridViewTextBoxColumn60.Name = "dataGridViewTextBoxColumn60";
        //
        // dataGridViewTextBoxColumn61
        //
        this.dataGridViewTextBoxColumn61.DataPropertyName = "random";
        this.dataGridViewTextBoxColumn61.HeaderText = "random";
        this.dataGridViewTextBoxColumn61.Name = "dataGridViewTextBoxColumn61";
        //
        // library_coun_4a_vBindingSource
        //
        this.library_coun_4a_vBindingSource.DataMember = "library_coun_4a_v";
        this.library_coun_4a_vBindingSource.DataSource = this.db_vt_1DataSet;
        //
        // library_v_aDataGridView
        //
        this.library_v_aDataGridView.AutoGenerateColumns = false;
        this.library_v_aDataGridView.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
        this.dataGridViewTextBoxColumn55,
        this.dataGridViewTextBoxColumn56,
        this.dataGridViewTextBoxColumn57});
        this.library_v_aDataGridView.DataSource = this.library_v_aBindingSource;
        this.library_v_aDataGridView.Location = new System.Drawing.Point(3, 264);
        this.library_v_aDataGridView.Name = "library_v_aDataGridView";
        this.library_v_aDataGridView.Size = new System.Drawing.Size(300, 73);
        this.library_v_aDataGridView.TabIndex = 58;
        //
        // dataGridViewTextBoxColumn55
        //
        this.dataGridViewTextBoxColumn55.DataPropertyName = "vul_id";
        this.dataGridViewTextBoxColumn55.HeaderText = "vul_id";
        this.dataGridViewTextBoxColumn55.Name = "dataGridViewTextBoxColumn55";
        //
        // dataGridViewTextBoxColumn56
        //
        this.dataGridViewTextBoxColumn56.DataPropertyName = "at_id";
        this.dataGridViewTextBoxColumn56.HeaderText = "at_id";
        this.dataGridViewTextBoxColumn56.Name = "dataGridViewTextBoxColumn56";
        //
        // dataGridViewTextBoxColumn57
        //
        this.dataGridViewTextBoxColumn57.DataPropertyName = "at_random";
        this.dataGridViewTextBoxColumn57.HeaderText = "at_random";
        this.dataGridViewTextBoxColumn57.Name = "dataGridViewTextBoxColumn57";
        //
        // library_v_aBindingSource
        //
        this.library_v_aBindingSource.DataMember = "library_v_a";
        this.library_v_aBindingSource.DataSource = this.db_vt_1DataSet;
        //
        // library_comp_vulnDataGridView
        //
        this.library_comp_vulnDataGridView.AutoGenerateColumns = false;
        this.library_comp_vulnDataGridView.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
        this.dataGridViewTextBoxColumn53,
        this.dataGridViewTextBoxColumn54});
        this.library_comp_vulnDataGridView.DataSource = this.library_comp_vulnBindingSource;
```

```
            this.library_comp_vulnDataGridView.Location = new System.Drawing.Point(3, 128);
            this.library_comp_vulnDataGridView.Name = "library_comp_vulnDataGridView";
            this.library_comp_vulnDataGridView.Size = new System.Drawing.Size(300, 51);
            this.library_comp_vulnDataGridView.TabIndex = 58;
            //
            // dataGridViewTextBoxColumn53
            //
            this.dataGridViewTextBoxColumn53.DataPropertyName = "c_id";
            this.dataGridViewTextBoxColumn53.HeaderText = "c_id";
            this.dataGridViewTextBoxColumn53.Name = "dataGridViewTextBoxColumn53";
            //
            // dataGridViewTextBoxColumn54
            //
            this.dataGridViewTextBoxColumn54.DataPropertyName = "vul_id";
            this.dataGridViewTextBoxColumn54.HeaderText = "vul_id";
            this.dataGridViewTextBoxColumn54.Name = "dataGridViewTextBoxColumn54";
            //
            // library_comp_vulnBindingSource
            //
            this.library_comp_vulnBindingSource.DataMember = "library_comp_vuln";
            this.library_comp_vulnBindingSource.DataSource = this.db_vt_1DataSet;
            //
            // library_compsDataGridView
            //
            this.library_compsDataGridView.AutoGenerateColumns = false;
            this.library_compsDataGridView.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[]
{
            this.dataGridViewTextBoxColumn44,
            this.dataGridViewTextBoxColumn45,
            this.dataGridViewTextBoxColumn46,
            this.dataGridViewTextBoxColumn47,
            this.dataGridViewTextBoxColumn48,
            this.dataGridViewTextBoxColumn49,
            this.dataGridViewTextBoxColumn50,
            this.dataGridViewTextBoxColumn51,
            this.dataGridViewCheckBoxColumn3,
            this.dataGridViewTextBoxColumn52});
            this.library_compsDataGridView.DataSource = this.library_compsBindingSource;
            this.library_compsDataGridView.Location = new System.Drawing.Point(3, 60);
            this.library_compsDataGridView.Name = "library_compsDataGridView";
            this.library_compsDataGridView.Size = new System.Drawing.Size(402, 69);
            this.library_compsDataGridView.TabIndex = 58;
            this.library_compsDataGridView.Visible = false;
            this.library_compsDataGridView.DoubleClick += new
System.EventHandler(this.library_compsDataGridView_DoubleClick);
            //
            // dataGridViewTextBoxColumn44
            //
            this.dataGridViewTextBoxColumn44.DataPropertyName = "c_id";
            this.dataGridViewTextBoxColumn44.HeaderText = "c_id";
            this.dataGridViewTextBoxColumn44.Name = "dataGridViewTextBoxColumn44";
            //
            // dataGridViewTextBoxColumn45
            //
            this.dataGridViewTextBoxColumn45.DataPropertyName = "c_name";
            this.dataGridViewTextBoxColumn45.HeaderText = "c_name";
            this.dataGridViewTextBoxColumn45.Name = "dataGridViewTextBoxColumn45";
            //
            // dataGridViewTextBoxColumn46
            //
```

```
this.dataGridViewTextBoxColumn46.DataPropertyName = "c_desc";
this.dataGridViewTextBoxColumn46.HeaderText = "c_desc";
this.dataGridViewTextBoxColumn46.Name = "dataGridViewTextBoxColumn46";
//
// dataGridViewTextBoxColumn47
//
this.dataGridViewTextBoxColumn47.DataPropertyName = "c_weight";
this.dataGridViewTextBoxColumn47.HeaderText = "c_weight";
this.dataGridViewTextBoxColumn47.Name = "dataGridViewTextBoxColumn47";
//
// dataGridViewTextBoxColumn48
//
this.dataGridViewTextBoxColumn48.DataPropertyName = "c_critic";
this.dataGridViewTextBoxColumn48.HeaderText = "c_critic";
this.dataGridViewTextBoxColumn48.Name = "dataGridViewTextBoxColumn48";
//
// dataGridViewTextBoxColumn49
//
this.dataGridViewTextBoxColumn49.DataPropertyName = "c_cap_cost";
this.dataGridViewTextBoxColumn49.HeaderText = "c_cap_cost";
this.dataGridViewTextBoxColumn49.Name = "dataGridViewTextBoxColumn49";
//
// dataGridViewTextBoxColumn50
//
this.dataGridViewTextBoxColumn50.DataPropertyName = "c_final_risk";
this.dataGridViewTextBoxColumn50.HeaderText = "c_final_risk";
this.dataGridViewTextBoxColumn50.Name = "dataGridViewTextBoxColumn50";
//
// dataGridViewTextBoxColumn51
//
this.dataGridViewTextBoxColumn51.DataPropertyName = "c_ecl";
this.dataGridViewTextBoxColumn51.HeaderText = "c_ecl";
this.dataGridViewTextBoxColumn51.Name = "dataGridViewTextBoxColumn51";
//
// dataGridViewCheckBoxColumn3
//
this.dataGridViewCheckBoxColumn3.DataPropertyName = "templ";
this.dataGridViewCheckBoxColumn3.HeaderText = "templ";
this.dataGridViewCheckBoxColumn3.Name = "dataGridViewCheckBoxColumn3";
//
// dataGridViewTextBoxColumn52
//
this.dataGridViewTextBoxColumn52.DataPropertyName = "c_random";
this.dataGridViewTextBoxColumn52.HeaderText = "c_random";
this.dataGridViewTextBoxColumn52.Name = "dataGridViewTextBoxColumn52";
//
// library_compsBindingSource
//
this.library_compsBindingSource.DataMember = "library_comps";
this.library_compsBindingSource.DataSource = this.db_vt_1DataSet;
//
// bnClosePan
//
this.bnClosePan.Location = new System.Drawing.Point(356, 2);
this.bnClosePan.Name = "bnClosePan";
this.bnClosePan.Size = new System.Drawing.Size(66, 23);
this.bnClosePan.TabIndex = 58;
this.bnClosePan.Text = "Close";
this.bnClosePan.UseVisualStyleBackColor = true;
this.bnClosePan.Click += new System.EventHandler(this.bnClosePan_Click);
```

```
            //
            // library_countsDataGridView
            //
            this.library_countsDataGridView.AutoGenerateColumns = false;
            this.library_countsDataGridView.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[]
{
            this.dataGridViewTextBoxColumn32,
            this.dataGridViewTextBoxColumn33,
            this.dataGridViewTextBoxColumn34,
            this.dataGridViewTextBoxColumn35,
            this.dataGridViewTextBoxColumn36,
            this.dataGridViewTextBoxColumn37,
            this.dataGridViewTextBoxColumn38,
            this.dataGridViewTextBoxColumn39,
            this.dataGridViewTextBoxColumn40,
            this.dataGridViewTextBoxColumn41,
            this.dataGridViewTextBoxColumn42,
            this.dataGridViewTextBoxColumn43,
            this.dataGridViewCheckBoxColumn2});
            this.library_countsDataGridView.DataSource = this.library_countsBindingSource;
            this.library_countsDataGridView.Location = new System.Drawing.Point(3, 421);
            this.library_countsDataGridView.Name = "library_countsDataGridView";
            this.library_countsDataGridView.Size = new System.Drawing.Size(369, 78);
            this.library_countsDataGridView.TabIndex = 57;
            this.library_countsDataGridView.Visible = false;
            this.library_countsDataGridView.DoubleClick += new
System.EventHandler(this.library_countsDataGridView_DoubleClick);
            //
            // dataGridViewTextBoxColumn32
            //
            this.dataGridViewTextBoxColumn32.DataPropertyName = "co_id";
            this.dataGridViewTextBoxColumn32.HeaderText = "co_id";
            this.dataGridViewTextBoxColumn32.Name = "dataGridViewTextBoxColumn32";
            //
            // dataGridViewTextBoxColumn33
            //
            this.dataGridViewTextBoxColumn33.DataPropertyName = "co_name";
            this.dataGridViewTextBoxColumn33.HeaderText = "co_name";
            this.dataGridViewTextBoxColumn33.Name = "dataGridViewTextBoxColumn33";
            //
            // dataGridViewTextBoxColumn34
            //
            this.dataGridViewTextBoxColumn34.DataPropertyName = "co_desc";
            this.dataGridViewTextBoxColumn34.HeaderText = "co_desc";
            this.dataGridViewTextBoxColumn34.Name = "dataGridViewTextBoxColumn34";
            //
            // dataGridViewTextBoxColumn35
            //
            this.dataGridViewTextBoxColumn35.DataPropertyName = "co_example";
            this.dataGridViewTextBoxColumn35.HeaderText = "co_example";
            this.dataGridViewTextBoxColumn35.Name = "dataGridViewTextBoxColumn35";
            //
            // dataGridViewTextBoxColumn36
            //
            this.dataGridViewTextBoxColumn36.DataPropertyName = "co_weight";
            this.dataGridViewTextBoxColumn36.HeaderText = "co_weight";
            this.dataGridViewTextBoxColumn36.Name = "dataGridViewTextBoxColumn36";
            //
            // dataGridViewTextBoxColumn37
            //
```

```
        this.dataGridViewTextBoxColumn37.DataPropertyName = "co_complexity";
        this.dataGridViewTextBoxColumn37.HeaderText = "co_complexity";
        this.dataGridViewTextBoxColumn37.Name = "dataGridViewTextBoxColumn37";
        //
        // dataGridViewTextBoxColumn38
        //
        this.dataGridViewTextBoxColumn38.DataPropertyName = "v_ids";
        this.dataGridViewTextBoxColumn38.HeaderText = "v_ids";
        this.dataGridViewTextBoxColumn38.Name = "dataGridViewTextBoxColumn38";
        //
        // dataGridViewTextBoxColumn39
        //
        this.dataGridViewTextBoxColumn39.DataPropertyName = "at_ids";
        this.dataGridViewTextBoxColumn39.HeaderText = "at_ids";
        this.dataGridViewTextBoxColumn39.Name = "dataGridViewTextBoxColumn39";
        //
        // dataGridViewTextBoxColumn40
        //
        this.dataGridViewTextBoxColumn40.DataPropertyName = "source";
        this.dataGridViewTextBoxColumn40.HeaderText = "source";
        this.dataGridViewTextBoxColumn40.Name = "dataGridViewTextBoxColumn40";
        //
        // dataGridViewTextBoxColumn41
        //
        this.dataGridViewTextBoxColumn41.DataPropertyName = "pl_id";
        this.dataGridViewTextBoxColumn41.HeaderText = "pl_id";
        this.dataGridViewTextBoxColumn41.Name = "dataGridViewTextBoxColumn41";
        //
        // dataGridViewTextBoxColumn42
        //
        this.dataGridViewTextBoxColumn42.DataPropertyName = "website";
        this.dataGridViewTextBoxColumn42.HeaderText = "website";
        this.dataGridViewTextBoxColumn42.Name = "dataGridViewTextBoxColumn42";
        //
        // dataGridViewTextBoxColumn43
        //
        this.dataGridViewTextBoxColumn43.DataPropertyName = "co_random";
        this.dataGridViewTextBoxColumn43.HeaderText = "co_random";
        this.dataGridViewTextBoxColumn43.Name = "dataGridViewTextBoxColumn43";
        //
        // dataGridViewCheckBoxColumn2
        //
        this.dataGridViewCheckBoxColumn2.DataPropertyName = "templ";
        this.dataGridViewCheckBoxColumn2.HeaderText = "templ";
        this.dataGridViewCheckBoxColumn2.Name = "dataGridViewCheckBoxColumn2";
        //
        // library_countsBindingSource
        //
        this.library_countsBindingSource.DataMember = "library_counts";
        this.library_countsBindingSource.DataSource = this.db_vt_1DataSet;
        //
        // library_attacksDataGridView
        //
        this.library_attacksDataGridView.AutoGenerateColumns = false;
        this.library_attacksDataGridView.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
        this.dataGridViewTextBoxColumn19,
        this.dataGridViewTextBoxColumn20,
        this.dataGridViewTextBoxColumn21,
        this.dataGridViewTextBoxColumn22,
```

```
            this.dataGridViewTextBoxColumn23,
            this.dataGridViewTextBoxColumn24,
            this.dataGridViewTextBoxColumn25,
            this.dataGridViewTextBoxColumn26,
            this.dataGridViewTextBoxColumn27,
            this.dataGridViewTextBoxColumn28,
            this.dataGridViewTextBoxColumn29,
            this.dataGridViewTextBoxColumn30,
            this.dataGridViewCheckBoxColumn1,
            this.dataGridViewTextBoxColumn31});
            this.library_attacksDataGridView.DataSource = this.library_attacksBindingSource;
            this.library_attacksDataGridView.Location = new System.Drawing.Point(0, 340);
            this.library_attacksDataGridView.Name = "library_attacksDataGridView";
            this.library_attacksDataGridView.Size = new System.Drawing.Size(542, 75);
            this.library_attacksDataGridView.TabIndex = 57;
            this.library_attacksDataGridView.Visible = false;
            this.library_attacksDataGridView.DoubleClick += new
System.EventHandler(this.library_attacksDataGridView_DoubleClick);
            //
            // dataGridViewTextBoxColumn19
            //
            this.dataGridViewTextBoxColumn19.DataPropertyName = "at_id";
            this.dataGridViewTextBoxColumn19.HeaderText = "at_id";
            this.dataGridViewTextBoxColumn19.Name = "dataGridViewTextBoxColumn19";
            //
            // dataGridViewTextBoxColumn20
            //
            this.dataGridViewTextBoxColumn20.DataPropertyName = "at_name";
            this.dataGridViewTextBoxColumn20.HeaderText = "at_name";
            this.dataGridViewTextBoxColumn20.Name = "dataGridViewTextBoxColumn20";
            //
            // dataGridViewTextBoxColumn21
            //
            this.dataGridViewTextBoxColumn21.DataPropertyName = "at_desc";
            this.dataGridViewTextBoxColumn21.HeaderText = "at_desc";
            this.dataGridViewTextBoxColumn21.Name = "dataGridViewTextBoxColumn21";
            //
            // dataGridViewTextBoxColumn22
            //
            this.dataGridViewTextBoxColumn22.DataPropertyName = "at_type_id";
            this.dataGridViewTextBoxColumn22.HeaderText = "at_type_id";
            this.dataGridViewTextBoxColumn22.Name = "dataGridViewTextBoxColumn22";
            //
            // dataGridViewTextBoxColumn23
            //
            this.dataGridViewTextBoxColumn23.DataPropertyName = "at_weight";
            this.dataGridViewTextBoxColumn23.HeaderText = "at_weight";
            this.dataGridViewTextBoxColumn23.Name = "dataGridViewTextBoxColumn23";
            //
            // dataGridViewTextBoxColumn24
            //
            this.dataGridViewTextBoxColumn24.DataPropertyName = "source";
            this.dataGridViewTextBoxColumn24.HeaderText = "source";
            this.dataGridViewTextBoxColumn24.Name = "dataGridViewTextBoxColumn24";
            //
            // dataGridViewTextBoxColumn25
            //
            this.dataGridViewTextBoxColumn25.DataPropertyName = "at_easiness";
            this.dataGridViewTextBoxColumn25.HeaderText = "at_easiness";
            this.dataGridViewTextBoxColumn25.Name = "dataGridViewTextBoxColumn25";
```

```
//
// dataGridViewTextBoxColumn26
//
this.dataGridViewTextBoxColumn26.DataPropertyName = "at_severity";
this.dataGridViewTextBoxColumn26.HeaderText = "at_severity";
this.dataGridViewTextBoxColumn26.Name = "dataGridViewTextBoxColumn26";
//
// dataGridViewTextBoxColumn27
//
this.dataGridViewTextBoxColumn27.DataPropertyName = "at_example";
this.dataGridViewTextBoxColumn27.HeaderText = "at_example";
this.dataGridViewTextBoxColumn27.Name = "dataGridViewTextBoxColumn27";
//
// dataGridViewTextBoxColumn28
//
this.dataGridViewTextBoxColumn28.DataPropertyName = "at_more_names";
this.dataGridViewTextBoxColumn28.HeaderText = "at_more_names";
this.dataGridViewTextBoxColumn28.Name = "dataGridViewTextBoxColumn28";
//
// dataGridViewTextBoxColumn29
//
this.dataGridViewTextBoxColumn29.DataPropertyName = "at_web_site";
this.dataGridViewTextBoxColumn29.HeaderText = "at_web_site";
this.dataGridViewTextBoxColumn29.Name = "dataGridViewTextBoxColumn29";
//
// dataGridViewTextBoxColumn30
//
this.dataGridViewTextBoxColumn30.DataPropertyName = "at_test";
this.dataGridViewTextBoxColumn30.HeaderText = "at_test";
this.dataGridViewTextBoxColumn30.Name = "dataGridViewTextBoxColumn30";
//
// dataGridViewCheckBoxColumn1
//
this.dataGridViewCheckBoxColumn1.DataPropertyName = "templ";
this.dataGridViewCheckBoxColumn1.HeaderText = "templ";
this.dataGridViewCheckBoxColumn1.Name = "dataGridViewCheckBoxColumn1";
//
// dataGridViewTextBoxColumn31
//
this.dataGridViewTextBoxColumn31.DataPropertyName = "at_random";
this.dataGridViewTextBoxColumn31.HeaderText = "at_random";
this.dataGridViewTextBoxColumn31.Name = "dataGridViewTextBoxColumn31";
//
// library_attacksBindingSource
//
this.library_attacksBindingSource.DataMember = "library_attacks";
this.library_attacksBindingSource.DataSource = this.db_vt_1DataSet;
//
// library_vulnsDataGridView
//
this.library_vulnsDataGridView.AutoGenerateColumns = false;
this.library_vulnsDataGridView.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
this.dataGridViewTextBoxColumn10,
this.dataGridViewTextBoxColumn11,
this.dataGridViewTextBoxColumn12,
this.dataGridViewTextBoxColumn13,
this.dataGridViewTextBoxColumn14,
this.dataGridViewTextBoxColumn15,
this.dataGridViewTextBoxColumn16,
this.dataGridViewTextBoxColumn17,
```

```
            this.dataGridViewTextBoxColumn18});
        this.library_vulnsDataGridView.DataSource = this.library_vulnsBindingSource;
        this.library_vulnsDataGridView.Location = new System.Drawing.Point(3, 185);
        this.library_vulnsDataGridView.Name = "library_vulnsDataGridView";
        this.library_vulnsDataGridView.Size = new System.Drawing.Size(332, 73);
        this.library_vulnsDataGridView.TabIndex = 57;
        this.library_vulnsDataGridView.Visible = false;
        this.library_vulnsDataGridView.DoubleClick += new
System.EventHandler(this.library_vulnsDataGridView_DoubleClick);
        //
        // dataGridViewTextBoxColumn10
        //
        this.dataGridViewTextBoxColumn10.DataPropertyName = "vul_id";
        this.dataGridViewTextBoxColumn10.HeaderText = "vul_id";
        this.dataGridViewTextBoxColumn10.Name = "dataGridViewTextBoxColumn10";
        //
        // dataGridViewTextBoxColumn11
        //
        this.dataGridViewTextBoxColumn11.DataPropertyName = "v_name";
        this.dataGridViewTextBoxColumn11.HeaderText = "v_name";
        this.dataGridViewTextBoxColumn11.Name = "dataGridViewTextBoxColumn11";
        //
        // dataGridViewTextBoxColumn12
        //
        this.dataGridViewTextBoxColumn12.DataPropertyName = "v_desc";
        this.dataGridViewTextBoxColumn12.HeaderText = "v_desc";
        this.dataGridViewTextBoxColumn12.Name = "dataGridViewTextBoxColumn12";
        //
        // dataGridViewTextBoxColumn13
        //
        this.dataGridViewTextBoxColumn13.DataPropertyName = "v_weight";
        this.dataGridViewTextBoxColumn13.HeaderText = "v_weight";
        this.dataGridViewTextBoxColumn13.Name = "dataGridViewTextBoxColumn13";
        //
        // dataGridViewTextBoxColumn14
        //
        this.dataGridViewTextBoxColumn14.DataPropertyName = "vt_id";
        this.dataGridViewTextBoxColumn14.HeaderText = "vt_id";
        this.dataGridViewTextBoxColumn14.Name = "dataGridViewTextBoxColumn14";
        //
        // dataGridViewTextBoxColumn15
        //
        this.dataGridViewTextBoxColumn15.DataPropertyName = "v_example";
        this.dataGridViewTextBoxColumn15.HeaderText = "v_example";
        this.dataGridViewTextBoxColumn15.Name = "dataGridViewTextBoxColumn15";
        //
        // dataGridViewTextBoxColumn16
        //
        this.dataGridViewTextBoxColumn16.DataPropertyName = "v_risk";
        this.dataGridViewTextBoxColumn16.HeaderText = "v_risk";
        this.dataGridViewTextBoxColumn16.Name = "dataGridViewTextBoxColumn16";
        //
        // dataGridViewTextBoxColumn17
        //
        this.dataGridViewTextBoxColumn17.DataPropertyName = "v_random";
        this.dataGridViewTextBoxColumn17.HeaderText = "v_random";
        this.dataGridViewTextBoxColumn17.Name = "dataGridViewTextBoxColumn17";
        //
        // dataGridViewTextBoxColumn18
        //
```

```
this.dataGridViewTextBoxColumn18.DataPropertyName = "v_platform";
this.dataGridViewTextBoxColumn18.HeaderText = "v_platform";
this.dataGridViewTextBoxColumn18.Name = "dataGridViewTextBoxColumn18";
//
// library_vulnsBindingSource
//
this.library_vulnsBindingSource.DataMember = "library_vulns";
this.library_vulnsBindingSource.DataSource = this.db_vt_1DataSet;
//
// settingsTableAdapter
//
this.settingsTableAdapter.ClearBeforeFill = true;
//
// library_vulnsTableAdapter
//
this.library_vulnsTableAdapter.ClearBeforeFill = true;
//
// library_attacksTableAdapter
//
this.library_attacksTableAdapter.ClearBeforeFill = true;
//
// library_countsTableAdapter
//
this.library_countsTableAdapter.ClearBeforeFill = true;
//
// library_compsTableAdapter
//
this.library_compsTableAdapter.ClearBeforeFill = true;
//
// library_comp_vulnTableAdapter
//
this.library_comp_vulnTableAdapter.ClearBeforeFill = true;
//
// library_v_aTableAdapter
//
this.library_v_aTableAdapter.ClearBeforeFill = true;
//
// library_coun_4a_vTableAdapter
//
this.library_coun_4a_vTableAdapter.ClearBeforeFill = true;
//
// Tree
//
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.AutoScroll = true;
this.ClientSize = new System.Drawing.Size(936, 633);
this.Controls.Add(this.panel_Lib);
this.Controls.Add(this.menuStrip1);
this.Controls.Add(this.paGrProperty);
this.Controls.Add(this.panel_wait);
this.Controls.Add(this.TreePicture);
this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
this.Name = "Tree";
this.StartPosition = System.Windows.Forms.FormStartPosition.Manual;
this.Text = "Tree";
this.Load += new System.EventHandler(this.Schema_Load);
((System.ComponentModel.ISupportInitialize)(this.TreePicture)).EndInit();
this.panel_wait.ResumeLayout(false);
this.panel_wait.PerformLayout();
this.paGrProperty.ResumeLayout(false);
```

```
    this.paGrProperty.PerformLayout();
    ((System.ComponentModel.ISupportInitialize)(this.settingsDataGridView)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.settingsBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.db_vt_1DataSet)).EndInit();
    this.menuStrip1.ResumeLayout(false);
    this.menuStrip1.PerformLayout();
    this.panel_Lib.ResumeLayout(false);
    this.panel_Lib.PerformLayout();
    ((System.ComponentModel.ISupportInitialize)(this.library_coun_4a_vDataGridView)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_coun_4a_vBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_v_aDataGridView)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_v_aBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_comp_vulnDataGridView)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_comp_vulnBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_compsDataGridView)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_compsBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_countsDataGridView)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_countsBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_attacksDataGridView)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_attacksBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_vulnsDataGridView)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.library_vulnsBindingSource)).EndInit();
    this.ResumeLayout(false);
    this.PerformLayout();

                }
                #endregion

                #region Servece functions & procedures
/// <summary>
/// Build a random string (for id, login, password...)
/// </summary>
public static string randomString()
{
    int length = new Random().Next(6, 10);
    return randomString(length);
}

/// <summary>
/// Build a random string (for id, login, password...)
/// </summary>
public static string randomString(int length)
{
    string tempString = Guid.NewGuid().ToString().ToLower();
    tempString = tempString.Replace("-", "");
    while (tempString.Length < length)
    {
        tempString += tempString;
    }
    tempString = tempString.Substring(0, length);
    return tempString;
}
                /// <summary>
                /// Show property of selected object
                /// </summary>
                public void ShowPropertyShape(int iMode)
                {
    TreeBase frm = (TreeBase)this.MdiParent;
    Panel panel      = frm.panelProperty;
    PropertyGrid pg = frm.propertyGrid;
```

```
                        if(indexSelect > - 1)
                        {
                                int id = shcol.GetIndexSelectObject();
    ShapeBase sb = (ShapeBase)shcol[id];
                                pg.SelectedObject = sb;
                        }

                        if(iMode == 1) panel.Visible    = true;
if (indexSelect == -1)
{
   paGrProperty.Visible = true;
}
                }
                /// <summary>
                /// This procedure deletes selected shape and all child shapes.
                /// </summary>
                /// <param name="sb">
                ///sb - Selected shape
                ///</param>
                public void RemoveShape(ShapeBase sb)
                {
                        //Get index sb
                        int indexSb = shcol.GetIndexShape(sb);
string st = Convert.ToString(indexSb);
                        if (indexSb == -1) return;
                        //Delete parent connection

if (indexSb > 0) // not asset
{
   int indexCon = concol.GetIndexParentConnection(sb.ID);
   if (indexCon > - 1) concol.RemoveObject(indexCon);
}

                        //Delete child shapes
                        for(int i = 0; i < sb.ChildCount; i ++)
                        {
                                ShapeBase csb = (ShapeBase) sb.GetChildShape(i);
                                RemoveShape(csb);
                        }
                        shcol.RemoveObject(indexSb);
                }
                /// <summary>
                ///
                /// </summary>
                /// <param name="indexShape">index ShapeBase
                /// 0 - Asset object
                /// 1 - Component object
                /// 2 - Vulnerability object
                /// 3 - Attack object
                /// 4 - Countermeasure object
                /// </param>
                /// <returns>ShapeBase Point position x</returns>
                public void GetPositionX(int indexShape, int width) //
                {
                        int numberShape = shcol.CountShape(indexShape);
                        //Total shape's unit
                        Rectangle rctUnit = new Rectangle();
                        rctUnit.Height    = 80;
int sum_w = 0;
switch (indexShape)
```

```
 {
  case 1:
    {
      sum_w = sumWidLevel1;
      break;
    }
  case 2:
    {
      sum_w = sumWidLevel2;
      break;
    }
  case 3:
    {
      sum_w = sumWidLevel3 ;
      break;
    }
  case 4:
    {
      sum_w = sumWidLevel4;
      break;
    }
}
rctUnit.Width = sum_w + (10 * (numberShape - 1));
                    rctUnit.Y       = 60 + (indexShape * 100);
                    rctUnit.X       = (int)(TreePicture.Width / 2) - (int)rctUnit.Width/2;
                    shcol.SetNewPositionShape(indexShape, rctUnit);
//indexShape
switch (indexShape)
{
  case 1:
    {
      sumWidLevel1 = sumWidLevel1 + width;
      break;
    }
  case 2:
    {
      sumWidLevel2=sumWidLevel2+ width;
      break;
    }
  case 3:
    {
      sumWidLevel3 = sumWidLevel3 + width;
      break;
    }
  case 4:
    {
      sumWidLevel4=sumWidLevel4 + width;
      break;
    }
}
            }
            /// <summary>
            ///
            /// </summary>
            /// <param name="sp"></param>
            /// <param name="flagVisible"></param>
            public void MakeVisibleShape(ShapeBase sp, bool flagVisible)
            {
                    int intChildShape = sp.ChildCount;
                    if (intChildShape == 0) return;
```

```
            for (int i = 0; i < intChildShape; i ++)
            {
                        ShapeBase cs  = (ShapeBase) sp.GetChildShape(i);
                        Connection cc = (Connection) sp.GetChildConnect(i);
                        cs.Visible    = flagVisible;
                        cs.Surrogate  = false;
                        cc.Visible    = flagVisible;

                        if (cs.ChildCount > 0) MakeVisibleShape(cs, flagVisible);
            }
}
/// <summary>
///
/// </summary>
/// <param name="p"></param>
/// <returns></returns>
private int GetObjectFromSchema(Point p, out int ind)
{
            int result   = - 100;
            //check shape collection
            ind = shcol.GetSelectArea(p);
            if (ind > -1)
            {
                        result = 1;
            }
            //in this place I have to add string for find connect object
            return result;
}
/// <summary>
/// Setup initial value "point", "indexSelect" and intMode"
/// </summary>
private void SetFlagIndex()
{
            for(int i = 0; i < 5; i ++)
            {
                        point[i].X = 0;
                        point[i].Y = 0;
            }

            intMode    = - 100;
            indexSelect = - 100;
}
/// <summary>
/// Return Type of object
/// </summary>
/// <returns>
/// 0 - Asset object
/// 1 - Component object
/// 2 - Vulnerability object
/// 3 - Thread object
/// 4 - LCM object
/// 7 - Connector
/// 8 - TreePicture
/// -1 - Object don't define
/// </returns>
private int GetTypeObjects()
{
            int result = -1;

            if((indexSelect > -100)&(intMode > -100))
```

```
                {
                        switch (intMode)
                        {
                                case 1:
                                {
                                        ShapeBase sb = (ShapeBase)shcol[indexSelect];

                                        if(sb is Asset) result = 0;
                                        if(sb is Component) result = 1;
                                        if(sb is Vulnerability) result = 2;
                                        if(sb is Attack) result = 3;
                                        if(sb is Countermeasure) result = 4;
                                        break;
                                } //end case 1
                        }//end switch
                }//end if

                return result;
        }

/// <summary>
/// Return Graphic ID of object
/// </summary>
/// <returns>
///
/// -1 - Object don't define
/// </returns>
private int GetGIdObject()
{
   int result = -1;

   if ((indexSelect > -100) & (intMode > -100))
   {
      switch (intMode)
      {
         case 1:
            {
               ShapeBase sb = (ShapeBase)shcol[indexSelect];

               if (sb is Asset) result = 0;
               if (sb is Component)
               {
                  Component comp = new Component();
                  comp = (Component)sb;
                  result = comp.gid ;
               }
               if (sb is Vulnerability)
               {
                  Vulnerability vv = new Vulnerability();
                  vv = (Vulnerability)sb;
                  result = vv.gid;
               }
               if (sb is Attack)
               {
                  Attack th = new Attack();
                  th = (Attack)sb;
                  result = th.gid;
               }
               if (sb is Countermeasure)
               {
```

```
                    Countermeasure countm = new Countermeasure();
                    countm = (Countermeasure)sb;
                    result = countm.gid;
                  }
                  break;
                } //end case 1
              }//end switch
            }//end if

            return result;
        }
                    /// <summary>
        /// Create New Shape - new object type will be depends from what object user pointed
                    /// </summary>
                    private void CreateNewShape(int selectShape, int gid_par)
                    {
                              Point position  = new Point();
            TreeBase frm = (TreeBase)this.MdiParent;
                              switch (selectShape)
                              {
                                        case -1:
                                        {

                  if (have_asset== true)
                  {

                    MessageBox.Show("You alredy define asset." );

                  }
                  else
                  {
                    position.X = (int)(TreePicture.Width / 2) - 50;
                    position.Y = 60;
                    SolidBrush asset_brush = new SolidBrush(frm.asset_color);
                    Asset asset = new Asset(position, shcol, this.TreePicture, asset_brush);
                    have_asset = true;
                  }
                  break;
                                        }

                                        case 0:
                                        {

                  Asset asset       = (Asset)shcol[indexSelect];
                  SolidBrush comp_brush = new SolidBrush(frm.comp_color);

                  Component component = new Component(point[0], shcol, this.TreePicture, indexSelect,
comp_brush);
                  component.gid_parent = gid_par;
                  component.id_db_parent = asset_now.a_id;
                  //string s_gid_par = Convert.ToString(component.id_db_parent);
                  component.gid = shcol.Count -1;
                  //string s_gid = Convert.ToString(component.gid);
                  GetPositionX(1, component.RectObject.Width);
                                                  Connection connect  = new Connection(asset, component,
concol);
                                                  asset.AddChild(component, connect);
                                                  break;
                                        }
```

```
                                    case 1:
                                    {
            SolidBrush vuln_brush = new SolidBrush(frm.vuln_color);
            Vulnerability vulnerability = new Vulnerability(point[0], shcol, this.TreePicture, indexSelect,
vuln_brush);
                                        Component component      = (Component) shcol[indexSelect];
            vulnerability.id_db_parent=component.c_id;
            vulnerability.gid_parent = gid_par;
            vulnerability.gid = shcol.Count -1;
                                        Connection connect  = new Connection(component,
vulnerability, concol);
                                        component.AddChild(vulnerability, connect);
            GetPositionX(2, vulnerability.RectObject.Width);
                                        break;
                                    }

                                    case 2:
                                    {
            SolidBrush attack_brush = new SolidBrush(frm.attack_color);

            Attack attack = new Attack(point[0], shcol, this.TreePicture, indexSelect, attack_brush);
                                        Vulnerability vulnerability = (Vulnerability)shcol[indexSelect];
            attack.id_db_parent = vulnerability.v_id;
            attack.vul_id = vulnerability.v_id;

            attack.gid_parent = gid_par;
            attack.gid = shcol.Count-1 ;
            Connection connect = new Connection(vulnerability, attack, concol);
                                        vulnerability.AddChild(attack, connect);
                                        GetPositionX(3, attack.RectObject.Width);
                                        break;
                                    }

                                    case 3:
                                    {
            Attack attack  = (Attack) shcol[indexSelect];

            SolidBrush countm_brush = new SolidBrush(frm.countm_color);
            Countermeasure countm = new Countermeasure(point[0], shcol, this.TreePicture, indexSelect,
countm_brush);
            countm.id_db_parent = attack.at_id;
            countm.gid_parent = gid_par;
            countm.gid = shcol.Count - 1;
            Connection connect = new Connection(attack, countm, concol);
            attack.AddChild(countm, connect);
            GetPositionX(4, countm.RectObject.Width);

                                        break;
                                    }

                                    case 5:
                                    {
                                        break;
                                    }
                            }

                            TreePicture.Invalidate();
                    }
```

```
        /// <summary>
        ///
        /// </summary>
        /// <param name="sb"></param>
        public void GetWidthUnit(ShapeBase sb, int iUnit)
        {
            if (sb.ChildCount > 0)
            {
                int newWeight = 105 * sb.ChildCount;
                if(UnitWigth[iUnit] < newWeight) UnitWigth[iUnit] = newWeight;

                for(int i = 0; i < sb.ChildCount; i ++)
                {
                    GetWidthUnit((ShapeBase)sb.GetChildShape(i), iUnit);
                }
            }

        }
        /// <summary>
        ///
        /// </summary>
        public void GetNumberUnits()
        {
            ShapeBase asset = shcol.GetRoot();
            int iUnit      = asset.ChildCount;
            if(iUnit > 0)
            {
                UnitWigth = new int[iUnit];
            }
            else return;

            for(int i = 0; i < iUnit; i ++)
            {
                UnitWigth[i] = 105;
                GetWidthUnit((ShapeBase)asset.GetChildShape(i), i);
            }

            for(int i = 0; i < iUnit; i ++)
            {
                MessageBox.Show(UnitWigth[i].ToString());
            }

}

public void setNotSaved()
{
   int g_id = shcol.GetIndexSelectObject();
   ShapeBase sb = (ShapeBase)shcol[g_id];
   sb.Saved = false;
}
public bool isShema_Saved(object sender, EventArgs e)
{
   bool result = true;// saved
   for (int i = 0; i < shcol.Count; i++)
   {
      ShapeBase sb = (ShapeBase)shcol[i];
      if (sb.Saved == false) result = false;
   }
   return result;
}
```

```
public void do_security_numbers(object sender, EventArgs e)
{
   int count_arr = 0;
   // need to collect all components for feature comarisons
   string str_tot = "";
   try
   {
      for (int i = 0; i < shcol.Count; i++)
      {
         ShapeBase sb = (ShapeBase)shcol[i];
         if (sb is Component)
         {
            Component comp = new Component();
            comp = (Component)sb;

            array_sec_comp[count_arr, 0] = comp.gid;
            array_sec_comp[count_arr, 1] = comp.Risk;        //ecl;
            string str_ = Convert.ToString(array_sec_comp[count_arr, 1]);
            int ii = Convert.ToInt16(array_sec_comp[count_arr, 0]);
            string sgid = Convert.ToString(ii);
            //str_tot = str_tot + " " + str_ + " " + sgid + "***";
            count_arr++;
         }
      }
      bool swap_done = true;
      double co_tempr = 0.0;
      double co_tempr2 = 0.0;
      while (swap_done == true)
      {
         swap_done = false;
         for (int n = 1; n < count_arr; n++) // we start from second member of array
         {

            if (array_sec_comp[n - 1, 1] < array_sec_comp[n, 1]) // do swap
            {
               co_tempr = array_sec_comp[n - 1, 1];
               co_tempr2 = array_sec_comp[n - 1, 0];
               array_sec_comp[n - 1, 1] = array_sec_comp[n, 1];
               array_sec_comp[n - 1, 0] = array_sec_comp[n, 0];
               array_sec_comp[n, 1] = co_tempr;
               array_sec_comp[n, 0] = co_tempr2;
               swap_done = true;
            }
         }

      } // end while

      //

      for (int n = 0; n < count_arr; n++) //array
      {
         string str_ = Convert.ToString(array_sec_comp[n, 1]);
         int ii = Convert.ToInt16(array_sec_comp[n, 0]);
         string sgid = Convert.ToString(ii);
         str_tot = str_tot + " " + str_ + " " + sgid + "***";
         ShapeBase sb = (ShapeBase)shcol[ii];
         Component comp = new Component();
         comp = (Component)sb;
         comp.sec_num = n + 1;
```

```
            string nna = Convert.ToString(comp.Name);
        }
        for (int n1 = 0; n1 < tot_critic_num; n1++) //array
        {
            int pgid_ = Convert.ToInt16(array_sec_comp[n1, 0]);
            do_security_numbers_vuln(sender, e, pgid_, n1);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(" do_security_numbers:\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
    }

}

public void do_security_numbers_vuln(object sender, EventArgs e, int pgid, int nn)
{
    int count_arr = 0;
    // need to collect all components for feature comarisons
    string str_tot = "";
    // count how many vuln we have for this component
    for (int i = 0; i < shcol.Count; i++)
    {
        ShapeBase sb = (ShapeBase)shcol[i];
        if (sb is Vulnerability)
        {
            Vulnerability vuln = new Vulnerability();
            vuln = (Vulnerability)sb;
            if (vuln.gid_parent == pgid)
            {
                count_arr++;

            }
        }
    }
    array_sec_vuln[nn] = new double[count_arr]; // create count_arr colomns for row nn
    double[,] array_sec_temp = new double[count_arr, 2];
    //fill tempr array before swaps by oreder
    int count_arr_v = 0;
    for (int i = 0; i < shcol.Count; i++)
    {
        ShapeBase sb = (ShapeBase)shcol[i];
        if (sb is Vulnerability)
        {
            Vulnerability vuln = new Vulnerability();
            vuln = (Vulnerability)sb;
            if (vuln.gid_parent == pgid)
            {
                array_sec_temp[count_arr_v, 0] = vuln.gid;
                array_sec_temp[count_arr_v, 1] = vuln.risk;
                string str_ = Convert.ToString(array_sec_temp[count_arr_v, 1]);
                int ii = Convert.ToInt16(array_sec_temp[count_arr_v, 0]);
                string sgid = Convert.ToString(ii);
                str_tot = str_tot + " " + str_ + " " + sgid + "***";
                count_arr_v++;

            }
        }
```

```
        }
        str_tot = str_tot + "  after swap: ";
        bool swap_done = true;
        double v_tempr = 0.0;
        double v_tempr2 = 0.0;
        while (swap_done == true)
        {
            swap_done = false;
            for (int n = 1; n < count_arr; n++) // we start from second member of array
            {

                if (array_sec_temp[n - 1, 1] < array_sec_temp[n, 1]) // do swap
                {
                    v_tempr = array_sec_temp[n - 1, 1];
                    v_tempr2 = array_sec_temp[n - 1, 0];
                    array_sec_temp[n - 1, 1] = array_sec_temp[n, 1];
                    array_sec_temp[n - 1, 0] = array_sec_temp[n, 0];
                    array_sec_temp[n, 1] = v_tempr;
                    array_sec_temp[n, 0] = v_tempr2;
                    swap_done = true;
                }
            }
        } // end while

        ////

        for (int n = 0; n < count_arr_v; n++) //array
        {
            string str_ = Convert.ToString(array_sec_temp[n, 1]);
            int ii = Convert.ToInt16(array_sec_temp[n, 0]);
            array_sec_vuln[nn][n] = array_sec_temp[n, 0];
            string sgid = Convert.ToString(ii);

            ShapeBase sb = (ShapeBase)shcol[ii];
            Vulnerability vv = new Vulnerability();
            vv = (Vulnerability)sb;
            vv.sec_num = n + 1;
            string ssec = Convert.ToString(vv.sec_num);
            string nna = Convert.ToString(vv.Name);
            string nr = Convert.ToString(vv.Risk);
        }
    }


    #endregion //Servece functions & procedures

    #region Form Event
    /// <summary>
                    ///
                    /// </summary>
                    /// <param name="sender"></param>
                    /// <param name="e"></param>
                    private void Schema_Load(object sender, System.EventArgs e)
                    {
        try
        {
            // TODO: This line of code loads data into the 'db_vt_1DataSet.library_coun_4a_v' table. You can move,
or remove it, as needed.
            this.library_coun_4a_vTableAdapter.Fill(this.db_vt_1DataSet.library_coun_4a_v);
            // TODO: This line of code loads data into the 'db_vt_1DataSet.library_v_a' table. You can move, or
```

```
remove it, as needed.
            this.library_v_aTableAdapter.Fill(this.db_vt_1DataSet.library_v_a);
            // TODO: This line of code loads data into the 'db_vt_1DataSet.library_comp_vuln' table. You can move,
or remove it, as needed.
            this.library_comp_vulnTableAdapter.Fill(this.db_vt_1DataSet.library_comp_vuln);
            // TODO: This line of code loads data into the 'db_vt_1DataSet.library_comps' table. You can move, or
remove it, as needed.
            this.library_compsTableAdapter.Fill(this.db_vt_1DataSet.library_comps);
            // TODO: This line of code loads data into the 'db_vt_1DataSet.library_counts' table. You can move, or
remove it, as needed.
            this.library_countsTableAdapter.Fill(this.db_vt_1DataSet.library_counts);
            // TODO: This line of code loads data into the 'db_vt_1DataSet.library_attacks' table. You can move, or
remove it, as needed.
            this.library_attacksTableAdapter.Fill(this.db_vt_1DataSet.library_attacks);
            // TODO: This line of code loads data into the 'db_vt_1DataSet.library_vulns' table. You can move, or
remove it, as needed.
            this.library_vulnsTableAdapter.Fill(this.db_vt_1DataSet.library_vulns);
            // TODO: This line of code loads data into the 'db_vt_1DataSet.settings' table. You can move, or remove
it, as needed.
            this.settingsTableAdapter.Fill(this.db_vt_1DataSet.settings);
            //set page size
            TreePicture.Width = SchWidth;
            TreePicture.Height = SchHeight;

            add_toMenu_show();
            //set new shape collection
            shcol = new ShapeCollection();
            shcol.dist = dist;
            shcol.dist1 = dist1;
            shcol.dist2 = dist2;
            shcol.dist3 = dist3;
            shcol.dist4 = dist4;
            shcol.array_sec_comp = array_sec_comp;
            //shcol.array_sec_vuln = array_sec_vuln;
            concol = new ConnectionCollection();
            asset_now = new Asset();

           // g.Clear(TreePicture.BackColor);
            if (this.MdiParent.Name == "TreeBase")
            {

                TreeBase frm = (TreeBase)this.MdiParent;
                TreePicture.BackColor = frm.graph_backGr; // System.Drawing.Color.Aquamarine;
                this.a_calcul_type = frm.a_calcul_type;
               // this.menuItemSave.Enabled = true;
                this.menuSaveTemAsPr.Enabled = false;

                switch (frm.form_type)
                {
                   case 1:
                      //form_type=1  new project from template

                      this.menuItemSaveAll.Enabled = false;
                      this.menuSaveTemAsPr.Enabled = true;
                      this.menuItem_as_NewTempl.Enabled = false;
                      break;

                   case 2:
                      //form_type=2  new project without template
```

```
        this.menuItemSaveAll.Enabled = true;
        this.menuSaveTemAsPr.Enabled = false;
        this.menuItem_as_NewTempl.Enabled = false;
        break;
    case 3:
        //form_type=3 Open existing project for editing

        this.menuItemSaveAll.Enabled = true;
        this.menuSaveTemAsPr.Enabled = false;
        this.menuItem_as_NewTempl.Enabled = true;
        break;

    case 5:
        //form_type=5  new template

        this.menuItemSaveAll.Enabled = true;
        this.menuSaveTemAsPr.Enabled = false;
        this.menuItem_as_NewTempl.Enabled = false;
        string str_lev = Convert.ToString(frm.comboBox_Level.Text);
        switch (str_lev)
        {
            case "Asset":
                this.temp_level = 0;
                break;
            case "Component":
                this.temp_level = 1;
                break;
            case "Vulnerability":
                this.temp_level = 2;
                break;
            case "Attack":
                this.temp_level = 3;
                break;
            case "Countermeasure":
                this.temp_level = 4;
                break;
        }
        //this.temp_level = frm.comboBox_Level.Items.IndexOf(frm.comboBox_Level.ValueMember);
       //Convert.ToInt16(frm.comboBox_Level.Items.IndexOf);
         break;
    case 7:
        //form_type=7 Open existing template

        this.menuItemSaveAll.Enabled = true;
        this.menuSaveTemAsPr.Enabled = false; //???
        this.menuItem_as_NewTempl.Enabled = false;
        break;


}
Graphics g = CreateGraphics();
con_type = frm.connect_type;

int flag = frm.flagObject;
Asset asset = new Asset();
if ((frm.form_type != 2)&&  (frm.form_type != 5))
{
    asset = do_RootGraph(sender);
}
```

```
                asset_now = asset;
                string st7 = Convert.ToString(asset_now.setting_id);
                set_schema_colors ( sender,  e, asset_now.setting_id);
            }
        }
        catch (Exception ex)
        {

            MessageBox.Show("Schema_Load:\r\n" + ex.Message,
                    "Schema_Load failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
        }

    }
    public void add_toMenu_show()
    {
        TreeBase frm = (TreeBase)this.MdiParent;
        foreach (Floaty f in frm.dockExtender.Floaties)
        {
            ToolStripItem item = menuView.DropDownItems.Add(f.Text);
            item.MouseUp += new MouseEventHandler(frm.item_MouseUp);
            item.Tag = f;
        }
    }
    #endregion //Form Event

    #region  Menu Events (Context menu and Strip menu)

    #region Strip menu

    private void saveGraphicToFileToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frmS = new form_Pict();
        frmS.con_type = con_type;
        frmS.shcol3 = shcol;
        frmS.concol3 = concol;
        frmS.picW = TreePicture.Width;
        frmS.picH = TreePicture.Height;
        frmS.ShowDialog();
    }

    private void exitToolStripMenuItem_Click(object sender, EventArgs e)
    {
        bool da = false;
        DialogResult dr = DialogResult.No;
        bool is_S = isShema_Saved(sender, e);
        if (!is_S)
        {
            dr = MessageBox.Show("Do You want to save changes before exit?", "Save?",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);

        }
        if (dr == DialogResult.Yes)
        {
            da = true;
        }
        if (da)
        {
            menuItemSaveAll_Click(sender, e);
```

```
            //this.Close();
        }

        //else
        //{
        Close();
        //}

    }
    private void toolStripMenuItemSettings_Click(object sender, EventArgs e)
    {
        paGrProperty.Visible = true;
        DataTable dt = db_vt_1DataSet.Tables["settings"];

        comBox_Settings.DataSource = dt;
        comBox_Settings.DisplayMember = "s_name";
        string sname =
Convert.ToString(this.settingsTableAdapter.ScalarQueryFindName_byId(asset_now.setting_id));
        laSettings_Now.Text = sname;
    }

    #endregion //  strip menu

    #region Context menu
    /// <summary>
    /// add any leaf to the tree
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void menuAddAsset_Click(object sender, System.EventArgs e)
    {
        if (this.temp_level>0) // do templates for components, vuln, attacks, countms
        {
            MessageBox.Show("Sorry, this  functionality didn't developed yet = " );
        }
        else // do project and template for project
        {
            int selectShape = GetTypeObjects();
            int gid_par = GetGIdObject();
            //string str2 = Convert.ToString(gid_par);
            CreateNewShape(selectShape, gid_par);
        }
    }

    private void menuItemDel_Click(object sender, EventArgs e)
    {
        panel_wait.Visible = true;

        progressBar1.Minimum = 0;
        progressBar1.Maximum = 10;
        progressBar1.Step = 1;
        progressBar1.PerformStep();
        TreeBase frmEx = (TreeBase)MdiParent;
        bool deleted_ = delete_item_db(sender, e);
        int g_id = shcol.GetIndexSelectObject();
        if (g_id > 0)
        {
            frmEx.assetTableAdapter.FillBy_Asset_Id(frmEx.db_vt_1DataSet.asset, asset_now.a_id);

            // TODO: This line of code loads data into the 'db_vt_1DataSet.component' table. You can move, or
```

```
remove it, as needed.
            frmEx.componentTableAdapter.Fill(frmEx.db_vt_1DataSet.component);
            progressBar1.PerformStep();
            // TODO: This line of code loads data into the 'db_vt_1DataSet.vulnerability' table. You can move, or
remove it, as needed.
            frmEx.vulnerabilityTableAdapter.Fill(frmEx.db_vt_1DataSet.vulnerability);
            progressBar1.PerformStep();
            // TODO: This line of code loads data into the 'db_vt_1DataSet.attack' table. You can move, or remove
it, as needed.
            frmEx.attackTableAdapter.Fill(frmEx.db_vt_1DataSet.attack);
            // TODO: This line of code loads data into the 'db_vt_1DataSet.countm' table. You can move, or remove
it, as needed.
            frmEx.countermeasureTableAdapter.Fill(frmEx.db_vt_1DataSet.countermeasure);
            progressBar1.PerformStep();
            frmEx.assetDataGridView_DoubleClick(sender, e);
            progressBar1.PerformStep();
        }
        else
        {
            if (deleted_)
            {

                MessageBox.Show(" Project deleted ");
                this.Close();
                /// <summary>
                /// form_type defines what data download to the form
                /// form_type=1 Open new project from template
                /// form_type=2 Open new project without template
                /// form_type=3 Open existing project for editing
                /// form_type=5 Open new template create
                /// form_type=7 Open existing template for Edit
                /// </summary>
                if ((frmEx.form_type == 1) || (frmEx.form_type == 5) || (frmEx.form_type == 7))
                {
                    frmEx.assetTableAdapter.FillByAt_Id_Tmp_Y(frmEx.db_vt_1DataSet.asset, asset_now.at_id);
                }
                if ((frmEx.form_type == 2) || (frmEx.form_type == 3))
                {

                    frmEx.assetTableAdapter.FillByAt_Id_Tmp_N(frmEx.db_vt_1DataSet.asset, asset_now.at_id);
                }
            }

        }
        panel_wait.Visible = false;

    }

    private void menuItemSaveAll_Click(object sender, EventArgs e)
    {
        // do save all schema
        TreeBase frmEx = (TreeBase)MdiParent;
        switch (frmEx.form_type)
        {
            case 1:
                //form_type=1  new project from template
                my_templ = false;
                this.menuItemSaveAll.Enabled = true;
                this.menuSaveTemAsPr.Enabled = false;
                this.menuItem_as_NewTempl.Enabled = false;
```

```
            break;

        case 2:
            //form_type=2  new project without template
            my_templ = false;
            this.menuItemSaveAll.Enabled = true;
            this.menuSaveTemAsPr.Enabled = false;
            this.menuItem_as_NewTempl.Enabled = false;
            break;
        case 3:
            //form_type=3 Open existing project for editing
            my_templ = false;
            this.menuItemSaveAll.Enabled = true;
            this.menuSaveTemAsPr.Enabled = false;
            this.menuItem_as_NewTempl.Enabled = true;
            break;

        case 5:
            //form_type=5  new template
            my_templ = true;
            this.menuItemSaveAll.Enabled = true;
            this.menuSaveTemAsPr.Enabled = false;
            this.menuItem_as_NewTempl.Enabled = false;
            break;
        case 7:
            //form_type=7 Open existing template
            my_templ = true;
            this.menuItemSaveAll.Enabled = true;
            this.menuSaveTemAsPr.Enabled = false; //???
            this.menuItem_as_NewTempl.Enabled = false;
            break;


    }
    bool save_as_new = false;
    saveAll_now(sender, e, my_templ, save_as_new);
}

private void saveAllToolStripMenuItem_Click(object sender, EventArgs e)
{

}

private void menuItem12_Click(object sender, System.EventArgs e)
{
    ShowPropertyShape(1);

}


private void menuItemListCountm_Click(object sender, EventArgs e)
{
    frmList = new forms_Data.form_ListCount();
    frmList.asset_id_now = asset_now.a_id;
    frmList.asset_name_now = asset_now.Name;
    frmList.ShowDialog();
}

private void menuItemListAttacks_Click(object sender, EventArgs e)
{
```

```
    frmAttacks = new forms_Data.form_ListAttacks();
    frmAttacks.asset_id_now = asset_now.a_id;
    frmAttacks.asset_name_now = asset_now.Name;
    frmAttacks.ShowDialog();
}

private void menuItemSaveTreeIm_Click(object sender, EventArgs e)
{
    saveGraphicToFileToolStripMenuItem_Click(sender, e);

}
#endregion //  context menu

#region Library_Data_to_fill

private void menuItemAdd_frLib_Click(object sender, EventArgs e)
{
    int selectShape = GetTypeObjects();
    panel_Lib.Visible = true;
    rBn_with_dep.Checked = true;
    rBn_without_dep.Checked = false;
    close_allDataGridViews(sender, e);

    switch (selectShape)
    {
        case 0:
            library_compsDataGridView.Visible = true;
            library_compsDataGridView.Size = new Size(400, 400);
            library_compsDataGridView.Location = new Point(3, 60);
            lbTableName.Text = "Components";
            break;
        case 1:
            library_vulnsDataGridView.Visible = true;
            library_vulnsDataGridView.Size = new Size(400, 400);
            library_vulnsDataGridView.Location = new Point(3, 60);
            lbTableName.Text = "Vulnerabilities";
            break;
        case 2:
            library_attacksDataGridView.Visible = true;
            library_attacksDataGridView.Size = new Size(400, 400);
            library_attacksDataGridView.Location = new Point(3, 60);
            lbTableName.Text = "Attacks";
            break;
        case 3:
            library_countsDataGridView.Visible = true;
            library_countsDataGridView.Size = new Size(400, 400);
            library_countsDataGridView.Location = new Point(3, 60);
            lbTableName.Text = "Countermeasures";
            break;
    }

    //
}
private void close_allDataGridViews(object sender, EventArgs e)
{
    library_compsDataGridView.Visible = false;
    library_attacksDataGridView.Visible = false;
    library_countsDataGridView.Visible = false;
    library_comp_vulnDataGridView.Visible = false;
    library_v_aDataGridView.Visible = false;
```

```
            library_coun_4a_vDataGridView.Visible = false;
        }
        private void library_compsDataGridView_DoubleClick(object sender, EventArgs e)
        {
            int selectShape = GetTypeObjects();
            if (selectShape == 0)
            {
                int gid_par = GetGIdObject();
                string str2 = Convert.ToString(gid_par);

                TreeBase frmEx = (TreeBase)MdiParent;
                Asset asset = (Asset)shcol[gid_par];
                SolidBrush comp_brush = new SolidBrush(frmEx.comp_color);
                Component component = new Component(point[0], shcol, this.TreePicture, indexSelect, comp_brush);

                component.criticality = 0;
                component.capitalcost = 0;
                component.risk = 0;
                component.ecl = 0;
                component.Saved = true;
                component.id_db_parent = asset_now.a_id;
                //Convert Object to Double;
                int i = 0;
                i = this.library_compsDataGridView.CurrentCell.RowIndex;
                string str = Convert.ToString(this.library_compsDataGridView.CurrentCell.RowIndex);

                component.a_calcul_type = a_calcul_type;
                // somehow function convert object to double doesn,t work. do first to string
//Convert.ToDouble(frmEx.library_compsDataGridView[4, i].Value);

                component.max_risk = 0;
                component.min_risk = 0;
                int c_id_lib = Convert.ToInt16(library_compsDataGridView[0, i].Value);

                component.Name = Convert.ToString(library_compsDataGridView[1, i].Value);
                component.c_id = 0;
                component.desc = Convert.ToString(library_compsDataGridView[2, i].Value);
                component.sec_num = 0;
                component.a_calcul_type = a_calcul_type;

                component.weight = Convert.ToInt16(library_compsDataGridView[3, i].Value);
                component.gid_parent = asset.gid;
                string str4 = Convert.ToString(library_compsDataGridView[4, i].Value);
                string str5 = Convert.ToString(library_compsDataGridView[5, i].Value);
                string str6 = Convert.ToString(library_compsDataGridView[6, i].Value);
                string str7 = Convert.ToString(library_compsDataGridView[7, i].Value);
                // somehow function convert object to double doesn,t work. do first to string
//Convert.ToDouble(frmEx.library_compsDataGridView[4, i].Value);
                component.criticality = Convert.ToDouble(str4);
                component.capitalcost = Convert.ToDouble(str5);
                component.risk = Convert.ToDouble(str6);
                component.ecl = Convert.ToDouble(str7);


                GetPositionX(1, component.RectObject.Width);


                // Adds elements to the collection.
```

```
        Connection connect = new Connection(asset, component, concol);
        asset.AddChild(component, connect);
        g_id = g_id + 1;
        component.gid = g_id;
        component.id_db_parent = asset.a_id;       //
        frmEx.comp_id_now = component.c_id;
        //frmEx.vulnerabilityTableAdapter.FillByCID(frmEx.db_vt_1DataSet.vulnerability, frmEx.comp_id_now);
        //frmEx.vulnerabilityTableAdapter.ClearBeforeFill;
        frmEx.comp_row_num = i;  // frmEx.library_compsDataGridView.CurrentRow.Index;
        mComponent_now = component.weight;
        comp_now = component;

        if (rBn_with_dep.Checked)
        {
           library_comps_Add_vulns(sender, e, g_id, c_id_lib);

        }

        library_compsDataGridView.Visible = false;
        panel_Lib.Visible = false;
     }
     else
     {
        MessageBox.Show("Please select Asset first");
     }
}// end of prog


private void library_vulnsDataGridView_DoubleClick(object sender, EventArgs e)
{
   //Vulnerability vuln_lib = new Vulnerability();
   //menuAddAsset_Click(sender, e);
   int selectShape = GetTypeObjects();
   if (selectShape == 1)
   {
      int gid_par = GetGIdObject();
      string str2 = Convert.ToString(gid_par);
      fill_library_vulns(sender, e, gid_par);


      MouseButtons mb = new MouseButtons();
      System.Windows.Forms.MouseEventArgs ee = new MouseEventArgs(mb, 1, 100, 100, 1);
      //TreePicture_MouseClick(sender, ee);
   }
   else
   {
      MessageBox.Show("Please select Component first");
   }

}
private void fill_library_vulns(object sender, EventArgs e, int gid_par)
{
   TreeBase frmEx = (TreeBase)MdiParent;
   int num_co = shcol.Count;
   try
   {
      Component component = (Component)shcol[gid_par];
      SolidBrush vuln_brush = new SolidBrush(frmEx.vuln_color);
      Vulnerability vuln = new Vulnerability(point[0], shcol, this.TreePicture, gid_par, vuln_brush);
```

```
            vuln.risk = 0;
            vuln.weight = 0;
            vuln.id_db_parent = component.c_id;
            vuln.sec_num = 0;
            vuln.Saved = false;
            //Convert Object to Double;
            int i = 0;
            i = this.library_vulnsDataGridView.CurrentCell.RowIndex;
            string str = Convert.ToString(this.library_vulnsDataGridView.CurrentCell.RowIndex);
            int vul_id_lib = 0;
            if (library_vulnsDataGridView[0, i].Value != null)
            {

               vuln.v_id = 0;
               vul_id_lib = Convert.ToInt16(library_vulnsDataGridView[0, i].Value);
               vuln.Name = Convert.ToString(library_vulnsDataGridView[1, i].Value);
               vuln.Desc = Convert.ToString(library_vulnsDataGridView[2, i].Value);
               vuln.weight = Convert.ToInt16(library_vulnsDataGridView[3, i].Value);
               vuln.vt_id = Convert.ToInt16(library_vulnsDataGridView[4, i].Value);
               vuln.example = Convert.ToString(library_vulnsDataGridView[5, i].Value);
               vuln.Risk = Convert.ToInt16(library_vulnsDataGridView[6, i].Value);
               vuln.platform = Convert.ToInt16(library_vulnsDataGridView[8, i].Value);
            }
            vuln.a_calcul_type = a_calcul_type;
            // somehow function convert object to double doesn,t work. do first to string
//Convert.ToDouble(frmEx.componentDataGridView[4, i].Value);

            vuln.max_risk = 0;
            vuln.min_risk = 0;
            GetPositionX(2, vuln.RectObject.Width);

            Connection connect = new Connection(component, vuln, concol);
            component.AddChild(vuln, connect);
            g_id = g_id + 1;
            vuln.gid_parent = component.gid;
            vuln.gid = g_id;
            vuln.id_db_parent = component.c_id;
            vuln_now = vuln;
            if (rBn_with_dep.Checked)
            {
               library_vulns_Add_attacks(sender, e, g_id, vul_id_lib);
            }
            library_vulnsDataGridView.Visible = false;
            panel_Lib.Visible = false;
         }
         catch (Exception ex)
         {
            MessageBox.Show(" fill_library_vulns:\r\n" + ex.Message,
                     " Failed", MessageBoxButtons.OK,
                     MessageBoxIcon.Error);
         }
      }
      private void library_attacksDataGridView_DoubleClick(object sender, EventArgs e)
      {
         int selectShape = GetTypeObjects();
         if (selectShape == 2)
         {
            int gid_par = GetGIdObject();
            string str2 = Convert.ToString(gid_par);
```

```
            fill_library_attack(sender, e, gid_par);
        }
        else
        {
            MessageBox.Show("Please select Vulnerability first");
        }

    }
    private void fill_library_attack(object sender, EventArgs e, int gid_par)
    {
        TreeBase frmEx = (TreeBase)MdiParent;
        try
        {
            Vulnerability vuln = (Vulnerability)shcol[gid_par];
            SolidBrush attack_brush = new SolidBrush(frmEx.attack_color);

            Attack attack = new Attack(point[0], shcol, this.TreePicture, gid_par, attack_brush);
            attack.risk = 0;
            attack.weight = 0;
            attack.id_db_parent = vuln.v_id;
            attack.Saved = false;
            int i = 0;
            i = this.library_attacksDataGridView.CurrentCell.RowIndex;

            if (library_attacksDataGridView[0, i].Value != null)
            {
                string str = Convert.ToString(this.library_attacksDataGridView.CurrentCell.RowIndex);

                attack.at_id = 0;
                int att_id_lib = Convert.ToInt16(library_attacksDataGridView[0, i].Value);

                attack.Name = Convert.ToString(library_attacksDataGridView[1, i].Value);
                attack.Desc = Convert.ToString(library_attacksDataGridView[2, i].Value);
                attack.at_type_id = Convert.ToInt16(library_attacksDataGridView[3, i].Value);
                attack.weight = Convert.ToInt16(library_attacksDataGridView[4, i].Value);
                attack.a_calcul_type = a_calcul_type;
                attack.vul_id = vuln.v_id;

                attack.source = Convert.ToString(library_attacksDataGridView[5, i].Value);
                attack.easienes = Convert.ToInt16(library_attacksDataGridView[6, i].Value);
                attack.severity = Convert.ToInt16(library_attacksDataGridView[7, i].Value);
                attack.example = Convert.ToString(library_attacksDataGridView[8, i].Value);
                attack.more_names = Convert.ToString(library_attacksDataGridView[9, i].Value);
                attack.website = Convert.ToString(library_attacksDataGridView[10, i].Value);
                attack.test = Convert.ToString(library_attacksDataGridView[11, i].Value);
                GetPositionX(3, attack.RectObject.Width);
                // Adds elements to the collection.
                //comp_.Add(vuln);
                Connection connect = new Connection(vuln, attack, concol);
                vuln.AddChild(attack, connect);
                g_id = g_id + 1;
                attack.gid = g_id;
                attack.gid_parent = vuln.gid;
                attack.id_db_parent = vuln.v_id;
                attack_now = attack;
                if (rBn_with_dep.Checked)
                {
                    library_attack_Add_countms(sender, e, g_id, att_id_lib);
                }
                library_attacksDataGridView.Visible = false;
```

```
            panel_Lib.Visible = false;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(" fill_library_attack:\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
    }
}
private void library_countsDataGridView_DoubleClick(object sender, EventArgs e)
{
    int gid_par = GetGIdObject();
    string str2 = Convert.ToString(gid_par);
    int selectShape = GetTypeObjects();
    if (selectShape == 3)
    {

        fill_library_countms(sender, e, gid_par);

        library_countsDataGridView.Visible = false;
        panel_Lib.Visible = false;

    }
    else
    {
        MessageBox.Show("Please select Attack first");
    }
}

private void fill_library_countms(object sender, EventArgs e, int gid_par)
{
    TreeBase frmEx = (TreeBase)MdiParent;
    Attack attack = (Attack)shcol[gid_par];
    SolidBrush countm_brush = new SolidBrush(frmEx.countm_color);

    Countermeasure countm = new Countermeasure(point[0], shcol, this.TreePicture, gid_par, countm_brush);
    //countm.risk = 0;
    countm.weight = 0;
    countm.id_db_parent = attack_now.at_id;
    countm.Saved = false;
    int nc = library_countsDataGridView.RowCount;
    nc = nc - 1;
    if (nc > 0)
    {
        int i = 0;
        i = this.library_countsDataGridView.CurrentCell.RowIndex;

        if (library_countsDataGridView[0, i].Value != null)
        {
            countm.countm_id = 0;
            countm.Name = Convert.ToString(library_countsDataGridView[1, i].Value);
            countm.Desc = Convert.ToString(library_countsDataGridView[2, i].Value);
            countm.example = Convert.ToString(library_countsDataGridView[3, i].Value);
            countm.weight = Convert.ToInt16(library_countsDataGridView[4, i].Value);
            countm.complexity = Convert.ToInt16(library_countsDataGridView[5, i].Value);

            countm.source = Convert.ToString(library_countsDataGridView[8, i].Value);
            countm.pl_id = Convert.ToInt16(library_countsDataGridView[9, i].Value);
            countm.website = Convert.ToString(library_countsDataGridView[10, i].Value);
```

```
            countm.implemented = false;

            GetPositionX(4, countm.RectObject.Width);
            // Adds elements to the collection.
            g_id = g_id + 1;
            countm.gid = g_id;
            countm.gid_parent = attack.gid;
            countm.v_id = attack.vul_id;
            countm.at_id = attack.gid;
            countm.id_db_parent = attack.at_id;
            Connection connect = new Connection(attack, countm, concol);
            attack.AddChild(countm, connect);
        }
    }
}
/// <summary>
/// ////////////////////////////////
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void library_comps_Add_vulns(object sender, EventArgs e, int gid_par, int c_id_lib)
{
    library_comp_vulnTableAdapter.FillBy_Comp_Id(db_vt_1DataSet.library_comp_vuln, c_id_lib);
    int nn = library_comp_vulnDataGridView.RowCount;
    int vul_id_lib = 0;
    nn = nn - 1;
    if (nn > 0)
    {
        for (int i = 0; i < nn; i++)
        {
            // find Id for vuln
            vul_id_lib = Convert.ToInt16(library_comp_vulnDataGridView[1, i].Value);
            library_vulnsTableAdapter.FillBy_Vul_Id(db_vt_1DataSet.library_vulns, vul_id_lib);
            //library_vulnsDataGridView_DoubleClick(sender, e);
            fill_library_vulns(sender, e, gid_par);
        }
    }

}

private void library_vulns_Add_attacks(object sender, EventArgs e, int gid_par, int vul_id_lib)
{
    library_v_aTableAdapter.FillBy_Vul_Id(db_vt_1DataSet.library_v_a, vul_id_lib);
    int nn = library_v_aDataGridView.RowCount;
    int att_id_lib = 0;
    nn = nn - 1;
    if (nn > 0)
    {
        for (int i = 0; i < nn; i++)
        {
            // find Id for vuln
            att_id_lib = Convert.ToInt16(library_v_aDataGridView[1, i].Value);
            library_attacksTableAdapter.FillBy_At_Id(db_vt_1DataSet.library_attacks, att_id_lib);

            //library_vulnsDataGridView_DoubleClick(sender, e);
            fill_library_attack(sender, e, gid_par);
        }
```

```
        }

    }

    private void library_attack_Add_countms(object sender, EventArgs e, int gid_par, int att_id_lib)
    {
        library_coun_4a_vTableAdapter.FillBy_At_Id(db_vt_1DataSet.library_coun_4a_v, att_id_lib);
        int nn = library_coun_4a_vDataGridView.RowCount;
        int count_id_lib = 0;
        nn = nn - 1;
        if (nn > 0)
        {
            for (int i = 0; i < nn; i++)
            {
                // find Id for vuln
                count_id_lib = Convert.ToInt16(library_coun_4a_vDataGridView[0, i].Value);
                library_countsTableAdapter.FillBy_Count_Id(db_vt_1DataSet.library_counts, count_id_lib);
                fill_library_countms(sender, e, gid_par);
            }
        }

    }

    private void bnClosePan_Click(object sender, EventArgs e)
    {
        panel_Lib.Visible = false;
    }
    #endregion //Library_Data_to_fill

    #region Do History

    private void toolStripMenuItemToHistory_Click(object sender, EventArgs e)
    {
        save_ToHistory(sender, e);
    }

    private void save_ToHistory(object sender, EventArgs e)
    {
        asset_now.random = randomString(10);
        TreeBase frmEx = (TreeBase)MdiParent;
        try
        {
            frmEx.asset_row_num = frmEx.assetDataGridView.NewRowIndex; //assetDataGridView1
            asset_now.random = randomString(10);
            if (asset_now.at_id == 0)
            {
                asset_now.at_id = frmEx.at_id_now;
            }
            frmEx.assetTableAdapter.InsertQueryGr(asset_now.Name, asset_now.Desc,
Convert.ToDecimal(asset_now.criticality), Convert.ToDecimal(asset_now.capitalcost),
Convert.ToDecimal(asset_now.risk), Convert.ToDecimal(asset_now.ecl), asset_now.at_id, false,
asset_now.random, 1, 1, asset_now.weight, Convert.ToDecimal(asset_now.max_risk),
Convert.ToDecimal(asset_now.min_risk), true, asset_now.a_id, System.DateTime.Today);

            asset_now.h_asset_id =
Convert.ToInt16(frmEx.assetTableAdapter.ScalarQueryFindId_ByRandom(asset_now.random));
            MessageBox.Show("Project was saved to History:\r\n" + Convert.ToString(asset_now.h_asset_id));
        }
        catch (Exception ex)
        {
```

```
            MessageBox.Show(" save_ToHistory:\r\n" + ex.Message,
                        " Failed", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
        }
    }
    #endregion // to History


    #endregion //Context menu Events

    #region Mouse Event
    /// <summary>
                    ///
                    /// </summary>
                    /// <param name="sender"></param>
                    /// <param name="e"></param>
                    private void TreePicture_MouseDown(object sender,
System.Windows.Forms.MouseEventArgs e)
                    {
                            point[0].X = e.X;
                            point[0].Y = e.Y;

                            indexSelect = -100;
                            intMode = GetObjectFromSchema(point[0], out indexSelect);

                            switch (intMode)
                            {
                                    case 1:
                                    {
                                            ShapeBase sb = (ShapeBase)shcol[indexSelect];
                                            if(sb.GetStorageArea(point[0]))
                                            {
                                                    bool fl = sb.Surrogate;
            MakeVisibleShape(sb,fl);
                                                    sb.Surrogate = !fl;
                                                    intMode     = - 100;
                                                    //break;
                                            }

                                            int iS = shcol.GetIndexSelectObject();
                                            if(iS > -1) shcol.SelectObject(iS);
                                            shcol.SelectObject(indexSelect);

                                            RecShape    = sb.RectObject;
                                            point[4].X  = e.X - RecShape.X;
                                            point[4].Y  = e.Y - RecShape.Y;

                                            ShowPropertyShape(0);

                                            TreePicture.Invalidate();

                                            break;
                                    }           // end case 1

                                    case -100:
                                    {
                                            int iS = shcol.GetIndexSelectObject();
                                            if(iS > -1) shcol.SelectObject(iS);
                                            TreePicture.Invalidate();
                                            break;
```

```
                        }               // end case -100
                    }               // end switch
                }
                /// <summary>
                ///
                /// </summary>
                /// <param name="sender"></param>
                /// <param name="e"></param>
                private void TreePicture_MouseMove(object sender,
System.Windows.Forms.MouseEventArgs e)
                {
                    if (e.Button == MouseButtons.Left)
                    {
                        switch (intMode)
                        {
                            case 1:
                            {
                                RecShape.X = e.X - point[4].X;
                                RecShape.Y = e.Y - point[4].Y;
                                break;
                            }
                        }

                        TreePicture.Invalidate();
                    }


                }
                /// <summary>
                ///
                /// </summary>
                /// <param name="sender"></param>
                /// <param name="e"></param>
                private void TreePicture_MouseUp(object sender, System.Windows.Forms.MouseEventArgs
e)
                {

                    if(e.Button == MouseButtons.Left)
                    {
                        switch (intMode)
                        {
                            case 1:
                            {
                                ShapeBase sb = (ShapeBase)shcol[indexSelect];
                                sb.RectObject = RecShape;
                                SetFlagIndex();
                                TreePicture.Invalidate();
                                break;
                            }//end case 1
                        }//end switch intMode
                    }
    }

    #endregion //Mouse Event

    #region TreePicture's Event

    private void TreePicture_Paint(object sender, System.Windows.Forms.PaintEventArgs e)
                {
```

```
                        //paint selected rectangle
                        switch (intMode)
                        {
                                case 1:
                                {
                                        Pen p = new Pen(Color.Black, 1);
                                        e.Graphics.DrawRectangle(p,RecShape);
                                        break;
                                }       //end case 1:
                        }               //end switch
                        //paint all shape
                        for (int i = 0; i < shcol.Count; i ++)
                        {
                                shcol.PaintObjects(e.Graphics, i);
                        }
    TreeBase frmEx = (TreeBase)MdiParent;

                        //paint all connector
                        for (int i = 0; i < concol.Count; i ++)
                        {
        concol.PaintConnector(e.Graphics, i, frmEx.connect_type);
                        }
                }
/// <summary>
/// Set colors for schema by setting id
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
/// <param name="setting_id"></param>
private void set_schema_colors(object sender, System.EventArgs e, int setting_id)
{
    TreeBase frmEx = (TreeBase)MdiParent;
    try
    {
        string st7 = Convert.ToString(setting_id);
        frmEx.settingsTableAdapter.FillById(frmEx.db_vt_1DataSet.settings, setting_id);

        DataRowCollection dr_settings;

        dr_settings = frmEx.db_vt_1DataSet.Tables["settings"].Rows;

        DataRow dr = dr_settings[0];
        Color myColor = Color.PaleVioletRed;

        // Create the ColorConverter.
        System.ComponentModel.TypeConverter converter =
        System.ComponentModel.TypeDescriptor.GetConverter(myColor);

        string str3=Convert.ToString(dr[3]);
        frmEx.asset_color = color_from_string(str3);
        frmEx.comp_color = color_from_string(Convert.ToString(dr[4]));
        frmEx.vuln_color = color_from_string(Convert.ToString(dr[5]));
        frmEx.attack_color = color_from_string(Convert.ToString(dr[6]));
        frmEx.countm_color = color_from_string(Convert.ToString(dr[7]));
        frmEx.graph_backGr = color_from_string(Convert.ToString(dr[8]));

        frmEx.connect_type = Convert.ToInt16(dr[2]);
    }
    catch (Exception ex)
    {
```

```
    MessageBox.Show("set_schema_colors:\r\n" + ex.Message,
            "set_schema_colors failed", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
 }
}
/// <summary>
/// Convertion program string convert to color
/// </summary>
/// <param name="our_string">string</param>
/// <returns>color</returns>
public Color color_from_string(string our_string)
{
   Color my_color= new Color();

   int nn = our_string.Length;

   int red;
   int green;
   int blue;
   if (nn==13)
   {
      string s_red = our_string.Substring(0, 3); //str3[0] + str3[1] + str3[2];
      string s_green = our_string.Substring(5, 3);
      string s_blue = our_string.Substring(10, 3);

   red = Convert.ToInt32(s_red);
   green = Convert.ToInt32(s_green);
   blue = Convert.ToInt32(s_blue);

   my_color = Color.FromArgb(red, green, blue);
   }
   return my_color;
}

private void TreePicture_DoubleClick(object sender, EventArgs e)
{
   ShowPropertyShape(1);
}

#endregion //TreePicture's Event

#region Do Graph
/// <summary>
/// Program for draw top level object - Asset for graph
/// </summary>
/// <param name="sender"></param>
/// <returns></returns>
private Asset do_RootGraph(object sender)
{
   Point position = new Point();
   have_asset = true;
   position.X = (int)(TreePicture.Width / 2) - 60;
   position.Y = 60;

      TreeBase frmEx = (TreeBase)MdiParent;
      SolidBrush asset_brush = new SolidBrush(frmEx.asset_color);
      Asset asset = new Asset(position, shcol, this.TreePicture, asset_brush);
   try
   {
```

```
        DataRow dr = frmEx.dr_asset[frmEx.asset_row_num];
        string aName = Convert.ToString(dr[1]);

        asset.Name = aName;
        asset.a_id = Convert.ToInt16(dr[0]);
        asset.Desc = Convert.ToString(dr[2]);
        asset.criticality = Convert.ToDouble(dr[3]);
        asset.capitalcost = Convert.ToDouble(dr[4]);
        asset.risk = Convert.ToDouble(dr[5]);
        asset.ecl = Convert.ToDouble(dr[6]);
        asset.at_id = Convert.ToInt16(dr[7]);
        asset.weight = Convert.ToInt16(dr[12]);
        asset.random = Convert.ToString(dr[9]);
        asset.Saved = true;

        asset.a_calcul_type = a_calcul_type;
        asset.setting_id = frmEx.assetTableAdapter.ScalarQueryFind_Settings_byId(asset.a_id).Value; //(dr[9]);
        Graphics g = CreateGraphics();
        asset.Paint(g, asset.GetStringProperty());

        asset.risk = do_Lev2Graph(sender, asset);
        asset.ecl = ecl_sum;
        asset.gid = 0;
        asset_now = asset;

    }
    catch (Exception ex)
    {
        MessageBox.Show(" do_RootGraph:\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
    }
    return asset;

}
/// <summary>
/// Program to draw  second level of tree all components here
/// </summary>
/// <param name="sender"></param>
/// <param name="asset"></param>
/// <returns>risk sum</returns>
private double do_Lev2Graph(object sender, Asset asset)
{
    //draw all components here, return comp risk sum
    TreeBase frmEx = (TreeBase)MdiParent;
    int nn = frmEx.componentDataGridView.RowCount;
    double risk_comp_sum = 0;
    double tot_max_risk = 0;
    double tot_min_risk = 0;
    double risk_comp = 0;
    try
    {

        ecl_sum = 0;
        nn = nn - 1;
        if ( nn > 0)
        {
            for (int i = 0; i < nn; i++)
            {
                SolidBrush comp_brush = new SolidBrush(frmEx.comp_color);
```

```
            Component component = new Component(point[0], shcol, this.TreePicture, indexSelect,
comp_brush);

            component.criticality = 0;
            component.capitalcost =0;
            component.risk = 0;
            component.ecl = 0;
            component.Saved = true;
            component.id_db_parent = asset_now.a_id;
            component.Name = Convert.ToString(frmEx.componentDataGridView[1, i].Value);
            component.c_id = Convert.ToInt16(frmEx.componentDataGridView[0, i].Value);
            component.desc = Convert.ToString(frmEx.componentDataGridView[2, i].Value);
            component.sec_num=0;
            component.a_calcul_type = a_calcul_type;
            component.weight = Convert.ToInt16(frmEx.componentDataGridView[10, i].Value);


            string st = Convert.ToString(frmEx.componentDataGridView[0, i].Value);
            component.gid_parent = asset.gid;
            if (frmEx.componentDataGridView[4, i].Value != null)
            {
               string str4 = Convert.ToString(frmEx.componentDataGridView[4, i].Value);
               string str5 = Convert.ToString(frmEx.componentDataGridView[5, i].Value);
               string str6 = Convert.ToString(frmEx.componentDataGridView[6, i].Value);
               string str7 = Convert.ToString(frmEx.componentDataGridView[7, i].Value);

               component.criticality = Convert.ToDouble(str4);
               component.capitalcost = Convert.ToDouble(str5);
               component.risk = Convert.ToDouble(str6);
               component.ecl = Convert.ToDouble(str7);
            }
            GetPositionX(1,component.RectObject.Width);
            // Adds elements to the collection.

            Connection connect = new Connection(asset, component, concol);
            asset.AddChild(component, connect);
            g_id = g_id + 1;
            component.gid =g_id ;
            component.id_db_parent = asset.a_id;
            // do security number here, for components
            for (int j = 0; j < tot_critic_num; j++)
            {
               if (component.gid ==Convert.ToInt16( array_sec_comp[j, 0]))
               {
                  component.sec_num = j+1;
               }
            }

            if ((component.sec_num<= tot_critic_num ) & ( component.sec_num >0))
            {
               component.add_ = "#" + component.sec_num;
            }
            frmEx.comp_id_now = component.c_id;

            frmEx.comp_row_num = i;  //
            mComponent_now = component.weight;
            risk_comp = do_Lev3Graph(sender, component, i);
            component.risk = risk_comp;
            risk_comp_sum = risk_comp_sum + component.risk;
            tot_max_risk = tot_max_risk + component.max_risk;
```

```
            tot_min_risk = tot_min_risk + component.min_risk;
          }
        }

        asset.max_risk = tot_max_risk;
        asset.min_risk = tot_min_risk;
      }
      catch (Exception ex)
      {
        MessageBox.Show(" do_Lev2Graph:\r\n" + ex.Message,
                  " Failed", MessageBoxButtons.OK,
                  MessageBoxIcon.Error);
      }
      return risk_comp_sum; //risk_comp_biggest;
    }
    /// <summary>
    /// Program to draw  3-d level -- all vulnerabilities for particular component
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="component"></param>
    /// <param name="co"></param>
     /// <returns> sumary of all risk for that component</returns>
     private double do_Lev3Graph(object sender,Component component,  int co)
     {
        //draw all vulnerabilities here, return sumary of all risk for component
        double risk_vuln_sum = 0;
        double tot_max_risk = 0;
        double tot_min_risk = 0;
        try
        {

          TreeBase frmEx = (TreeBase)MdiParent;
          // our current row in component table
          frmEx.comp_row_num = co;
          frmEx.comp_id_now = component.c_id;
          frmEx.fill_vuln_table(sender);
          int nn = frmEx.vulnerabilityDataGridView.RowCount;
          nn = nn - 1;

          int last_one = nn - 1;
          double risk_vuln = 0;
          if (nn > 0)
          {
            for (int i = 0; i < nn; i++)
            {
              SolidBrush vuln_brush = new SolidBrush(frmEx.vuln_color);
              Vulnerability vuln = new Vulnerability(point[0], shcol, this.TreePicture, indexSelect, vuln_brush);

              vuln.risk = 0;
              vuln.weight = 0;
              vuln.id_db_parent = comp_now.c_id;
              vuln.sec_num = 0;
              vuln.Saved = true;
              if (frmEx.vulnerabilityDataGridView[0, i].Value != null)
              {
                vuln.v_id = Convert.ToInt16(frmEx.vulnerabilityDataGridView[0, i].Value);
                vuln.Name = Convert.ToString(frmEx.vulnerabilityDataGridView[1, i].Value);
                vuln.Desc = Convert.ToString(frmEx.vulnerabilityDataGridView[2, i].Value);
                vuln.weight = Convert.ToInt16(frmEx.vulnerabilityDataGridView[3, i].Value);
```

```
                vuln.vt_id = Convert.ToInt16(frmEx.vulnerabilityDataGridView[4, i].Value);
                vuln.example = Convert.ToString(frmEx.vulnerabilityDataGridView[5, i].Value);
                vuln.Risk = Convert.ToInt16(frmEx.vulnerabilityDataGridView[6, i].Value);
                vuln.id_db_parent = Convert.ToInt16(frmEx.vulnerabilityDataGridView[7, i].Value);
                vuln.platform = Convert.ToInt16(frmEx.vulnerabilityDataGridView[10, i].Value);
            }
            vuln.a_calcul_type = a_calcul_type;
            vuln.max_risk = 0;
            vuln.min_risk = 0;
            GetPositionX(2, vuln.RectObject.Width);

            Connection connect = new Connection(component, vuln, concol);
            component.AddChild(vuln, connect);
            g_id = g_id + 1;
            vuln.gid_parent = component.gid;
            vuln.gid = g_id;
            vuln.id_db_parent = component.c_id;

            if ((vuln.sec_num <= tot_critic_num_v) & (vuln.sec_num > 0))
            {
               vuln.add_ = "#" + vuln.sec_num;
            }
            mVul_now = vuln.weight;
            risk_vuln = do_Lev4Graph(sender, component, vuln, i);
            vuln.risk = risk_vuln;
            risk_vuln_sum = risk_vuln_sum + risk_vuln;
            tot_max_risk = tot_max_risk + vuln.max_risk;
            tot_min_risk = tot_min_risk + vuln.min_risk;
            vuln_now = vuln;
         }
      }
      component.max_risk = tot_max_risk;
      component.min_risk = tot_min_risk;
   }
   catch (Exception ex)
   {
      MessageBox.Show(" do_Lev3Graph:\r\n" + ex.Message,
                  " Failed", MessageBoxButtons.OK,
                  MessageBoxIcon.Error);
   }
   return  risk_vuln_sum;
}
/// <summary>
/// Program to draw 4 level of out tree - Attacks for particular vulnerability
/// </summary>
/// <param name="sender"></param>
/// <param name="e">risk </param>
///
private double do_Lev4Graph(object sender,Component component, Vulnerability vuln,  int v_row)
{
   //draw all Attacks here and return sum risk
   double risk_sum = 0;
   double tot_max_risk = 0;
   double tot_min_risk = 0;
   try
   {

      TreeBase frmEx = (TreeBase)MdiParent;
      frmEx.vuln_row_num = v_row;
      frmEx.vuln_id_now = vuln.v_id; // our current id in vuln table
```

```
            frmEx.fill_attack_table(sender);
            int nn = frmEx.attackDataGridView.RowCount;


            nn = nn - 1;
            int last_one = nn - 1;
            if (nn > 0)
            {
               for (int i = 0; i < nn; i++)
               {
                  SolidBrush attack_brush = new SolidBrush(frmEx.attack_color);

                  Attack attack = new Attack(point[0], shcol, this.TreePicture, indexSelect, attack_brush);
                  attack.risk = 0;
                  attack.weight = 0;
                  attack.id_db_parent = vuln_now.v_id;
                  attack.Saved = true;
                  attack.Name = Convert.ToString(frmEx.attackDataGridView[1, i].Value);
                  attack.Desc = Convert.ToString(frmEx.attackDataGridView[2, i].Value);

                  attack.at_id = Convert.ToInt16(frmEx.attackDataGridView[0, i].Value);
                  attack.weight = Convert.ToInt16(frmEx.attackDataGridView[5, i].Value);

                  attack.a_calcul_type = a_calcul_type;
                  attack.vul_id = vuln_now.v_id;
                  attack.at_type_id = Convert.ToInt16(frmEx.attackDataGridView[3, i].Value);


                  GetPositionX(3, attack.RectObject.Width);
                  // Adds elements to the collection.
                  Connection connect = new Connection(vuln, attack, concol);
                  vuln.AddChild(attack, connect);
                  g_id = g_id + 1;
                  attack.gid = g_id;
                  attack.gid_parent = vuln.gid;
                  attack.id_db_parent = vuln.v_id;
                  double mCountm_now = do_Lev5Graph(sender, component,  vuln, attack, i);
                  double danger = Convert.ToDouble(attack.weight * vuln.weight);
                  attack.max_risk = danger * component.weight;
                  attack.min_risk =Convert.ToDouble(component.weight * (danger-( (danger/10)*attack.countm_t)));
//
                  attack.risk = Convert.ToDouble(component.weight * (danger-( (danger/10)*attack.countm_now))); //
risk now
                  attack_now = attack;
                  risk_sum = risk_sum + attack.risk;
                  tot_max_risk = tot_max_risk + attack.max_risk;
                  tot_min_risk = tot_min_risk + attack.min_risk;
                  string str = Convert.ToString(attack.risk);

                  string str3 = Convert.ToString(mComponent_now);
                  string str4 = Convert.ToString(mVul_now);

               }
            }

            vuln.max_risk = tot_max_risk;
            vuln.min_risk = tot_min_risk;
         }
         catch (Exception ex)
         {
```

```
            MessageBox.Show(" do_Lev4Graph:\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
        }
        return  risk_sum;
    }
    /// <summary>
    /// Program to do 5 level of out tree -Countermeasures for particular attack
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e">risk reduction </param>
    ///
    private double do_Lev5Graph(object sender,Component component, Vulnerability vuln, Attack attack, int
th_row)
    {
        //draw all Countermeasures here
        double result_m = 1;
        int result_m_sum = 0;
        int result_m_sum_now = 0;
        try
        {

            TreeBase frmEx = (TreeBase)MdiParent;
            frmEx.attack_row_num = th_row; // our current row in attack table
            frmEx.attack_id_now = attack.at_id;
            frmEx.fill_countm_table(sender);
            int nn = frmEx.countermeasureDataGridView.RowCount;
            string str5 = Convert.ToString(nn);
            nn = nn - 1;
            int n_now=0; // number of count implemented
            string str1 = Convert.ToString(nn);
            if (nn > 0) //
            {
                for (int i = 0; i < nn; i++)
                {
                    SolidBrush countm_brush = new SolidBrush(frmEx.countm_color);

                    Countermeasure countm = new Countermeasure(point[0], shcol, this.TreePicture, indexSelect,
countm_brush);

                    countm.weight = 0;
                    countm.id_db_parent = attack_now.at_id;
                    countm.Saved = true;
                    countm.Name = Convert.ToString(frmEx.countermeasureDataGridView[1, i].Value);
                    countm.Desc = Convert.ToString(frmEx.countermeasureDataGridView[2, i].Value);
                    countm.countm_id = Convert.ToInt16(frmEx.countermeasureDataGridView[0, i].Value);

                    countm.implemented = Convert.ToBoolean(frmEx.countermeasureDataGridView[11, i].Value);

                    if (frmEx.countermeasureDataGridView[5, i].Value != null)
                    {
                        countm.at_id = Convert.ToInt16(frmEx.countermeasureDataGridView[5, i].Value);
                    }
                    if (frmEx.countermeasureDataGridView[4, i].Value != null)
                    {
                        countm.v_id = Convert.ToInt16(frmEx.countermeasureDataGridView[4, i].Value);
                    }

                    /// <summary>
                    /// form_type defines what data download to the form
```

```
        /// form_type=1 Open new project from template
        /// form_type=2 Open new project without template
        /// form_type=3 Open existing project for editing
        /// form_type=5 Open new template create
        /// form_type=7 Open existing template for Edit
        /// </summary>
        String delim = "12:00:00 a.m.";

        DateTime dd = new DateTime();
        String str_date="";


        if (frmEx.form_type == 3)
        {
          if (countm.implemented)
          {
            string str_date1 = Convert.ToString(frmEx.countermeasureDataGridView[12, i].Value);
            str_date = Convert.ToString(str_date1);


          }
          else
          {
            dd = DateTime.Today;
            str_date = Convert.ToString(dd);
          }
          String str_d = str_date.TrimEnd(delim.ToCharArray());
          countm.impl_date = Convert.ToString(str_d);
        }
        countm.weight = Convert.ToInt16(frmEx.countermeasureDataGridView[13, i].Value);
        countm.complexity = Convert.ToInt16(frmEx.countermeasureDataGridView[14, i].Value);
        result_m = Convert.ToDouble(countm.weight);
        GetPositionX(4, countm.RectObject.Width);
        Connection connect = new Connection(attack, countm, concol);
        attack.AddChild(countm, connect);
        g_id = g_id + 1;
        countm.gid = g_id;
        countm.gid_parent = attack.gid;
        countm.id_db_parent = attack.at_id;
        if (countm.implemented)
        {
          result_m_sum_now = result_m_sum_now + countm.weight;
          n_now = n_now + 1;
        }
        result_m_sum = result_m_sum + countm.weight;
      }// for
    }
    attack.countm_now = result_m_sum_now;
    attack.countm_t  = result_m_sum;
    if (n_now > 1)
    {
      attack.countm_now = result_m_sum_now / n_now;
    }
    if (nn > 1)
    {
      attack.countm_t = result_m_sum / (nn);
    }
    result_m = attack.countm_now;
}
catch (Exception ex)
```

```
        {
            MessageBox.Show(" do_Lev5Graph:\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
        }
        return result_m;
    }
    #endregion // Do Graph

    #region Do DB

    #region save to db
    public bool save_asset_db_update(object sender, bool my_templ, TreeBase frmEx, Asset asset_now)
    {
        bool ok_saved = false;
        try
        {

            frmEx.assetTableAdapter.UpdateQueryGr(asset_now.Name, asset_now.Desc,
Convert.ToDecimal(asset_now.criticality), Convert.ToDecimal(asset_now.capitalcost),
Convert.ToDecimal(asset_now.risk), Convert.ToDecimal(asset_now.ecl), Convert.ToInt16(asset_now.at_id),
my_templ, asset_now.random, Convert.ToInt16(asset_now.setting_id), Convert.ToInt16(asset_now.calcul_type),
Convert.ToInt16(asset_now.weight), Convert.ToDecimal(asset_now.max_risk),
Convert.ToDecimal(asset_now.min_risk), false, 0, System.DateTime.Today, asset_now.a_id);
            //                  UPDATE `asset` SET `a_name` = ?, `a_desc` = ?, `a_critic` = ?,
`a_cap_cost` = ?,                   `a_final_risk` = ?,                 `a_ecl` = ?,    `at_id` = ?,      `templ` = ?,
`a_random` = ?, `a_settings` = ?, `a_calcul_type` = ?, `a_weight` = ? WHERE ((`asset_id` = ?) AND ((? = 1 AND
`a_name` IS NULL) OR (`a_name` = ?)) AND ((? = 1 AND `a_desc` IS NULL) OR (`a_desc` = ?)) AND ((? = 1
AND `a_critic` IS NULL) OR (`a_critic` = ?)) AND ((? = 1 AND `a_cap_cost` IS NULL) OR (`a_cap_cost` = ?)) AND
((? = 1 AND `a_final_risk` IS NULL) OR (`a_final_risk` = ?)) AND ((? = 1 AND `a_ecl` IS NULL) OR (`a_ecl` = ?))
AND ((? = 1 AND `at_id` IS NULL) OR (`at_id` = ?)) AND ((? = 1 AND `templ` IS NULL) OR (`templ` = ?)) AND ((?
= 1 AND `a_random` IS NULL) OR (`a_random` = ?)) AND ((? = 1 AND `a_settings` IS NULL) OR (`a_settings` =
?)) AND ((? = 1 AND `a_calcul_type` IS NULL) OR (`a_calcul_type` = ?)) AND ((? = 1 AND `a_weight` IS NULL)
OR (`a_weight` = ?)))

            asset_now.Saved = true;
            ok_saved = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(" save_asset_db:\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
        }
        return ok_saved;

    }
    public bool save_asset_db_insert(object sender, bool my_templ, TreeBase frmEx, Asset asset_now)
    {
        bool ok_saved = false;
        asset_now.random = randomString(10);
        try
        {
            frmEx.asset_row_num = frmEx.assetDataGridView.NewRowIndex; //assetDataGridView1
            asset_now.random = randomString(10);
            if (asset_now.at_id == 0)
            {
                asset_now.at_id = frmEx.at_id_now;
            }
            frmEx.assetTableAdapter.InsertQueryGr(asset_now.Name, asset_now.Desc,
```

```
Convert.ToDecimal(asset_now.criticality), Convert.ToDecimal(asset_now.capitalcost),
Convert.ToDecimal(asset_now.risk), Convert.ToDecimal(asset_now.ecl), asset_now.at_id, my_templ,
asset_now.random, 1, 1, asset_now.weight, Convert.ToDecimal(asset_now.max_risk),
Convert.ToDecimal(asset_now.min_risk), false, 0, System.DateTime.Today);
            //INSERT INTO asset
            //     (a_name, a_desc, a_critic, a_cap_cost, a_final_risk, a_ecl, at_id, templ, a_random, a_settings,
a_calcul_type, a_weight)

            asset_now.Saved = true;
            ok_saved = true;

asset_now.a_id=Convert.ToInt16(frmEx.assetTableAdapter.ScalarQueryFindId_ByRandom(asset_now.random));

        }
        catch (Exception ex)
        {
            MessageBox.Show(" save_asset_db:\r\n" + ex.Message,
                        " Failed", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
        }
        return ok_saved;

    }
     public bool save_asset_db(object sender, bool my_templ, bool save_as_new)
                    {
        bool ok_saved = false;
        TreeBase frmEx = (TreeBase)MdiParent;
        frmEx.asset_id_now = asset_now.a_id;

        if ((asset_now != null) && (!save_as_new))
        {
            ok_saved = save_asset_db_update(sender, my_templ, frmEx, asset_now);
        }
        if (save_as_new)
        {

            ok_saved = save_asset_db_insert(sender, my_templ, frmEx, asset_now);


        }
        return ok_saved;
                    }


    public bool save_component_db_insert(object sender, bool my_templ, TreeBase frmEx, Component
comp_now, int as_id)
    {
        bool ok_saved = false;
        comp_now.random = randomString(10);
        try
        {

            frmEx.componentTableAdapter.InsertQueryGr(comp_now.Name, comp_now.desc, as_id,
Convert.ToDecimal(comp_now.criticality), Convert.ToDecimal(comp_now.capitalcost),
Convert.ToDecimal(comp_now.risk), Convert.ToDecimal(comp_now.ecl), my_templ, comp_now.random,
Convert.ToInt16(comp_now.weight), Convert.ToDecimal(comp_now.max_risk),
Convert.ToDecimal(comp_now.min_risk));
            comp_now.Saved = true;
            ok_saved = true;
            comp_now.c_id =
```

```
 Convert.ToInt16(frmEx.componentTableAdapter.ScalarQuery_findID_byRandom_asId(asset_now.a_id,comp_no
w.random));

        }
        catch (Exception ex)
        {

            MessageBox.Show(" save_component_db_insert:\r\n" + ex.Message +" "+ ex.Source,
                        " Failed", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
        }
        return ok_saved;
    }


    public bool save_component_db_update(object sender, bool my_templ, TreeBase frmEx, Component
comp_now )
    {
        bool ok_saved = false;
        frmEx.comp_id_now = comp_now.c_id;
        try
        {
            frmEx.componentTableAdapter.UpdateQueryGr(comp_now.Name, comp_now.desc,Convert.ToInt16
(comp_now.asset_id), Convert.ToDecimal(comp_now.criticality), Convert.ToDecimal(comp_now.capitalcost),
Convert.ToDecimal(comp_now.risk), Convert.ToDecimal(comp_now.ecl), my_templ, comp_now.random,
Convert.ToInt16(comp_now.weight), Convert.ToDecimal(comp_now.max_risk),
Convert.ToDecimal(comp_now.min_risk), comp_now.c_id);
            comp_now.Saved = true;
            ok_saved = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(" save_component_db_update:\r\n" + ex.Message,
                        " Failed", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
        }
        return ok_saved;
    }


    public bool save_component_db(object sender, bool my_templ)
    {
        bool ok_saved = false;
        int comp_id = comp_now.c_id;
        TreeBase frmEx = new TreeBase(); //// (TreeBase)MdiParent;
        if (comp_id != 0)
        {
            ok_saved = save_component_db_update(sender, my_templ, frmEx, comp_now);

        }
        else
        {
            int up_id = shcol.GetParGId(1, g_id);
            int as_id = shcol.GetRootDB_ID();                //shcol.GetDbId(0, up_id);
            ok_saved = save_component_db_insert(sender, my_templ, frmEx, comp_now, as_id);
        }
        return  ok_saved;
    }
```

```
    public bool save_vulnerability_db_insert(object sender, bool my_templ, TreeBase frmEx, Vulnerability
vuln_now, int comp_id )
    {
        bool ok_saved = false;
        vuln_now.random = randomString(10);
        try
        {
            string str3 = Convert.ToString(comp_id);

            string s_cr = Convert.ToString(vuln_now.Name);
            frmEx.vulnerabilityTableAdapter.InsertQueryGr(vuln_now.Name, vuln_now.desc, vuln_now.weight,
vuln_now.vt_id, vuln_now.example, Convert.ToDecimal(vuln_now.risk), comp_id, my_templ, vuln_now.random,
vuln_now.platform);

            vuln_now.Saved = true;
            vuln_now.v_id =
Convert.ToInt16(frmEx.vulnerabilityTableAdapter.ScalarQuery_FindId_byName_byRandom(vuln_now.Name,
vuln_now.random));

            ok_saved = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(" save_vulnerability_db_insert:\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
        }
        return ok_saved;
    }


    public bool save_vulnerability_db(object sender, bool my_templ,  int gnid)
    {
        bool ok_saved = false;
       // int g_id = shcol.GetIndexSelectObject();
        int vuln_id = shcol.GetDbId(2, gnid);


        vuln_now = (Vulnerability)shcol[gnid];
        //TreeBase frmEx = (TreeBase)MdiParent;
        TreeBase frmEx = new TreeBase();
        if (vuln_id != 0)
        {
            string st = vuln_now.Name;
            frmEx.vuln_id_now = vuln_now.v_id;
            try
            {
                frmEx.vulnerabilityTableAdapter.UpdateQueryGr(vuln_now.Name, vuln_now.desc, vuln_now.Weight,
Convert.ToInt16(vuln_now.vt_id), vuln_now.example, Convert.ToDecimal(vuln_now.risk),
Convert.ToInt16(vuln_now.id_db_parent), my_templ, vuln_now.random, Convert.ToInt16(vuln_now.platform),
Convert.ToInt16(vuln_now.v_id));
                vuln_now.Saved = true;
                ok_saved = true;
            }
            catch (Exception ex)
            {
                MessageBox.Show(" save_vulnerability_db UPDATE:\r\n" + ex.Message,
                        " Failed", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
            }
```

```
        }
        else
        {
            ok_saved = save_vulnerability_db_insert(sender, my_templ, frmEx, vuln_now, vuln_now.id_db_parent);
//comp_id);
        }
            return ok_saved;
    }


    public bool save_attack_db_insert(object sender, bool my_templ, TreeBase frmEx,  Attack attack_now, int
vul_id)
    {
        bool ok_saved = false;
        attack_now.random = randomString(10);

        try
        {
            //INSERT INTO `attack`              (`at_name`,   `at_desc`,    `at_type_id`,         `v_id`,
`at_weight`,       `source`,       `at_risk`,              `at_severity`,    `at_example`,       `at_more_names`,
`at_web_site`,    `at_test`,    `templ`,   `at_random`) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
            attack_now.vul_id = attack_now.id_db_parent; ////vul_id;
            frmEx.attackTableAdapter.InsertQueryGr(attack_now.Name, attack_now.desc, attack_now.at_type_id,
attack_now.vul_id, attack_now.weight, attack_now.source, Convert.ToDecimal(attack_now.risk),
attack_now.severity, attack_now.example, attack_now.more_names, attack_now.website, attack_now.test,
my_templ, attack_now.random);
            attack_now.at_id =
Convert.ToInt16(frmEx.attackTableAdapter.ScalarQuery_FindId_byRandom(attack_now.random));
            attack_now.Saved = true;
            ok_saved = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(" save_attack_db: \r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
        }
        return ok_saved;
    }

    public bool save_attack_db(object sender, bool my_templ, bool save_as_new, int gnid)
    {
        bool ok_saved = false;
        //int g_id = shcol.GetIndexSelectObject();
        int attack_id = shcol.GetDbId(3, gnid);
        if (save_as_new)
        {
            attack_id = 0;

        }
        //TreeBase frmEx = (TreeBase)MdiParent;
        TreeBase frmEx = new TreeBase();
        attack_now = (Attack)shcol[gnid];
        //attack_now.vul_id = vul_id;
        if (attack_id != 0)
        {
            string st = attack_now.Name;
            frmEx.attack_id_now = attack_now.at_id;
            try
```

```
            {
                attack_now.vul_id = attack_now.id_db_parent;
                frmEx.attackTableAdapter.UpdateQueryGr(attack_now.Name, attack_now.desc,
Convert.ToInt16(attack_now.at_type_id), Convert.ToInt16(attack_now.vul_id),
Convert.ToInt16(attack_now.weight), attack_now.source, Convert.ToDecimal(attack_now.risk),
Convert.ToInt16(attack_now.severity), attack_now.example, attack_now.more_names, attack_now.website,
attack_now.test, my_templ, attack_now.random, attack_now.at_id);
                attack_now.Saved = true;
                ok_saved = true;
            }
            catch (Exception ex)
            {
                MessageBox.Show(" save_attack_db Update: \r\n" + ex.Message,
                        " Failed", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
            }

        }
        else
        {
            int up_id = shcol.GetParGId(3, gnid);
            int vul_id = shcol.GetDbId(2, up_id);
            ok_saved = save_attack_db_insert(sender, my_templ, frmEx, attack_now, vul_id);
        }
        return  ok_saved;
    }

    public bool save_countm_db_insert(object sender, bool my_templ,  TreeBase frmEx,Countermeasure
countm_now)
    {        bool ok_saved = false;

        countm_now.random = randomString(10);
        DateTime dd = new DateTime();
        countm_now.at_id = countm_now.id_db_parent;

        try
        {
            if (countm_now.implemented)
            {
                dd = DateTime.Today;//Convert.ToDateTime(countm_now.impl_date),
            }
            else
            {
                countm_now.impl_date = null;
            }
            frmEx.countermeasureTableAdapter.InsertQueryGr(countm_now.Name, countm_now.desc,
countm_now.example, countm_now.v_id, countm_now.at_id, countm_now.source, countm_now.pl_id,
countm_now.website, countm_now.random, my_templ, dd, countm_now.implemented, countm_now.complexity,
countm_now.weight);
            countm_now.Saved = true;
            ok_saved = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(" save_countm_db :\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
        }
        return ok_saved;
    }
```

```
public bool save_countm_db(object sender, bool my_templ, bool save_as_new, int gnid)
{
   bool ok_saved = false;
   //int g_id = shcol.GetIndexSelectObject();
   string str = Convert.ToString(gnid);
   //TreeBase frmEx = (TreeBase)MdiParent;
   TreeBase frmEx = new TreeBase();
   int countm_id = shcol.GetDbId(4, gnid);
   DateTime dd = new DateTime();
   if (save_as_new)
   {
      countm_id = 0;

   }
   countm_now = (Countermeasure)shcol[gnid];
   if (countm_id != 0)
   {

      string st = countm_now.Name;
      frmEx.countm_id_now = countm_now.countm_id;
      try
      {
         if (!(countm_now.implemented))
         {
            countm_now.impl_date = null;
         }
         dd = Convert.ToDateTime(countm_now.impl_date);
         if ((countm_now.implemented) && (countm_now.impl_date== null ))
         {
            dd = DateTime.Today;//Convert.ToDateTime(countm_now.impl_date),
         }


         countm_now.at_id = countm_now.id_db_parent;
         frmEx.countermeasureTableAdapter.UpdateQueryGr(countm_now.Name, countm_now.desc,
countm_now.example, countm_now.v_id, countm_now.at_id, countm_now.source, countm_now.pl_id,
countm_now.website, countm_now.random, my_templ, dd, countm_now.implemented, countm_now.complexity,
countm_now.weight, countm_now.countm_id);
         countm_now.Saved = true;
         ok_saved = true;
      }
      catch (Exception ex)
      {
         MessageBox.Show(" save_countm_db :\r\n" + ex.Message,
                  " Failed", MessageBoxButtons.OK,
                  MessageBoxIcon.Error);
      }

   }
   else
   {
      //int up_id = shcol.GetParGId(4, gnid);
      //int att_id = shcol.GetDbId(3, up_id);
      ok_saved = save_countm_db_insert(sender, my_templ, frmEx, countm_now);
   }
   return ok_saved;
}

private int asset_newSave(object sender, TreeBase frmEx, Asset asset_now, bool my_templ)
```

```
    {
        int id_db = 0;
        bool ok_saved = save_asset_db_insert(sender, my_templ, frmEx, asset_now);
        frmEx.assetTableAdapter.Fill(frmEx.db_vt_1DataSet.asset);
        //id_db =
Convert.ToInt16(frmEx.assetTableAdapter.ScalarQueryFindID_byName(Convert.ToString(asset_now.Name),
Convert.ToString(asset_now.random)));
        id_db =
Convert.ToInt16(frmEx.assetTableAdapter.ScalarQueryFindId_ByRandom(Convert.ToString(asset_now.random)))
;
        asset_now.a_id = id_db;
        frmEx.assetDataGridView.Visible = true;
        //frmEx.form_type = 3;
        //frmEx.Text = "Existing Project";
        // this.menuItemSave.Enabled = true;
        this.menuSaveTemAsPr.Enabled = false;
        frmEx.assetTableAdapter.FillBy_Asset_Id(frmEx.db_vt_1DataSet.asset, asset_now.a_id);
        return id_db;
    }
    /// <summary>
    /// To Show message OK to save or "No"
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    /// <returns>ture of false depends of user respond</returns>


    private bool save_OK(object sender, EventArgs e)
    {
        bool result = false;
        DialogResult res;
        res = MessageBox.Show(" Do you want to Save record to Database? :\r\n",
                              " Save ", MessageBoxButtons.YesNo,
                              MessageBoxIcon.Question);
        if (res == DialogResult.Yes)
        {
            result = true;
        }
        return result;

    }


    private void bnSaveGr_Click(object sender, EventArgs e)
    {
        frmS = new form_Pict();
        frmS.con_type = con_type;
        frmS.shcol3 = shcol;
        frmS.concol3 = concol;
        frmS.picW = TreePicture.Width;
        frmS.picH = TreePicture.Height;
        frmS.ShowDialog();
    }

    private void menuItem_as_NewTempl_Click(object sender, EventArgs e)
    {
        my_templ = true;
        TreeBase frmEx = (TreeBase)MdiParent;

        bool save_as_new = true;
```

```
    saveAll_now(sender, e, my_templ, save_as_new);
  frmEx.form_type = 7;
  frmEx.Text = "Existing Templates";
}

private void menuSaveTemAsPr_Click(object sender, EventArgs e)
{
  // do save all schema
  my_templ = false;
  TreeBase frmEx = (TreeBase)MdiParent;

  bool save_as_new = true;
  saveAll_now(sender, e, my_templ, save_as_new);
  frmEx.form_type = 3;
  frmEx.Text = "Existing Project";
}
private void saveAll_now(object sender, EventArgs e, bool my_templ, bool save_as_new)
{
  // do save all schema
  //int numberShape = shcol.CountShape(indexShape);
  int o_type = 0; // GetTypeObjects();
  bool ok_saved = false;
  int id_db = asset_now.a_id; // asset_id now
  //int id_com = 0;// com_id now
  //int id_vul = 0;// com_id now
  //int id_thr = 0;// com_id now
  //int id_countm = 0;// com_id now
  panel_wait.Visible = true;

  progressBar1.Minimum = 0;
  // Sets the progress bar's maximum value to a number representing
  // all operations complete -- in this case, all five files read.
  progressBar1.Maximum = shcol.Count + 5;
  // Sets the Step property to amount to increase with each iteration.
  // In this case, it will increase by one with every file read.
  progressBar1.Step = 1;

  TreeBase frmEx = (TreeBase)MdiParent;
  //progressBar

  for (int i = 0; i < shcol.Count; i++)
  {
    progressBar1.PerformStep();

    // shcol.PaintObjects(g1, i);
    ShapeBase sb = (ShapeBase)shcol[i];
    if (sb is Asset) o_type = 0;
    if (sb is Component) o_type = 1;
    if (sb is Vulnerability) o_type = 2;
    if (sb is Attack) o_type = 3;
    if (sb is Countermeasure) o_type = 4;
    string str = Convert.ToString(i);
    int _id = shcol.GetDbId(1, i);
    string str3 = Convert.ToString(_id);
    string str2 = Convert.ToString(o_type);

    switch (o_type)
    {

      case 0:
```

```
                {
                    if ((frmEx.form_type == 2) || (frmEx.form_type == 5) || (frmEx.form_type == 1) ||
((frmEx.form_type == 3) && (save_as_new)))
                    {
                        ok_saved = false;
                        asset_now = (Asset)shcol[0];
                        id_db = asset_newSave(sender, frmEx, asset_now, my_templ);
                        if (id_db > 0) ok_saved = true;
                        // frmEx.componentTableAdapter.Fill(frmEx.db_vt_1DataSet.component);
                        //frmEx.assetTableAdapter.FillBy_Asset_Id(frmEx.db_vt_1DataSet.asset, id_db);

                        if (frmEx.form_type == 5)
                        {
                            frmEx.form_type = 7;
                            frmEx.Text = "Existing Templates";
                        }
                        if ((frmEx.form_type == 1) || (frmEx.form_type == 2))
                        {
                            frmEx.form_type = 3;
                            frmEx.Text = "Existing Project";
                        }
                        frmEx.assetDataGridView_DoubleClick(sender, e);
                    }
                    else
                    {
                        if (save_asset_db(sender, my_templ, false))
                        {
                            frmEx.assetDataGridView_DoubleClick(sender, e);
                        }
                    }
                    break;

                }

                case 1:
                    {
                        //save_component_db(sender, my_templ, true, i);
                        comp_now = (Component)shcol[i];
                        comp_now.asset_id = asset_now.a_id;
                        if (save_as_new)
                        {
                            comp_now.c_id = 0;
                        }
                        save_component_db(sender, my_templ);
                        //frmEx.saveDB(sender, e);
                        frmEx.componentTableAdapter.Fill(frmEx.db_vt_1DataSet.component);

                        break;
                    }

                case 2:
                    {

                        vuln_now = (Vulnerability)shcol[i];
                        //
                        //int g_id = shcol.GetIndexSelectObject();
                        if (save_as_new)
                        {
                            vuln_now.v_id = 0;
                            vuln_now.id_db_parent = comp_now.c_id;
```

```
                }
                //vuln_now.id_db_parent = comp_now.c_id;
                int g_id = vuln_now.gid;
                int gid_par = shcol.GetParGId(o_type, g_id);
                int _id_db_par = shcol.GetDbId(o_type - 1, gid_par);
                if (vuln_now.id_db_parent == 0)
                {
                    vuln_now.id_db_parent = _id_db_par;//comp_now.c_id; //
                }
                ok_saved = save_vulnerability_db(sender, my_templ, i);
                vuln_now.v_id =
Convert.ToInt16(frmEx.vulnerabilityTableAdapter.ScalarQuery_FindId_byName_byRandom(Convert.ToString(vuln
_now.Name), Convert.ToString(vuln_now.random)));

                break;
            }

        case 3:
            {
                //save_attack_db(sender, my_templ, true, i);
                attack_now = (Attack)shcol[i];
                if (save_as_new)
                {
                    attack_now.at_id = 0;
                    attack_now.id_db_parent = vuln_now.v_id;
                }

                int g_id = attack_now.gid;
                int gid_par = shcol.GetParGId(o_type, g_id);
                int _id_db_par = shcol.GetDbId(o_type - 1, gid_par);
                if (attack_now.id_db_parent == 0)
                {
                    attack_now.id_db_parent = _id_db_par;//vuln_now.v_id;
                }

                ok_saved = save_attack_db(sender, my_templ, save_as_new, i);

                break;
            }
        case 4:
            {
                countm_now = (Countermeasure)shcol[i];
                if (save_as_new)
                {
                    countm_now.countm_id = 0;
                    countm_now.id_db_parent = attack_now.at_id;
                }
                int g_id = countm_now.gid;
                int gid_par = shcol.GetParGId(o_type, g_id);
                int _id_db_par = shcol.GetDbId(o_type - 1, gid_par);

                if (countm_now.id_db_parent == 0)
                {
                    countm_now.id_db_parent = _id_db_par; // attack_now.at_id;
                }
                ok_saved = save_countm_db(sender, my_templ, save_as_new, i);

            }
            break;
    }// end swich
```

```
        }// end for

        string at_name = "";
        this.menuSaveTemAsPr.Enabled = false;
        at_name =
frmEx.asset_typeTableAdapter.ScalarQuery_findName_ById(Convert.ToInt16(asset_now.at_id));
        int resultIndex = frmEx.comBox_aType.FindStringExact(at_name);
        frmEx.comBox_aType.SelectedIndex = resultIndex;
        frmEx.comBox_aType_SelectedIndexChanged(sender, e);
        frmEx.assetTableAdapter.FillBy_Asset_Id(frmEx.db_vt_1DataSet.asset, id_db);

        // TODO: This line of code loads data into the 'db_vt_1DataSet.component' table. You can move, or
remove it, as needed.
        frmEx.componentTableAdapter.Fill(frmEx.db_vt_1DataSet.component);

        // TODO: This line of code loads data into the 'db_vt_1DataSet.vulnerability' table. You can move, or
remove it, as needed.
        frmEx.vulnerabilityTableAdapter.Fill(frmEx.db_vt_1DataSet.vulnerability);

        // TODO: This line of code loads data into the 'db_vt_1DataSet.attack' table. You can move, or remove it,
as needed.
        frmEx.attackTableAdapter.Fill(frmEx.db_vt_1DataSet.attack);
        // TODO: This line of code loads data into the 'db_vt_1DataSet.countm' table. You can move, or remove it,
as needed.
        frmEx.countermeasureTableAdapter.Fill(frmEx.db_vt_1DataSet.countermeasure);

        frmEx.assetDataGridView_DoubleClick(sender, e);
        panel_wait.Visible = false;
    }
    #endregion // save to db

    #region delete from DB
    /// <summary>
    /// delete item from db
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="frmEx"></param>
    /// <param name="id_db"></param>
    /// <returns>if deleted retutn - true</returns>
    private bool delete_item_db(object sender, EventArgs e)      //ShapeBase rsb
    {
        bool deleted = false;
        int g_id = shcol.GetIndexSelectObject();
        string str = Convert.ToString(g_id);
        progressBar1.PerformStep();
        //int id_db = 0;
        int o_type = GetTypeObjects();
        string str55 = Convert.ToString(o_type);
        TreeBase frmEx = (TreeBase)MdiParent;
        asset_now = (Asset)shcol[0];
        progressBar1.PerformStep();
        //if exist correstpondent object at DB
        switch (o_type)
        {

            case 0:
                {
                    if (delete_OK(sender, e))
                    {
                        deleted = delete_asset_db(sender, frmEx, asset_now.a_id);
```

```
                progressBar1.PerformStep();
            }
            break;
        }

    case 1:
        {
            if (delete_OK(sender, e))
            {
                comp_now = (Component)shcol[g_id];
                deleted = delete_component_db(sender, frmEx, comp_now.c_id);
                progressBar1.PerformStep();
            }
            break;
        }

    case 2:
        {
            if (delete_OK(sender, e))
            {
                vuln_now = (Vulnerability)shcol[g_id];
                deleted = delete_vulnerability_db(sender, frmEx, vuln_now.v_id);
                progressBar1.PerformStep();
            }
            break;
        }

    case 3:
        {
            if (delete_OK(sender, e))
            {
                attack_now = (Attack)shcol[g_id];
                deleted = delete_attack_db(sender, frmEx, attack_now.at_id);
                progressBar1.PerformStep();
            }
            break;
        }
    case 4:
        {

            if (delete_OK(sender, e))
            {
                countm_now = (Countermeasure)shcol[g_id];
                deleted = delete_countm_db(sender, frmEx, countm_now.countm_id);
                progressBar1.PerformStep();
            }
            break;
        }


}


if (o_type > 0)
{
    frmEx.assetTableAdapter.FillBy_Asset_Id(frmEx.db_vt_1DataSet.asset, asset_now.a_id);
    frmEx.assetDataGridView_DoubleClick(sender, e);
}
progressBar1.PerformStep();
return deleted;
```

```
}
private bool delete_OK(object sender, EventArgs e)
{
   bool result = false;
   DialogResult res;
   res = MessageBox.Show(" Do you want to Delete record from Database? :\r\n",
                         " Delete ", MessageBoxButtons.YesNo,
                         MessageBoxIcon.Question);


   if (res == DialogResult.Yes)
   {
      result = true;
   }
   return result; // result;

}
/// <summary>
/// delete asset from db
/// </summary>
/// <param name="sender"></param>
/// <param name="frmEx"></param>
/// <param name="id_db"></param>
/// <returns>if deleted retutn - true</returns>
 public bool delete_asset_db(object sender, TreeBase frmEx,  int as_id)
{
   bool ok_delete = false;

   try
   {
      string str = Convert.ToString(as_id);
      progressBar1.PerformStep();
      frmEx.assetTableAdapter.DeleteQuery_byID(as_id);

      ok_delete = true;
   }
   catch (Exception ex)
   {
      MessageBox.Show(" delete_asset_db:\r\n" + ex.Message,
               " Failed", MessageBoxButtons.OK,
               MessageBoxIcon.Error);
   }
   return ok_delete;
}
/// <summary>
/// delete component from db
/// </summary>
/// <param name="sender"></param>
/// <param name="frmEx"></param>
/// <param name="id_db"></param>
/// <returns>if deleted retutn - true</returns>
public bool delete_component_db(object sender, TreeBase frmEx, int c_id)
{
   bool ok_delete = false;

   try
   {
      string str = Convert.ToString(c_id);
      progressBar1.PerformStep();
      frmEx.componentTableAdapter.DeleteQuery_byID(c_id);
```

```
            ok_delete = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(" delete_component_db:\r\n" + ex.Message,
                        " Failed", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
        }
        return ok_delete;
    }
    /// <summary>
    /// delete vulnerability from db
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="frmEx"></param>
    /// <param name="id_db"></param>
    /// <returns>if deleted retutn - true</returns>
    public bool delete_vulnerability_db(object sender, TreeBase frmEx, int v_id)
    {
        bool ok_delete = false;

        try
        {
            string str = Convert.ToString(v_id);
            progressBar1.PerformStep();
            frmEx.vulnerabilityTableAdapter.DeleteQuery_byID(v_id);

            ok_delete = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(" delete_vulnerability_db:\r\n" + ex.Message,
                        " Failed", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
        }
        return ok_delete;
    }
    /// <summary>
    /// delete attack from db
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="frmEx"></param>
    /// <param name="id_db"></param>
    /// <returns>if deleted retutn - true</returns>
    public bool delete_attack_db(object sender, TreeBase frmEx, int at_id)
    {
        bool ok_delete = false;

        try
        {
            string str = Convert.ToString(at_id);
            progressBar1.PerformStep();
            frmEx.attackTableAdapter.DeleteQuery_byID(at_id);

            ok_delete = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(" delete_attack_db:\r\n" + ex.Message,
```

```
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
            }
            return ok_delete;
        }
        /// <summary>
        /// delete countermeasure from db
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="frmEx"></param>
        /// <param name="id_db"></param>
        /// <returns>if deleted retutn - true</returns>
        public bool delete_countm_db(object sender, TreeBase frmEx, int countm_id)
        {
            bool ok_delete = false;

            try
            {
                string str = Convert.ToString(countm_id);
                progressBar1.PerformStep();
                frmEx.countermeasureTableAdapter.DeleteQuery_byID(countm_id);

                ok_delete = true;
            }
            catch (Exception ex)
            {
                MessageBox.Show(" delete_countm_db:\r\n" + ex.Message,
                    " Failed", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
            }
            return  ok_delete;
        }

        #endregion // delete db

        #endregion // to db

        #region Change Graph setting



        private void bnPan_close_Click_1(object sender, EventArgs e)
        {
            paGrProperty.Visible = false;
        }



        private void bnSaveSettings_Click(object sender, EventArgs e)
        {
            //UpdateQuerySet_Settings
            TreeBase frmEx = (TreeBase)MdiParent;
            int set_id =
Convert.ToInt16(this.settingsTableAdapter.ScalarQueryFind_Id_byName(comBox_Settings.Text));
            frmEx.assetTableAdapter.UpdateQuerySet_Settings(set_id, asset_now.a_id);
            asset_now.setting_id = set_id;
            paGrProperty.Visible = false;
            menuItemSaveAll_Click(sender, e);
```

```
        }
        #endregion // Change Graph setting

            }
}
```