

Risk Optimization: A Technical Introduction

Abstract

IT Risk is often thought of in terms of exploits and attack vectors, a reasonable approach when responding to (or preventing) specific incidents. As the number of threats and vulnerabilities continues to grow, a strategy of mitigating all risk equally becomes unsustainable. At the same time, focus and spend is increasingly driven by the perception of susceptibilities instead of material probabilities and impact.

Changing this requires a holistic approach. This paper describes the major components of a risk management system whose goal is prioritize and optimize residual risk against spend.

Addressing risk holistically starts with an understanding of why risks need to be controlled instead of what risks need to be controlled. Vulnerabilities are first classified into groups that represent the type of undesirable impact they would have (all risks exist because they represent the potential for an undesirable occurrence, whether driven by business need or mandated by a compliance directive).

Each vulnerability in each group is scored, and a total is computed that represents the aggregate risk of each type of problem. At that point, one can manage risk investment on a portfolio basis, applying spend and effort to those groups with a high residual risk of an undesirable occurrence and less to areas where the outcome is already well contained. Benchmarks can be set for total risk of each issue and program spend prioritized to prevent over-investing in some areas and under-investing in others. In short, a focus on types of risk from an impact perspective enables optimization of risk.

Approach

Managing risk holistically requires more than just classification of vulnerabilities by type of impact. It demands a decision support infrastructure with four components: a Vulnerability Management Framework, Vulnerability Storage, Vulnerability Processing, and Visualization.

1. Vulnerability Management Framework

The primary goal of the management framework is to ensure that information collected about risks is complete and consistent. It begins with a set of rules that govern what information to collect and how that information is processed. Completeness is necessary for two reasons: an accurate view of residual risk clearly requires that the data exist (you can't analyze what you didn't collect). The second reason for completeness is shared by the requirement for consistency; data is often sourced from very disparate systems, and the opportunity for a loss of integrity within a risk management system is highest at this point.

Consider a financial institution with multiple businesses, and a set of risk data repositories (SIEM and vulnerability scanning tools, self-assessment databases, audit findings, etc.) in each business. Should a back-office business report on all residual risks and a retail business report only on some, it is possible to substantially underestimate the total risk of leaking

customer PII (Personally Identifiable Information). Completeness is necessary, but not sufficient. If each business has a different definition as to what constitutes "high impact," aggregating and making risk-based decisions across those businesses becomes meaningless. Completeness and consistency of collected vulnerability information is paramount.

An interesting side effect of aggregating large sets of vulnerability data into portfolios is that it tends to be self-correcting. Any large system is bound to have errors. In vulnerability data, those errors are often in either classification or scoring. If a consistent process is followed, and is well designed to ensure that errors are random (i.e., human errors vs. those biased by the methodology towards one set of answers), the total values will tend to be error free.

While the probability of any individual vulnerability containing an error is unchanged, it is increasingly probable that an inverse (a similar error in the other direction) exists. We may have misclassified type "A" as type "B," but we are equally likely to misclassify another "B" as an "A." This self-correcting effect increases with the amount of data analyzed.

Once the need for data consistency is understood, along with the importance of a well-designed system to eliminate bias of errors, the next natural step is the definition and implementation of processes that support these needs. A set of best practices for data collection and management form the framework for vulnerability management. These are analogous to the rigid requirements of an ERP system, where value is derived from consistent implementation.

There may be more than one "right" way to collect and manage this data (though there are very many "wrong" ways), but it is critical that one and only one right way is implemented. These best practices, derived from years of experience, are used to audit a current snapshot of risk data tools and processes. Making decisions based on aggregated risk data is unwise without this first, important step.

Embedded within these best practices are basic assumptions as to how the data will be used. In the case of risk optimization, materiality and the ability to filter are key. Almost by definition, immaterial risks do not need to be controlled, even if some of their attributes are significant. Value is derived immediately from a framework that enforces best practices and totals risk scores, but substantial additional value comes from feeding the output of this framework through filters that allow focus on the most interesting data. Filtering, often through visualization, can identify pockets of material residual risk worthy of focus vs. areas where investment will have little effect.

The Aberdeen Group (Boston, MA, <http://aberdeen.com>) reports that billions of dollars are wasted each year by companies seeking to control risks that are not material. Materiality is the intersection of three distinct components, each of which must be scored independently:

1. The first, Susceptibility, defines the probability of an occurrence succeeding. It incorporates both compensating controls as well as the likelihood that the vulnerability exists as defined (the probability that an occurrence can be affected).
2. Exploitability measures the probability of an occurrence happening, and captures the difficulty of execution (the probability that one can or will be attempted). Note that exploitability tends to change over time as the broader environment changes, making this

component particularly useful for trending.

3. The final component is Impact, which describes the potential damage (the criticality). It is also used to classify a Vulnerability by the type of undesirable outcome to be avoided. This classification is key to acting and investing in a risk-based manner. What matters is ultimately managing the probability of bad things happening. Understanding how likely those bad things are, by category, enables us to decide if additional controls are warranted or if they would only serve to control a level of risk already deemed acceptable.

2. Vulnerability Storage

Once the framework is defined and consistent information management processes in effect, the mechanics of data collection can begin. Criteria here clearly include a robust and extensible storage infrastructure, but also one designed to enforce the framework's best practices. This is done by tightly controlling the APIs and administrative interfaces used to create and populate database tables. A large variation in data sources is inevitable as both homegrown risk repositories and vendor-provided tools are encountered. Vulnerability data must be transformed, and often supplemented, to meet the framework mandated criteria. The design of the tables in the database used to store this information must also be strictly controlled.

It is important that this strictness does not affect extensibility. Certain parameters are required in a consistent manner for the Vulnerability data to have value, but other parameters may be locally desirable. The APIs and data storage system should allow for both new attributes to be added, as well as for segmentation of existing classes of data. Examples include reporting on risk from the perspective of each business, or creating locally meaningful subcategories of the risk data classes already defined. As long as the minimum set of requirements are met, additional flexibility is warranted.

3. Vulnerability Processing and Business Logic

Processing occurs once the data describing vulnerability attributes has been loaded into a common store. This processing includes scoring susceptibility, exploitability, and impact through a well-defined model (several exist: some particularly good ones involve derivations of CVSS, the Common Vulnerability Scoring System). Other processing may include various attribute groupings to support specific report formats, comparison of calculated scores against internal targets and industry benchmarks, and trending. Once again, an administrative interface that enables company specific analysis is important.

Of great interest are benchmarks. It's one thing to compute a score of total material risk of a specific undesirable event. To really understand that score, and to defend the level of investment as sufficient or justify more, benchmarks should be used. Not all risk can or should be eliminated. Benchmarks are metrics that set the bar. Once again, the framework's consistency enables meaningful comparisons of scores against those metrics.

Trend analysis takes benchmark comparisons to their next level. Trends can be used to demonstrate improved risk performance over time, but can be much more powerful. As the previously described exploitability vectors change (due to changes in the nature of the threat, availability of tools that simplify attacks, etc.) trends can be used to see the emergence of risks over time. Trending often involves sampling data from both internal and external sources to incorporate both micro and macro views of threats.

4. Visualization Layer

The final component of this system is the visualization layer. Its function is more than to serve as a GUI front end that displays canned reports. Instead, its role is to visually highlight "areas of interest" without knowing in advance what questions to ask. This is accomplished by showing different attributes from the processing engine as visual dimensions, so that unusual combinations stand out.

Applied to risk optimization, we look for unexpected pockets of residual risk. These may be unexpected because they deviate from a standard distribution of risk values, or may be unexpected clusters of Susceptibility, Exploitability and Impact. One should continue to focus on risk by Impact type to manage outcomes, but present these dimensions as overlays.

The visualization engine also allows for drill-down into specific events, so that common controls can be sought. For this to be useful, there must be filtering capabilities that exclude "uninteresting" events. The total number of vulnerabilities collected is often so large that one must hide most of the data to allow focused analysis.

Lastly, the visualization layer must offer reporting on benchmarks and trends as well as before/after analysis.

Summary

A holistic view of risk allows organizations to prioritize activities and optimize spend vs. residual risk. This view treats risk as a portfolio of potential outcomes (Impact types), and highlights areas of potential over or under-investment. It is based on the premise that we manage risk to manage the probability of an undesirable outcome, and answers the questions:

What is my total material risk of something bad happening?

What do those numbers mean (through benchmarks)?

How do I optimize my risk profile and/or dollars spent (where should I increase investment and where can I decrease)?

Are there unusual pockets of risk?

Are common controls available?

Are new areas of risk emerging or declining?

This view is created by implementing a framework that enforces best practices of risk data collection and management, combined with a scoring system and a sophisticated visualization engine.

For more information, please contact info@rev2.net.

Copyright © 2011, Rev2. All rights reserved. The RiskView® application is a registered trademark of Rev2. All other trademarks in this document are the properties of their respective owners.