Graphical Risk Analysis (GRA): A Methodology To Aid In Modeling Systems For Information Security Risk Analysis

Omar A. Herrera R., CISSP
(oherrera@dttmx.com)

Deloitte & Touche,
Jaime Balmes #11, Edif. C,
8° piso, C.P. 11510, Polanco,
México D.F., México

Abstract

Risk analysis for information security, as we know it today, is a difficult task involving experience and extensive knowledge of the environment being analyzed. There are already many methodologies out there but most fall in the category of what we call “check lists” or “questionnaires”.

I will present a methodology, “Graphical Risk Analysis” (GRA), that can aid in risk analysis activities for information security. The approach is based on well-known system security principles and uses diagrams to model the system at different abstract levels. The information security risk analyst can, with this approach, ensure that he/she understands the main business processes which the system environment supports, analyze in detail the critical parts of an information system that are most critical from a business perspective.

GRA intends to be a simple risk analysis methodology focused on availability and dependency of services and systems; although it is not intended to be an all inclusive solution, it will certainly is very useful, in conjunction with other methodologies, in all information security risk analysis activities.

1. Introduction

It is well known that there are many deficiencies with the current methodologies for risk analysis in information security, there are checklists and other types of methodologies out there, but most have one or more of the following limitations:

- They are technology specific (lack of portability; usually not useful if you have different hardware/software)
- Too extensive (difficult to manage; there is too much information to collect and you often spend many hours filtering irrelevant information)
- No single standard (too many variations; there are many ways to conduct an analysis and usually you can't compare different analyses even when they were made with the same methodology)
- Geared towards technology and not business processes (they tend only to discover security holes in hardware/software but forget about the business process and system design)
- Only one level of abstraction (you can't analyze with greater detail critical parts of the system)
- Incomplete (some methodologies fail to include many important aspects of risk analysis)

Risk analysis is not an easy task, many risk analyses fail because they don't provide useful information for decision makers. Every risk analysis has a purpose and providing the proper information to the correct people is essential to its success.

I present here a methodology to ease some of the issues mentioned.

2. Background

Before I present the details of the methodology, I would like to review some important concepts and assumptions so that the approach proposed in this paper is better understood.

2.1 History

I mentioned before that there are many methodologies available to support in risk analysis, some of them are implemented in software; this significantly improves visualization but it may not be enough. Most methodologies fall in one (or several) of the following categories:

- Checklists
- Questionnaires
- Probabilistic Analysis
- Weight/impact Analysis
- Vulnerability-Menace correlation
- Graphic Analysis

Also, there is another (more common) way to categorize risk analysis methodologies:

- Qualitative Risk Analysis - Tries to assess risk by personal and professional experience, assigning labels such as high, medium and low.
- Quantitative Risk Analysis - Tries to determine the value of risk by quantifying the risk with two variables: monetary loss and time (usually bye means of a probability).

Both types of methodologies have been very useful to assess risk and none of them is better than the other (each has its own advantages and disadvantages).

2.2 Common risk analysis techniques used today

When we talk about quantitative risk analysis, there are some formulas commonly used to estimate the risk and other parameters useful for risk management. A small compendium of these formulas/concepts and their uses are described next [T.Peltier01] [L.Krutz01].

Concept	Description	Formula
Exposure Factor **(EF)**	Percentage of loss a specific event would have on an asset. This value is hard to estimate, however, estimating this value based on the importance of the asset for business would be easier and more representative.	(see description)
Asset Value **(AV)**	Monetary value of the asset (as with EF, it is better to base this value on the asset's value to the business).	(see description)
Single Loss Expectancy **(SLE)**	Monetary loss derived from the occurrence of a specific event.	**SLE= AV * EF**

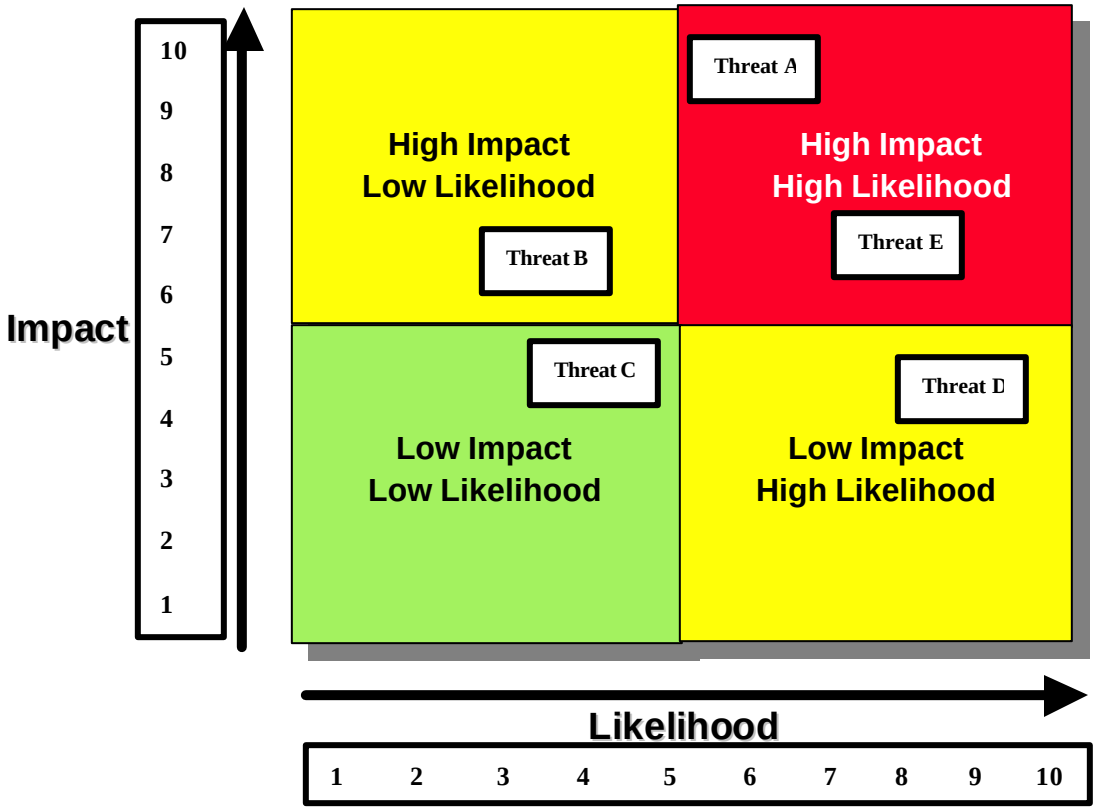
Annualized Rate of Occurrence **(ARO)**	Estimated frequency in which an event is expected to occur in a year.	(see description)
Annualized Loss Expectancy **(ALE)**	Annual expected financial loss caused by the occurrence of a specific event.	**ALE= SLE* ARO**

We can clearly see that we can't define with precision a value; everything is estimated in some way using the experience, knowledge and judgment of the people involved in the risk analysis. Still, quantitative risk analysis is one of the best tools at hand to justify information security investments and defining cost-benefit.

Another tool worth mentioning is the "Risk Matrix" (also known as "Threat vs. Likelihood Matrix") [T.Peltier01] [A.Summers98] [O'Farrell02]. The goal of this tool is to classify several threats in terms of the impact for business and the likelihood of that threat becoming real. It is important to look at the definition of some concepts to understand this better:

- Threat – something that might harm an asset
- Vulnerability – lack of or ineffective security control that makes an asset more vulnerable (might increase the impact and/or likelihood of a threat)
- Risk – in this context, risk can also be defined as: "the likelihood of a threat becoming real multiplied by the impact on business", where both likelihood and impact might increase because of some vulnerabilities that are present.

An example of a this type of matrix is shown below:



Qualitative Risk Analysis is also common; it relies heavily on the analyst experience and as such usually requires the presence of the enterprise's internal experts [T.Peltier01]. Common Qualitative Risk analysis include:

- Facilitated Risk Analysis Process (FRAP)

- Vulnerability Analysis
- Hazard Impact Analysis
- Threat Analysis
- Single-time loss algorithm

3. The GRA approach

GRA will give the risk analyst many tools to determine different elements of risk and has some useful features like the possibility to formally compare several risk analyses under certain conditions. Be advised, however, that GRA will not eliminate subjectivity from the analysis (you will still depend on experience, knowledge and good judgment to define certain elements) however, GRA is defined in such a way that it allows to match similarly performed analyses so that you can compare one with the other(s).

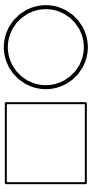
The key element of GRA is that the analyst must define elements and relationships (which we will call a system) rather than values. It is easier to define and identify the importance of assets in a diagram where the business flow is illustrated and the formulas defined in the GRA methodology are designed to quantify this importance in a clear and simple way.

This subjectivity is still present, in some way, in other quantitative methodologies but being a graphical methodology, GRA lets the analyst see the differences more clearly, allowing him/her to better define the system according to his/her needs.

An important fact about the GRA methodology is that it is information and process oriented. The relationships between elements (relationships are represented with accesses in GRA) are what will actually define risk indexes.

3.1 GRA graphical elements

We will start with the definitions of GRA elements in diagrams. Making a GRA diagram is the first step in this methodology and, as you will see, it is fairly easy to create and understand.

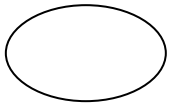


- **Resource (Circle)** – This is one of the basic elements of a System and could represent virtually anything (concrete or abstract); examples of resources are: a document, a computer, a Web page, a router and even a person (from a system's point of view).
- **Group (Box)** – This element actually refers to an undetermined number of the same type (this means resources with identical characteristics from the system's point of view); examples of groups are: computers, system's users (persons), web sites and files.

- **Access (black filled arrow)** – One of the two types of relationships present in GRA, an access specifies "a request for a service" between two resources, two groups or a resource and a group. We call the requesting element the "client" (from which the arrow will start) and the element to whom a service is requested (where the arrow points to). Examples of accesses are: any request to modify, obtain, create, delete or store information related with a server element (where the information might be stored or processed), any request to modify behavior or characteristics of a server element (functions like turning on and off, and setting certain modes of operation).

- **Pass-through Access (open point, grey arrow o dashed arrow)** – Pass-through access are basically the same as normal accesses (black arrows), except that are directed to pass-through elements; they are differentiated from normal accesses in order to keep track of dependency issues in GRA. The definition that best describes pass-through elements in this context is that of a "proxy"; "filters" are also good

definitions of pass-through elements, like firewalls. For example, a user in a Local Area Network (LAN) might need to request information from a web site (server) through a firewall (pass-through element), if there were any other means of requesting this information the firewall wouldn't be doing its job, so the user depends on the firewall to establish the connection with an external network (probably Internet) and obtain the information he/she wants. Of course, there has to be some normal access at some point and we define in GRA that all accesses between the server (final element) and the last pass-through element must be an "access" (black filled arrow) as defined in the GRA methodology. For the sake of clarity in black and white printed diagrams, pass-through accesses may be represented also with a black, open point, dashed arrow.



- **System (Oval)** – This element actually represents a set of elements by itself acting as a container; blank ovals make reference to systems whose composition is unknown to us. A system, by definition, is a set of elements (groups and resources) and its relationships (accesses).

3.1.1 Labeling and element dictionaries

Labels are an important part of a GRA diagram; labels identify the elements and help keep track of related elements (like a path of defined by several access arrows).

For standardization and clarity, it is recommended that labels are always defined below the elements that they refer to:

Label standard in GRA define labels for each type of element as following:

- For **systems**, **resources** and **groups**:
  - Labels will consist of capital letters {A..Z}
  - Should the number of letters in the alphabet be insufficient to label the elements in a GRA diagram, a label consisting of 2 or more capital letters will be created where: a) the labels with less character will precede those formed by more characters (example: "A" has precedence over "AA"); b) Label "A" will have the highest priority/precedence; c) with multi-character labels, precedence will be read from left to right, being the leftmost character the one with more weight in precedence (as with multiple digit numbers).
  - Labels should be preferably assigned in the order of business process or flow of information from the business point of view, where "A" corresponds to the first element (according to business).
- For **accesses** and **pass-through accesses**:
  - Labels will consist of lower case letters {a..z}
  - Should the number of letters in the alphabet be insufficient to label the elements in a GRA diagram, a label consisting of 2 or more lower case letters will be created where: a) the labels with less character will precede those formed by more characters (example: "a" has precedence over "aa"); b) Label "a" will have the highest priority/precedence; c) with multi character labels, precedence will be read from left to right, being the leftmost character the one with more weight in precedence (as with multiple digit numbers).
  - Labels should be preferably assigned in the order of business process or flow of information from the business point of view, where "a" corresponds to the first element in a business process flow.
  - Related accesses and pass-through accesses will have the same label with a number (indicating the order to reach the server element) at its right side in parenthesis. Consecutive numbers will be assigned beginning with "1", setting the highest number in the label of the arrow representing the last access in the path (the one pointing to the server element). Some examples: a(1), a(2), a(3), be(3), c(5).
  - Redundant paths will follow the rules stated above and will include as well a sub-indexed (subscript) number after the lowercase letter(s) that define the label; this sub-index will indicate the consecutive number that refers to a particular redundant path and will be a constant through all accesses in the path. All accesses participating in redundant paths must contain sub-index and the first redundant path will use sub-index 1. For example, we might

have to redundant paths labeled as: $a_1(1)$, $a_1(2)$, $a_1(3)$, $a_1(4)$ (first path); $a_2(1)$, $a_2(2)$, $a_2(3)$ (second path).

With The intent of maintaining clarity in GRA diagrams, definitions are actually kept on a separate table called the element dictionary; describing the elements on the same diagram might be easier when diagrams are simple, but could be confusing with complex diagrams.

Element dictionaries are defined as a 3 column table with as many rows as there are elements within a System, where:

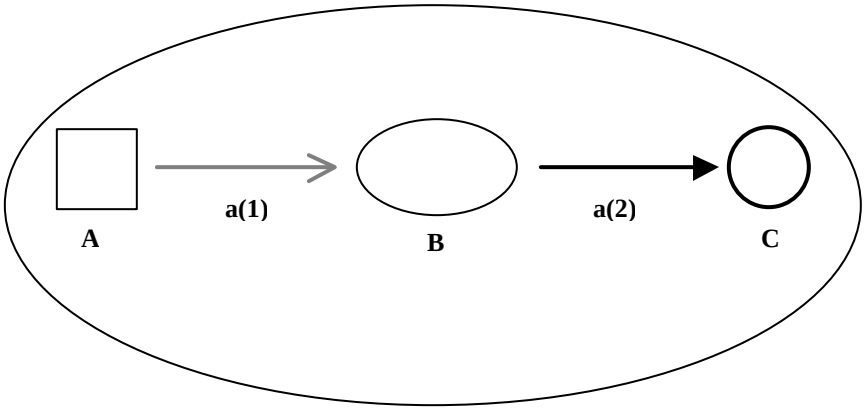
- The first column will contain the label of the element as shown in the diagram
- The second column will contain a short description of the element (characteristics and commonly used labels that describe the elements)
- The third column will optionally contain a detailed description of the element (mainly to clarify the function of the element in the system and to include important notes related with the element).

Note that "the system" being analyzed is not necessarily included as part of an element in the table, but you can provide a description and detailed description for it in your documentation though a label which will be associated with it (you will still have labels for other systems within your system).

3.1.2 Diagram Examples

This being said, let us look at some diagram examples using the GRA methodology:

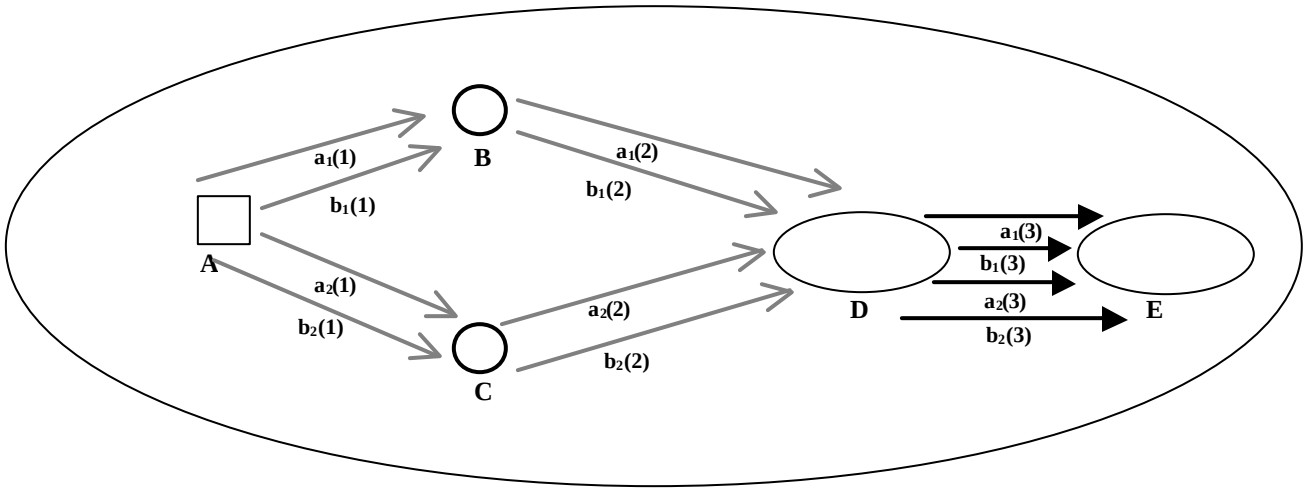
Example 1: "web page service":



Element Dictionary for system "web page service":

Label	Description	Detailed Description / Notes
A	"Users"	All personnel with Internet connection in the organization
B	"The Internet"	
C	"Web Server"	
a(1)	Pass-through access to the "Web Server"	In order to request a web page in this system, a user has to connect to a Web server through the Internet
a(2)	Access to the "Web Server" to request web page on the Internet	

Example 2: “common internet access configuration from LAN”



Element Dictionary for system “common internet access configuration from LAN”:

Label	Description	Detailed Description / Notes
A	“Users”	All personnel with Internet connection in the organization
B	“E1 line to internet”	
C	“ADSL line to the internet”	
D	“Internet Service Provider”	Provider of access to the Internet (the same provider for the E1 Line and the ADSL line)
E	“The Internet”	
$a_1(1)$	Pass-through access, to the “Web Service” through “E1 line” (path 1)	
$a_1(2)$	Pass-through access, to the “Web Service” through ISP (path 1)	
$a_1(3)$	Access to the web Service (path 1)	Path 1 is normal access for users
$a_2(1)$	Pass-through access, to the “Web Service” through “ADSL line” (path 2)	
$a_2(2)$	Pass-through access, to the “Web Service” through ISP (path 2)	
$a_2(3)$	Access to the web Service (path 2)	Path 2 is alternate access for users
$b_1(1)$	Pass-through access, to the “email Service” through “E1 line” (path 1)	
$b_1(2)$	Pass-through access, to the “email Service” through ISP (path 1)	
$b_1(3)$	Access to the email Service (path 1)	Path 1 is normal access for users
$b_2(1)$	Pass-through access, to the “email Service” through “ADSL line” (path 2)	
$b_2(2)$	Pass-through access, to the “email Service” through ISP (path 2)	
$b_2(3)$	Access to the email Service (path 2)	Path 2 is alternate access for users

3.2 Risk analyses in GRA, risk indexes and formulas

We will see now how to estimate risk related to a system, through the use of formulas and risk indexes in GRA.

Note that strictness in formality has been sacrificed for the sake of clarity. The goal of general risk is to identify risk levels by looking at the specific elements for each risk level.

3.2.1 General risk

General risk (**GR**) is determined by the principle of "risk by service" (see chapter 4, "Principles in the GRA methodology") and it is based on the number of accesses directed to certain element (acting the element as a server element).

- We define the general element risk (**GER**) as follows: $GER_E = |A_E|$, where **A** represents the set of accesses pointed to the element **E** and **| A |** represents the cardinality of this set (in other words, the number of accesses); in this context means any type of access (pass-through and normal accesses).
- We define the general system risk (**GSR**) as follows: $GSR = \sum_{i=0}^{i=n} GER_i$ (the sum of all GERs of elements in the system being analyzed), where **n** is the number of elements in the system being analyzed.

3.2.1.1 Risk indexes

There are some considerations while calculating risks for the elements of a system. There exist 3 types of risk levels which are called indexes in the GRA methodology, each of which refers to a certain type of element; just as we don't mix apples with oranges (or sum ordinary numbers with imaginary numbers), risk level indexes are different for resources, groups and systems and you can' t make arithmetic operations with all of them (at least not directly):

- **Resource index** – this is the ordinary and basic index and is not denoted by any particular notation since groups and systems are composed by sets of resources (resources are a basic elements).
- **g index (groups)** – This index reflects the risk level of a group and is denoted by the number representing the risk level, followed by letter "**g**" (lowercase g).
- **S index (systems)** - This index reflects the risk level of a system and is denoted by the number representing the risk level, followed by letter "**s**" (lowercase s).

Indexes are mainly useful for system (the system being analyzed) risk analysis where different type of risk level values are computed and expressed in a risk equation (Like GSR).

Groups are basically systems that contain any number of identical elements (same characteristics) and would be called homogeneous systems as well; Systems on the other hand, may contain any number and type of elements as well as relationships.

3.2.1.2 Example of GER and GSR calculation

Using the diagram example 1 ("web page service"), we will calculate GER and GRS as follows:

GER of element A (**EA**): $GER_{EA} = |A_{EA}| = 0g = 0$ (A is a group)

GER of element B (**EB**): $GER_{EB} = |A_{EB}| = 1s$ (since B is a system)

GER of element A (**EC**): $GER_{EC} = |A_{EC}| = 1$ (because C is just a resource)

Then, GSR for the whole system being analyzed is: $GSR = \sum_{i=0}^{i=n} GER_i = 1+1s$ (and not = 2 !)

See chapter 5 for abstraction levels and comparing results of different analyses, there you will see that to get rid of all "g" and "s" indexes and compute a single resource value you will have to go to other abstraction levels. In this case, it is useless because it would be extremely complex to identify all relevant elements and then you might just end up with some other "s" and "g" values.

Who could say that he/she has determined the risk level of the Internet? What is stated above is that, for this particular system that has been analyzed, whatever the risk level of the Internet (GER_{EB}) is (from this systems' point of view), it will be multiplied by a factor of 1.

3.2.2 Dependency Risk

This risk level indicates the dependency factor of a certain element from another element(s); this type of risk is based on the "principle of risk by dependencies" (See Chapter 4 for more information). In other words, this risk level for a certain element will indicate the impact it has on another

- The element dependency risk (**EDR**) for a certain element (**E**) is defined as follows: $EDR_E = |B_E|$, where **B** represents the set of pass-through accesses pointed to the element **E** and **|B|** represents the cardinality of this set.
- We define the system dependency risk (**SDR**) as follows: $SDR = \sum_{i=0}^{i=n} EDR_i$ (the sum of all EDRs of elements in the system being analyzed), where **n** is the number of elements in the system being analyzed.

It is important to note that risk indexes also apply in dependency risk calculation (see 3.2.1.1 for more information about risk indexes).

3.2.2.1 Example of EDR and SDR calculation

Using the diagram example 1 ("web page service"), we will calculate EDR and SDR as follows:

EDR of element A (**EA**): $EDR_{EA} = |B_{EA}| = 0g = 0$ (A is a group and no pass-through access is directed to it)

EDR of element B (**EB**): $EDR_{EB} = |B_{EB}| = 1s$ (since B is a system and a pass-through access points to it)

EDR of element A (**EC**): $EDR_{EC} = |B_{EC}| = 0$ (because C is actually the server element)

Then, SDR for the whole system being analyzed is: $SDR = \sum_{i=0}^{i=n} EDR_i = 1s$

See chapter 5 for abstraction levels and comparing results of different analyses, there you will see that to get rid of all "g" and "s" indexes and compute a single resource value you will have to go to other abstraction levels. In this case, it is useless because it would be extremely complex to identify all relevant elements and then even, you might just end up with some other "s" and "g" values.

As with the example of general risk, the results also make sense. Who could say that he/she has determined the dependency risk level of the Internet? What is stated above is that, for this particular system that has been analyzed, whatever the dependency risk level of the Internet (EDR_{EB}) is (from this system's point of view), it will be multiplied by a factor of 1.

4. Principles in the GRA Methodology

1. **Principle of risk by service**: Each access implies a risk (both in impact and likelihood), though we can't determine with precision the likelihood of the risk materializing or the impact, we know for sure that both variables will have a value higher than 0, thus, the Risk Level (**R**) will also be higher than 0. The more accesses directed to an element the higher the risk level is, so, the principle dictates that for two identical resources, one element (**E1**) with more accesses (offering more services) will certain have a higher risk level (**R**) than other identical element (**E2**) with lesser accesses: **R(E1) > R(E2)**. In the context of GRA, we will assume that this same principle applies for any given set of elements (be they

identical in characteristics or not).

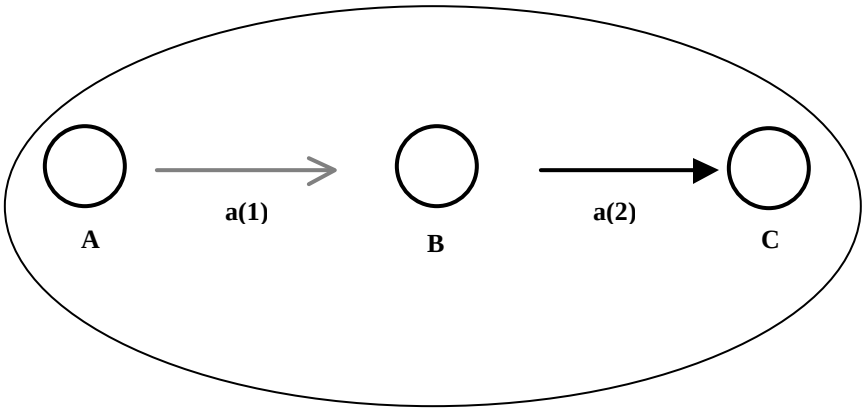
2. **Principle of risk by dependencies**: Each element contains inherent risks and imply a certain risk level for any process or function where they are involved; the more elements participate in a certain process, the higher level of risk. Since the probability of having a certain level of risk is the same for all unknown elements it is fair to assume that any element in a given process will increase the risk in the same proportion as the other elements participating in the same process. Derived from the last statements, a dependency risk will be defined as the number of pass-through accesses directed to the element (number of pass-through accesses in which the element participates). This type of risk is mainly related with the "availability" security requirement.
3. **Principle of business**: "Security supports the business goals, rather than being a goal itself (generally); in order to maintain focus on business all GRA diagrams should reflect business critical processes and flows of information". The problem with other methodologies (like checklists) is that they tend to focus more on technical issues ignoring the business process.
4. **Principle of aggregation**: Since all groups and systems are also composed by elements whose risk could be calculated, the risk level associated with this element types ("s" and "g") can be decomposed as following: Let **U** be an element of a type of group or system whose risk level is represented by the variable **X,** X_U can be calculated in terms of other risk levels by multiplying it with the system risk resulting from its own analysis (GSR or SDR).
5. **Principle of homogeneous groups**: A group of elements is considered homogenous if there are no relationships between the elements in the group (any type of access between elements), and so, the only relationships (accesses) are defined from each element in the group to the outside, exactly in the same way as the accesses are defined for the group in a higher level of abstraction; if this holds true, then the risk level (**R**) from the group (**R**g; meaning risk **R** with g index) will be equal to a resource risk level (**L**) where **L=R*n**, where n is the number of elements in the group and the group.

5. Abstraction Levels and comparison of analyses

Abstraction levels simplify the job of analyzing complex system. Basically, all systems defined within "the system" being analyzed can be decomposed and analyzed separately; the world is composed by systems within systems within systems within systems.

However, we cannot endlessly decompose or aggregate systems or otherwise we will loose the focus of our analysis. To avoid this we should always apply the "principle of business" (see chapter 4 for more detail).

After decomposing groups and systems, we can substitute "s" and "g" types of risk levels and obtain resource levels of risk, in this way, we approximate the real risk of the system and its element. The "principle of aggregation" defines how this relation works. So, if we want to decompose an equation like: $GSR = \sum_{i=0}^{i=n} GER_i = 1+1s$ as in example 1, we should analyze the GSR risk in element B in example (system); for the sake of simplicity, let us assume that "The internet" the element B in example 1 is composed by a few resources defined as follows:



Element Dictionary for system "The internet" (weird simplification):

Label	Description	Detailed Description / Notes
A	"Perimeter router of LAN"	All LAN connections are routed through this device
B	"Backbone router"	
C	"perimeter router of destination server's network"	
a(1)	Pass-through access to the "router C"	All traffic from C to A must be routed through a backbone router B
a(2)	Access to the "router C"	

Now, with some quick calculations, we determine that the GSR of the Internet ($GSR_{Internet}$) is: $GSR_{Internet} = \sum_{i=0}^{i=n} GER_{Internet_i} = 1+1 = 2$ then the GSR for the actual system we are analyzing is: $GSR = \sum_{i=0}^{i=n} GER_i = 1+1s = 1+1*2 = 3$ (since the system risk "s" for the Internet, in our example was worth a 2 resource risk level).

Note also that in this case, the Internet was seen as a group rather than a system, because it contained the same type of resources in it (routers), but for our example this won't matter.

6. General guidelines for risk analysis with GRA

As the principle of reduction states, you cannot further analyze resource elements, if you consider them complex enough and worthy of analysis from a business point of view, define those elements as groups or systems and leave resources as your baseline (without a baseline you might get lost in different abstraction levels). Define as few systems and groups as necessary; for example, you might treat computers as resources or computers as systems with resources like CPU, software, input devices and output devices, but most organizations won't require that level of detail (maybe some governmental agencies though).

To start, consider including some users to define the business process in diagrams (either as resources or as groups), and always create the diagram from a business point of view; you will be surprised many times by discovering that risks are not in a computer related element and that to mitigate risks you will need to implement a control other than a firewall, for instance.

The GRA methodology allows the risk analyst to make relatively faster, business oriented risk analysis with less subjectivity involved in the process. It will complement other analysis (both quantitative and qualitative) with useful information. GRA also might be better suited for executive presentations of results by illustrating the business process while showing clearly dependency points.

7. References

Printed References:

[L.Krutz01] **"The CISSP Prep Guide: Mastering the Ten Domains of Computer Security"**, Ronald L. Krutz, Russell Dean Vines and Edward M. Stroz, published by John Wiley & Sons.

[O'Farrell02] "**Hack Proofing your Wireless Network**", Neal O'Farrell, Christian Barnes, Tony Bautts, Donald Lloyd, Eric Ouellet, Jeffrey Posluns and David M. Zendzian, published by Syngress.

[T.Peltier01] "**Information Security Risk Analysis**", Thomas R. Peltier, published by Auerbach.

[M.Piattini01] "**Auditoría Informática, un Enfoque Práctico**", Mario G. Piattini, Emilio del Peso, 2nd Edition, Published by Alfaomega.

Electronic References:

[Globalcontinuity] "**A glossary of business continuity and business risk management terms**", http://www.globalcontinuity.com/static/glossary/glossary.html

[A.Summers98] "**Techniques for Assigning A Target Safety Integrity Level**", Angela E. Summers, http://www.iceweb.com.au/sis/target_sis.htm

[F.Cohen97] "**Risk Management or Risk Analysis?",** Fred Cohen, http://www.all.net

[Dittrich00] "**Estimating the cost of damages due to a security incident – FAQ**", http://staff.washington.edu/dittrich/