Business risks of software vulnerabilities

Four sources of risk are relevant for evaluating the influence of software vulnerabilities on businesses

by Hilbrand Kramer MSc (Royal Holloway, 2013)
and Colin Walter, ISG, Royal Holloway

The influence of software vulnerabilities on business risks

Four sources of risk are relevant for evaluating the influence of software vulnerabilities on businesses

by Hilbrand Kramer MSc (Royal Holloway, 2013)
and Colin Walter, ISG, Royal Holloway

Why is it necessary to understand the business risk of vulnerabilities? Within the field of information security, we perform many audits and assessments, discovering all kinds of vulnerabilities. Vulnerabilities are often stated in rather technical language, and so the manager may not be able to recognise the problems. It also appears that discovered vulnerabilities are usually rated as high, which means that the organisation needs to invest its scarce resources to mitigate all vulnerabilities.

We need some measure to distinguish between vulnerabilities. Business risk is a commonly used measure to triage unwanted events within an organisation.

Software vulnerabilities

Today the security of software is a topic that has the attention of organisations and regulators. Managers ask for a penetration test to provide evidence about the security of their applications. If the resulting report states a couple of vulnerabilities that are rated as “high risk”, the manager asks: “What does it really mean for me? Do I have a severe issue?” This situation is encountered in day-to-day practice.

Software vulnerabilities are stated in rather technical language, and the manager may not be able to recognise the problem. Even as an information risk manager, it is often rather difficult to present the business risk. Vulnerabilities that are seen as a significant problem to the penetration testers can be difficult or impossible to relate to business (process) risks.

This article presents the results of an investigation of literature and the development of a method that management can use to understand and prioritise software vulnerabilities.

Before we are able to judge how severe a vulnerability is, we must closely inspect the environment in which the vulnerability is found. We must:

- Understand where the system belongs to: what the greater context is.
- Understand what the system does: what the elements and functions are.
- Understand how the system acts: the data and control flow of the (sub)system.

Probability of a threat

From the definition of business risk, we know that before we can explain the risk to a manager, we need to know the vulnerability that is likely to occur and need to know what the impact of this occurrence is, using the risk language that he will understand.

Glossary

A business risk is a threat event occurring that exploits a vulnerability causing an impact on the organisation’s operation, assets or individuals.

A software vulnerability is a weakness in the code of an information system that could be exploited by a threat source.

A threat is any circumstance or event with the potential to adversely impact organisational operations and assets, individuals, or other organisations.

The likelihood of occurrence of the threat that will exploit the vulnerability can be assessed using the following model:

Natural threat agent	Fire, flood, power failures, and other.
Unintentional threat agents	Parties that cause damage or loss of service without direct intent.
Intentional threat agents	Those parties that would knowingly seek to make a threat manifest
Capability	Capability to conduct and sustain an attack or totally destroy the system and any replacement.
Threat catalyst	Those factors or actions that cause an attack to be initiated at the time and on the target selected.
Threat agent motivators	Factors and influences that motivate a threat agent.
Threat inhibitors	Any factor that decreases either the likelihood of an attack taking place or being successful.
Threat amplifiers	Any factor that increases either the likelihood of an attack taking place or being successful

In day-to-day practice, most reports will only describe some of the threat agents that exist (if any). Some reports present threats to the level of access, but the level of inhibitors and amplifiers are rarely mentioned. To be able to include these factors, one must understand the environment of the vulnerability – the threat combination.

Source of risk model

From a broad range of methods and frameworks that address the relationship between IT risk and business risk/objectives, concepts linking IT risks to risks relevant for management were assessed.

This led to four sources of risks or impacts that are relevant for evaluating software:

- Operational risk related to business processes.
- Regulatory risk effects of non-compliance with the applicable laws and regulations.
- Reputational risk related to the brand of the organisation.
- IT generic risk stemming from the non-compliance of software with technical standards that leads to reduction of the overall information security.

Figure 2 (page 4) provides an overview of the above mentioned different contexts that will affect the evaluation of software security.

A model to link software vulnerabilities to business risks

Based on a review of the literature and practical knowledge, a model was developed to link software vulnerabilities to business risk. It consists of four parts:

- The **environment** in which the software vulnerability is discovered should be thoroughly understood.
- A **threat score model** is developed to assess the likelihood of a threat exploiting the software vulnerability.

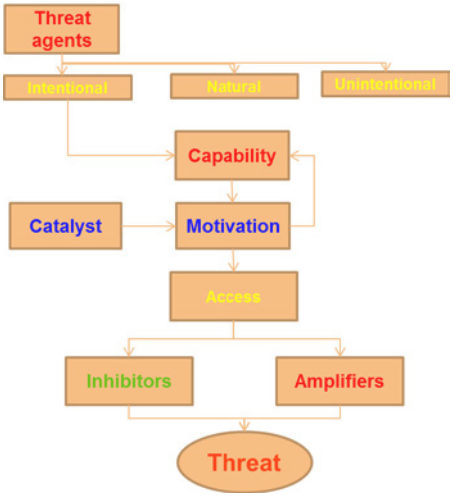


Figure 1: Threat components and their relation

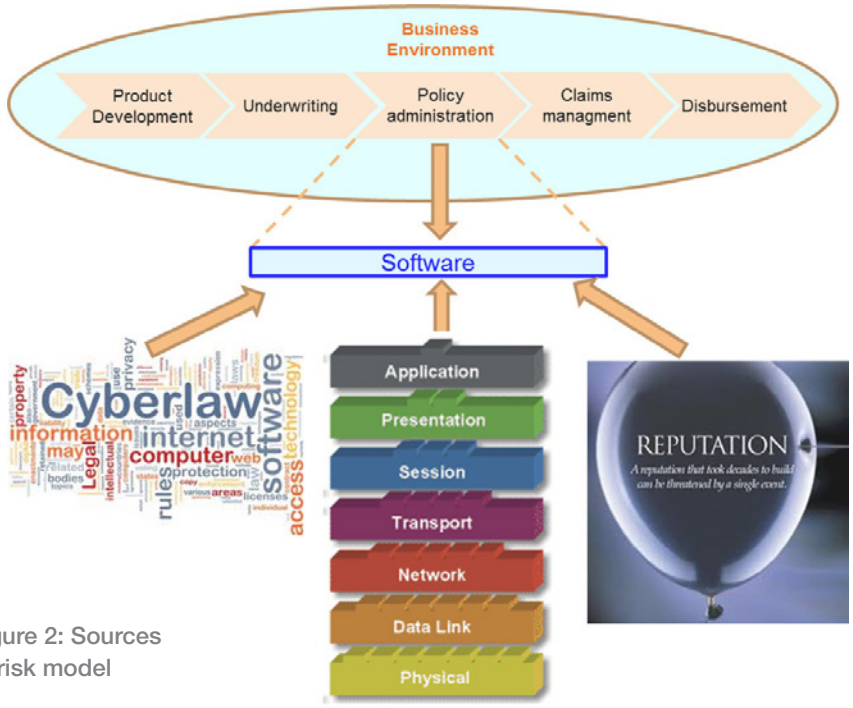


Figure 2: Sources of risk model

- The **impact of an exploited vulnerability** must be assessed. This will be performed with the information risk analysis methodology (IRAM) of the information security forum (ISF) to relate software vulnerabilities to the first three sources of risk. COBIT5 is selected to relate software risk to the fourth of the above sources of risk: the generic IT risk.

- The **resulting risk** for the broad business environment should be established and plotted in a 5x5 matrix. This will reveal the business risk and can be used during discussions to achieve mutual understanding about the risk.

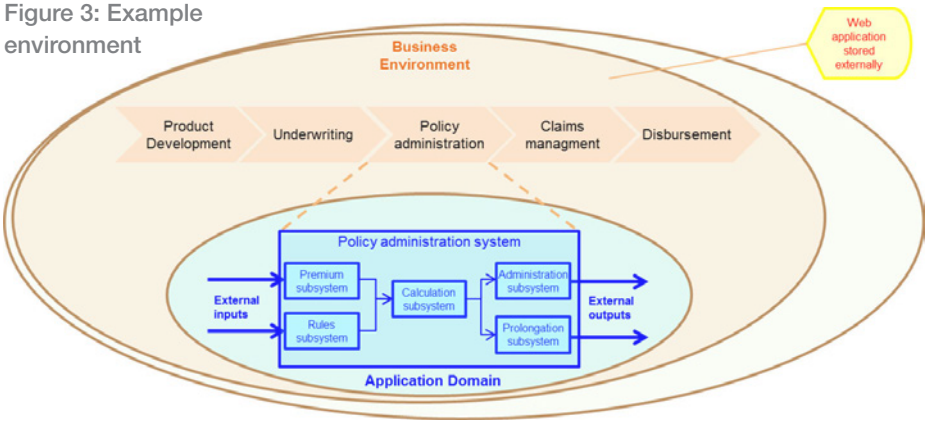
To illustrate this model, a case study of four hypothetical software vulnerabilities within a hypothetical environment is presented next and the software vulnerabilities are thereby related to business risks.

1. Environment

The environment of a hypothetical insurance organisation is illustrated in Figure 3 (below). The following components are shown:

- **A wider environment** that, for example, will impose legislation that will force the organisation to change the retirement age from 65 to 67.

Figure 3: Example environment



The case study presents four kinds of hypothetical software vulnerabilities, discovered during a software penetration test:

- The software of the premium subsystem includes a bug that opens a back door to enable an attacker to change the beneficiary of the payment of a pension.
- An externally-hosted website includes a software flaw and makes it vulnerable for injection of pictures that compromise the organisation.
- Due to a software bug, the SQL statement to retrieve customer privacy data can be misused to get a dump of all of the customer data within the system.
- The software is developed in such a way that it requires a particular port within the firewall to be opened in order to function properly.

- **A business environment** that influences how software risks are perceived. For example, within an insurance organisation, people tend to be driven to ensure that there is no risk. Also, cultural goals such as “we do what we promise”.

- **A process with goals** that will execute a certain activity for the organisation to contribute to the organisational goals. For example, the insurance process will process an inquiry of a customer for a pension policy to ultimately yield the payment of the pension.

- **An application system** that will support a (sub)process to attain its goals. In the example below, the policy administration system has the same scope as the process. The goals for the system will be equal to the process.

- **Application subsystems** that will perform particular tasks within the whole application domain. For example, the goal of a calculation subsystem might be to calculate the premium of a customer accurately and on time.

2. Threat score model

For the four hypothetical software vulnerabilities, the proposed model below is filled in. With the results of the model, the assessor and the business manager can discuss whether they agree with the estimates made and thus with the overall threat level.

Based on outcomes of the examples in the model, we can draw some conclusions. Changes in the circumstances can have a huge impact, such as in

With the results of the model, the assessor and the business manager can discuss whether they agree with the estimates made

Threat score model		Vulnerability 1		Vulnerability 2		Vulnerability 3		Vulnerability 4	
	Possible rating	The software of the premium subsystem includes a bug that opens a backdoor to enable an attacker to change the beneficiary of the payment of a pension.	Rating	Externally hosted website includes a software flaw and makes it vulnerable for injection of pictures that compromise the organization.	Rating	Due to a software bug the SQL statement to retrieve customer privacy data can be misused to get a dump of all the customer data within the other system.	Rating	The software is developed in such a way that it needs some port within the firewall to be opened to function properly.	Rating
Natural threat agent	0-4	N/A		N/A		N/A		N/A	
Unintentional threat agents	0-4	N/A		N/A		N/A		N/A	
Intentional threat agents	0-4	Disaffected employee.	4	Hackers or pressure groups.	3	Hackers or unauthorised staff.	3	Hackers.	4
A = highest of three agents			4		3		3		4
Capability	0-4	Staff with no development knowledge is not able to understand what needs to be performed. Developer has no access to the production environement.	0	Software, technology, facilities, education and training, methods and books and manuals are easy to retrieve.	4	Hacker have the knowlegde to perform the attack. Normal users will be less knowledgable.	2	Hackers have the technology and the knowledge to perform port scans to search for vulnerabilities.	4
B = score agent +			4		7		5		8
Threat catalyst	0-4	Developer get access rights for production environment to solve a production issue.	4	Dispute about a policy hold with the insurance company owning the website.	2	N/A	0	Technology change can effect that a port that was e.g. open for an application is now widely open.	2
C = B + Catalyst			8		9		5		10
Threat agent motivators	0-4	Personal gain.	4	Try to get even.	3	Power for the hacker. Curiosity for the normal user.	3	Power to be able to hack the environment.	3
Access: D = C +			12		12		8		13
Threat inhibitors	0-16	Deveoper will lose his job. Change will be discovered in payment system	10	Fear to get captured by police force.	4	Viewing of information is hardly monitored nowadays.	0	The use of the ports can be monitored and the hacker can be noticed.	2
E = D - inhibitors			2		8		8		11
Threat amplifiers	0-16	N/A	0	Target vulnerabilities, script available to execute it.	2	Access to information is a strong amplifier.	2	A server within a big Insurance firm could be a target to be proud of.	2
Overall rating threat			2		10		10		13

Figure 4: Case study: Threat score model

the first example. Previously, the vulnerability could not be exploited because the developer was not able to access production systems. The threat catalyst (that the developer gains access to the production environment to solve a production problem) changes that situation, and now he is able to exploit the vulnerability.

Although this vulnerability appears very severe, the threat inhibitor that the change of the beneficiary will be discovered in the payment system makes it useless to exploit. This, however, requires that one has knowledge of the broad environment of the software to be able to assess the threat on its real merits.

3. Impact of an exploited vulnerability

Based on the results of the threat assessment, only the impact of the threats of examples 2, 3 and 4 needs to be assessed. However, for illustration purposes, we will assume that Threat 1 will also be likely to occur.

Bug in the software of the premium subsystem

Definition of the example vulnerability: The software of the premium subsystem includes a bug that opens a back door to enable an attacker to change the beneficiary of the payment of a pension.

The impact of this vulnerability is related to the business process, so we need to relate the discovered vulnerability back to the impact on the business process. At this point, we need to use information about how the software acts, what the system does, and understand the environment of the system. We will use the environment shown in Figure 5 (below).

Information about how the software acts should normally be collected during the software security assessment phase. The data flow diagram in which the flow of information is drawn can provide knowledge about the impact of the vulnerability on how the software acts.

Information about how the software acts should normally be collected during the software security assessment phase

Changes in beneficiaries are entered by staff of the policy administration department via an input screen of the beneficiary subsystem. The access to this functionality is restricted to several employees. The changes will only be validated if the changes are approved by another staff member (4-eyes check). However, by including a backdoor in the software of the beneficiary, a fraudulent user is able to retrieve a shell and enter beneficiary data without the proper authorisation and without the 4-eyes check.

It is necessary to know what the system does in order to know why it is important that the piece of software that contains the vulnerability works in the

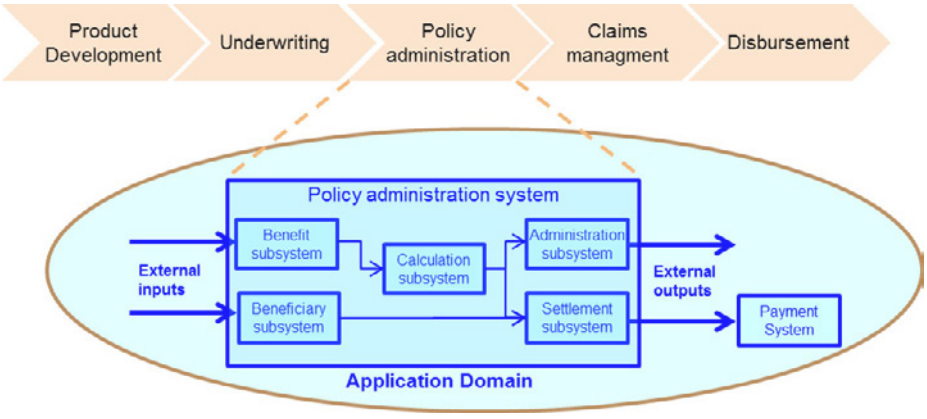


Figure 5: Relation process and application

intended manner. What are the relations between the subsystems? What are the input and output relations?

In our system, the settlement subsystem uses the information of the calculation subsystem and the beneficiary subsystem as input. When the input from the beneficiary subsystem is incorrect, the settlement can then be drafted for the wrong beneficiary.

"To which environment does the system belong?" is the latest important question at system level. The system can receive input from other systems and provide output to other systems. It is also possible that the system will be supervised by a monitoring system. This can involve monitoring of the process flow through the system or monitoring that is more security-related (for example, determining that there are no direct changes at the database level).

The policy administration system has a user interface for, for example, input of beneficiary information. Settlements will be transferred to the payment system, which will distribute the pensions to the customers.

It is important to know the business context relevant for the discovered bug in order to gain insight into the business impact of the vulnerability. It is also important to know which process (part) the system supports. The process will have control objectives that need to be supported by the software if portions of the process are automated.

In the hypothetical environment, the policy administration system supports the policy administration process. One of the controls defined for the process is that a beneficiary may only be changed by an authorised person and must be approved by a second authorised person.

A business manager of the process to which the control objective is related is able to assess the impact of an event that occurs. This assessment should be performed without taking into account the measures that have already been completed in this process (step) or in another process (step). Ultimately, the question would be what the impact is when the vulnerability materialises.

When a business impact assessment (BIA) is performed for the application of the process, a link with this assessment can be made to rate the impact of a vulnerability. When a fraudulent person can change the beneficiary, the impact should be assessed. This can, for example, be related to the maximum amount of a settlement or a settlement run (for example, the business-related fraudulent payments have a "high" impact rating – see Figure 6: relevant part of the BIA).

It is important to know the business context relevant for the discovered bug in order to gain insight into the business impact of the vulnerability

User guide

Business Impact Rating

Integrity

Ref.	Business impact type *Business impact of errors in information or of deliberate manipulation of information to perpetrate or conceal fraud (most serious case)*	Business impact rating A-Very high, B-High, C-Medium, D-Low, E-Very low					Explanatory comments
		A	B	C	D	E	
Financial							
F1	Loss of sales, orders or contracts	20% +	11% to 20%	6% to 10%	1% to 5%	Less than 1%	
F2	Loss of tangible assets (eg fraud, theft of money, lost interest)	$20m+	$1m to $20m X	$100K to $1m	$10K to $100K	Less than $10K	If someone is able to change all the beneficiaries of a pension settlement run to him as being the beneficiary.

Figure 6: Case study: IRAM, F1 and F2 part of BIA sheet

User guide

Business Impact Rating

Integrity

Ref.	Business impact type *Business impact of errors in information or of deliberate manipulation of information to perpetrate or conceal fraud (most serious case)*	Business impact rating A-Very high, B-High, C-Medium, D-Low, E-Very low					Explanatory comments
		A	B	C	D	E	
Customer-related							
C4	Damage to reputation (eg confidential financial information published in media)	World-wide negative publicity	Continent-wide negative publicity	Nation-wide negative publicity	Local negative publicity	Minor negative publicity	When web page will present adverse information this will cause broad negative publicity
			X				

Figure 7: Case study – IRAM, C4 part of BIA sheet

Software flaw in externally-hosted website

Definition of the example vulnerability: Externally hosted website includes a software flaw that makes it vulnerable for injection of images that compromise the organisation.

All applications that an organisation uses, internally or externally hosted, should be rated for their impact on the company's objectives. The relevant risk that the organisation can face in this situation is reputation risk. This type of risk does not have a direct link to the business processes and should be assessed at an organisational level.

The board members of the hypothetical organisation stated in the company's mission that they are against war and do not want to be involved with parties selling arms. Pressure groups who oppose war associate the company with insuring companies that sell arms. They hacked the website and used it as their publishing platform. The damage to the company's reputation is rated as in Figure 7 (relevant part of BIA).

Software bug in SQL statement

Definition of the example vulnerability: Due to a software bug, the SQL statement to retrieve customer privacy data within a policy administration system can be misused to accomplish a dump of all the customer data within the customer relation system where the request is sent to.

This vulnerability can be related to the business process, but can also be viewed as a separate regulatory risk category. Besides, the organisation can face reputation risk. This type of risk does not have a direct link to the business processes and should be assessed at an organisational level.

Regulatory bodies have established regulations for the use and storage of privacy information and (rather vague) requirements for automated systems. Legislation states that a business should take appropriate technical and organisational measures against unauthorised or unlawful processing of personal data, and against accidental loss or destruction of, or damage to, personal data.

The board members of the hypothetical organisation stated in the company's mission that they will support their customers, regardless of the situation. This company-wide philosophy also has an effect on the way the company will handle the data of its customers.

The board members have rated the impact of a breach of compliance with Data Privacy Legislation and Reputation Damage as "high" (see Figure 8, page 9).

However, before an impact can be assessed, it is first necessary to have a clear understanding of precisely which privacy data will be disclosed. It is also

All applications that an organisation uses, internally or externally hosted, should be rated for their impact on the company's objectives

User guide

Business Impact Rating

Confidentiality

Ref.	Business impact type *Business impact of unintended or unauthorised disclosure of information (most serious case)*	Business impact rating A-Very high, B-High, C-Medium, D-Low, E-Very low					Explanatory comments
		A	B	C	D	E	
Operational							
O4	Breach of operating standards (eg contravention of regulatory standards)	Closure of building or operation	Serious sanctions imposed	Significant sanctions imposed	Moderate sanctions imposed	Minor sanctions imposed	A breach that will disclose large number of customer data will result in serious fines of the regulator.
			X				
Customer-related							
C4	Damage to reputation (eg confidential financial information published in media)	World-wide negative publicity	Continent-wide negative publicity	Nation-wide negative publicity	Local negative publicity	Minor negative publicity	A breach that will disclose large number of customer data will result in broadly negative publicity.
			X				

Figure 8: Case study – [16] IRAM, O4 and C4 part of Excel sheet

necessary to establish which information, when combined, constitutes privacy-related data.

How a software flaw affects network exposure.

Definition of the vulnerability: The software is developed in such a way that it requires some port within the firewall to be opened in order to function properly.

The attack surface of the entire security infrastructure can be negatively impacted by a software design flaw. Software can be designed in such a way that it requires certain ports on servers to be opened in order to function properly. This can introduce vulnerabilities on the generic infrastructure level that will also affect other components (including “crown jewel” applications).

This vulnerability is related to the technical context of the software. The developed software has to comply with the technological standards of the organisation. The information technology infrastructure is designed as a layered environment with certain controls on different layers that, as a whole, ensure the security of the environment. COBIT5 may be used for these processes. For example:

- APO01.08: Maintain compliance with policies and procedures. Establish procedures to maintain compliance with, and performance measurement of, policies and other enablers of the control framework, and enforce the consequences of non-compliance or inadequate performance. Track trends and performance, and consider these in the future design and improvement of the control framework. One of the activities is: analyse non-compliance and take appropriate action.

- DSS05.02: Manage network and connectivity security. Use security measures and related management procedures to protect information over all methods of connectivity.

The application is tested in the test environment using dummy interfaces. During installation on the production environment, the implementing team recognises that certain required input does not reach the application. The developers inspect the production system and tell the implementers that a certain port on the firewall needs to be open.

The people who implemented the infrastructure for the application are requested to open the port. They, however, inform the implementers that the installed firewall is operating according to the operational security guidelines and do not want to open the port unless someone accepts the risk. The application owner

The attack surface of the entire security infrastructure can be negatively impacted by a software design flaw

Vulnerability	Threat level	Impact	Risk
Bug in software of the premium subsystem	2	High	Low
Software flaw in externally hosted website	10	High	High
Software bug in SQL statement	10	High	High
Software flaw effecting exposure of network	9	Medium	Low

Figure 9: Case study – Risk evaluation

is the only party willing to accept this risk. He has already paid millions of euros for the application and wants it to work as soon as possible.

The conclusion is that the application owner should not be able to accept this risk, even when he wants to accept it. He has a strong motivation not to follow the standard and lacks knowledge about the overall security environment. When he accepts this risk, he will only be attentive to the component that he understands, which is his business application environment.

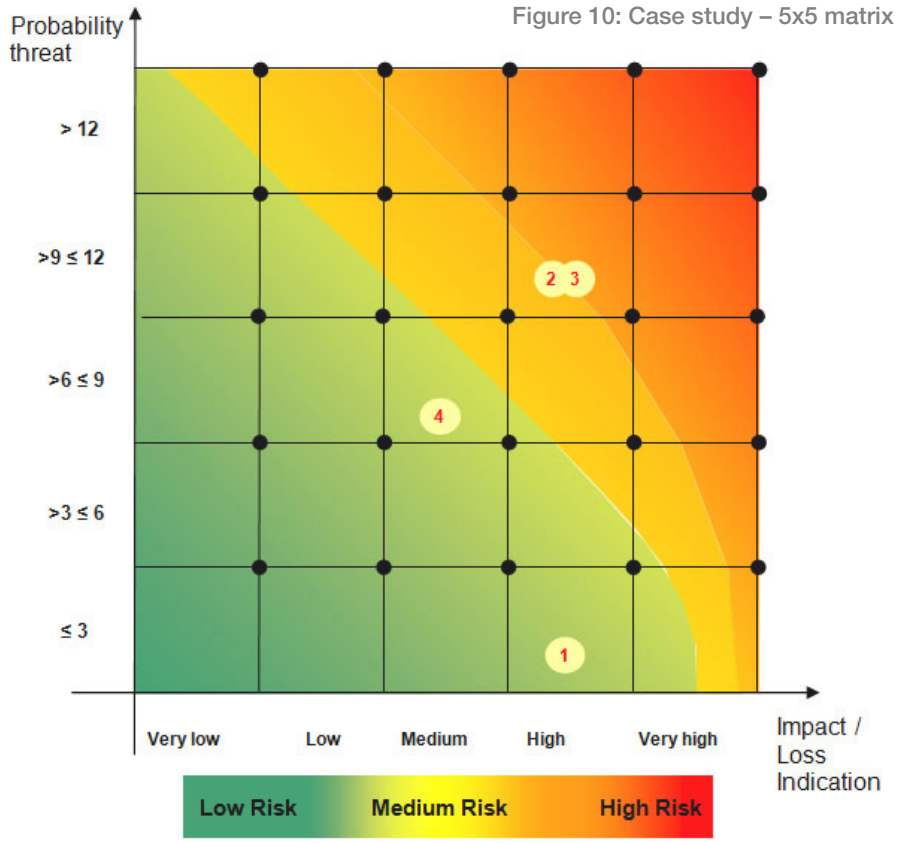
The impact of this kind of vulnerability should not be assessed by someone from the business, but by someone with the technical security knowledge of the comprehensive environment. Information for this kind of assessment cannot be gathered from a business impact assessment. It may require a technique such as fault tree analysis to deduce the “impact” on the overall security architecture.

For the example, we assume that the impact of the vulnerability is rated as “medium”.

4. Resulting risk

The risk of the example software vulnerabilities can be assessed using the model, the probability of the threat occurring, and the impact of the vulnerability.

Figure 10: Case study – 5x5 matrix



Based on the risk model, the assessed risk has been completed in the risk column of the table (Figure 9, page 10).

Based on this model, we can now show the business manager the effect of the software vulnerabilities on the risk profile of the organization.

Conclusion

Software vulnerabilities will be an issue in the coming decades since much software is developed without taking security into account. Besides this, as millions of lines of codes are written, the occurrence of faults having security exposure is inevitable. Knowing this, the need for some method to distinguish between software vulnerabilities based on the level of business risk will become more and more desirable. Scarce resources can only be invested once.

Security assessments and audits of software have to be extended. Nowadays assessment stops in general when vulnerabilities are presented in a rather technical language with some severity rating. Software vulnerabilities have to be presented in a language that clarifies which vulnerabilities can have the highest negative impact on business goals. This enables a business manager to make an informed decision about which vulnerabilities have to be mitigated first. ■

About the authors

Hilbrand Kramer is working as information risk manager at insurance company Nationale Nederlanden. Formerly he worked as information risk manager at ING within the financial markets environment and as auditor at ING and external audit firms. He acquired an MSc in information security at the Royal Holloway University of London in 2014 and is registered as certified EDP-auditor (RE).

Colin Walter recently retired from the ISG at Royal Holloway, where he was director of the MSc in information security by distance learning. Formerly he worked for certificate authority Comodo and the University of Manchester Institute of Technology (UMIST). He has written a number of scientific research articles, including many on improving the efficiency and side channel security of the algorithms used in hardware for public key cryptography.