RISK ASSESSMENT AND MITIGATION AT THE INFORMATION TECHNOLOGY COMPANIES

Ben Marx*, Deon Oosthuizen**

*Department of Accountancy, University of Johannesburg, South Africa
Tel: 011-5593156
Fax: 011-5592777
**FirstRand Group Internal Audit, South Africa

Abstract

Developing computer software that is free from material defects is the ultimate goal for software developers; however, due to the cost and complexity of software development, it is a goal that is unlikely to be achieved. As a consequence of the inevitable defects that manifest within computer software, the task of software patch management becomes a key focus area for software companies, IT departments, and even end users. Audit departments, as part of their responsibilities, are required to provide assurance on the patching process and therefore need to understand the various decision-making factors. Software flaws that exist within computer systems may put confidential information at risk and may also compromise the availability of such systems. The study investigated the recommended approaches for the task of software patching, with a view to balancing the sometimes conflicting requirements of security and system availability. The study found that there are a number of key aspects that are required to ensure a successful patching process and that the internal auditors of the 'big four' South African banks considered most of these factors to be important.

Keywords: Software Patches, Software Patch Management, Software Flaws, Risk Assessment, Risk Mitigation, Confidentiality, Integrity, Availability, Downtime, Information Security

1. INTRODUCTION

There has been a rapid increase in the prevalence of computers and related devices in South Africa (Statistics South Africa, 2012:71). However, it is the greater connectivity between computers brought about by the Internet in the 1990s that really changed the dynamics of how society operates today. This is evidenced by the Internet penetration in South Africa being measured at 35.2% in 2011 and increasing at a steady pace, most notably though mobile phone connectivity (Statistics South Africa, 2012:72).

The greater prevalence of computers, allied to the increase in Internet penetration in the last 15 years, has resulted in a far greater amount of digitally stored information being available to the world's population. As organisations began to build a web presence in order to both connect with and market to their ever-increasing Internet-based target consumer base, they quickly learned that the threat to Internet-connected systems is very serious. Sommerville (2011:367) notes that "as more and more systems were connected to the Internet, a variety of different external attacks were devised to threaten these systems."

The effect that these trends have had on the discipline of software engineering has been immense. Prior to the advent of the Internet, when computers were primarily standalone, the only manner in which to exploit a software loophole was to have physical access to the computer. Since the inception of interconnected computers over a wide area network such as the Internet, it is now possible to connect to a computer from the other side of the world in order to exploit a software flaw. This has made it imperative for computer software to be designed in a manner that it is more resistant to malicious attack and hacking, and introduced a new challenge for software engineers to design and implement systems that are secure to address these risks (Sommerville, 2011:367). The Kindsight (2014:3) malware report further highlight these risks by identifying the following trends with respect to virus or malware infections:

- Mobile device malware infections are accelerating, with an increase of 14% noted for the first half of 2014 alone. It is estimated that approximately 15 million devices worldwide may be infected with some form of virus or malware;
- Spyware (software designed to steal personal information) is also on the increase, especially in the mobile phone environment; and
- With respect to computers connected to home networks, it is estimated that approximately 18% of homes may have devices that are infected with malware as at June 2014.

From the above it is evident that the importance of understanding IT risks and the process of software patch management is critical for auditors in assessing these risks and formulating their audit approaches, whether as internal or external audit.

2. OBJECTIVE, SCOPE AND LIMITATIONS

The objective of the paper is to investigate the considerations relevant to software patching with a view to identifying the most important aspects required to implement a successful software patching process and the impact thereof on auditors. The methodology followed for the empirical study consists of the analysis of a questionnaire sent to South Africa's largest banking institutions.

The surveys were sent to the internal audit departments at the big four South African banks. Given the high level of risk maturity in the South African banking environment, these findings may not necessarily represent other industries in South Africa or abroad. The major South African banks were selected as the population for the survey due to the importance of both security and availability considerations to their businesses.

Due to the research population being limited to the financial services (banking) sector, it is probable that patching requirements and approaches may be different for other types of organisations. This is especially true where there are vastly differing levels of risk maturity and risk tolerance across different industries.

3. THEORETICAL BACKGROUND

3.1. Risk management

3.1.1. Risk based auditing

The concept of risk-based auditing is introduced by Griffiths (2005:1), who explains that "risk-based audit is probably the most exciting and significant development in the internal audit profession's history. It has the potential to catapult the reputation of and the value added by this profession into the stratosphere." He further expands on the concept to explain what it means when an audit is conducted using a risk-based approach: "The simplest way to think about risk-based audit conceptually is to audit the things that really matter to your organisation. Which are the issues that really matter? Probably those areas that pose the greatest risks" (Griffiths, 2005:5). The concept of risk-based auditing highlights the importance for auditors to assess potential audit risk areas and make recommendations based on the principle of risk.

Due to the fact that patch management is a discipline that emanates from the development of software, it is necessary to consider the impact of software risks. Pressman (2010:745) gives his view on software risk as follows: although there has been considerable debate about the proper definition for software risk, there is general agreement that risk always involves two characteristics: uncertainty that the risk may or may not happen and loss if the risk becomes a reality.

Software patches are implemented to address either a software flaw that results in unexpected or unwanted behaviour or security vulnerability in the software that could compromise the integrity of the software and allow unauthorised access. As a result, security is one of the key drivers behind the need to deploy software patches. Sommerville (2011:369) emphasises the importance of assessing possible losses that might ensue from attacks on assets in systems, and balancing these losses against the costs of security procedures that may reduce these losses. This indicates that risk management is an integral part of any software development initiative, where the developers are constantly balancing the costs of additional development time against the risk of software flaws. The approach of balancing the cost of losses against the cost of additional controls is therefore relevant to any software developer. The costs involved to develop a completely secure software package may be exorbitant, and doing so could possibly be so costly that it may be unaffordable to the end user. This is a reason why software companies may accept that certain security and functionality issues will always be present in the software they create. The trade-off is that fixing these issues after the software is released may be far more economical than attempting to find all problems prior to the software's release. This is the concept that underpins the practice of software patch management, as it is the reason why patching exists.

The process of risk assessment is an on-going rather than a one-time event, and as such should be considered throughout the lifetime of the system. Thus patch management is one of the key aspects to consider after an application has been delivered to the business.

3.1.2. Risk assessment

Subsequent to the identification of risks, the next step is that of determining how to go about assessing these risks. Rainer, Snyder and Carr (1991:133) note that there are many methodologies currently in use that attempt to measure the loss exposure of assets. These methodologies can be broadly categorised as either quantitative or qualitative.

Most quantitative methods are based on loss exposure as a function of the vulnerability of an asset to a threat multiplied by the probability of the threat becoming a reality (Rainer et al, 1991:133). The most basic and universal approach to assessing risk, as noted by Collier (2009:85), is to use an impact/likelihood matrix. This process is also commonly called risk mapping. The likelihood or probability of occurrence may in the most simplistic form be categorised as high, medium or low. Similarly, impact or consequences in terms of downside risk (threats) or upside risk (opportunities) may also be categorised as high, medium or low.

Qualitative risk analysis methodologies which use risk factors that are not numeric in nature may save time, effort, and expense over quantitative methodologies because IT assets need not have exact monetary values, nor do threats need to have exact probabilities. Furthermore, qualitative methodologies may be useful in identifying gross weaknesses in a risk management portfolio, but may often be imprecise as the variables used (i.e. low, medium, and high) are not always clearly understood by all parties involved in the risk analysis process.

3.1.3. Risk appetite and tolerance

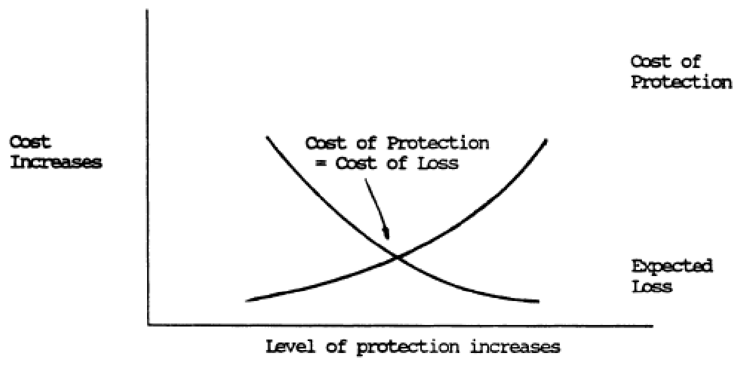
The Information Systems Audit and Control Association (hereafter ISACA) and Collier (2009:69) provide similar definitions for both risk appetite and risk tolerance. Risk appetite is seen as the amount of risk a company or other entity is willing to accept in pursuit of its objectives, while risk tolerance is the acceptable variation relative to the achievement of an objective (ISACA, 2009:17). In contrast to risk appetite, risk tolerance is defined as the tolerable deviation from the level set by the risk appetite and business objectives. In other words, this is the acceptable deviation that the organisation will accept, which, for example, could be in the form of overruns of 10% of budget or 20% of time, etc. (ISACA, 2009:17).

It is clear that risk tolerance for software patching will differ from organisation to organisation depending on the extent and significance of IT infrastructure implemented. For purposes of consistency in comparison, this study will focus on the risk posture of large banks in South Africa with regard to software patching.

3.1.4. Risk treatment

Risk treatment has at its core the process of selecting and implementing measures to modify or reduce risk. These can include, among others, risk control/mitigation, risk avoidance, risk transfer and risk financing (e.g. hedging, insurance). Risk treatment, sometimes also called risk response, involves decisions as to whether particular risks should be avoided, reduced, transferred or accepted (Collier, 2009:89). Hopkin (2010:245) also identifies four strategies for addressing risk, referred to as the "Four T's": namely, tolerate the risk, treat the risk, transfer the risk (insurance) and terminate the activity giving rise to the risk. While the terms may be slightly different, the objectives are broadly the same. The figure below by Rainer, et.al (1991:132) illustrate the relationship between protection vs. expected loss.

Figure 1. Cost of protection vs. expected loss



Source: Rainer et al (1991:132)

When considering whether to implement any software patch, it will be necessary to understand the impact and likelihood of the weakness. For instance, it may not be necessary to implement a software patch that has a low impact and likelihood due to the cost involved as well as the risk of downtime.

3.2. Software defects

3.2.1. Defects in the software development process

Harris (2013:1085) notes that programming code is complex and costly and highlights the need for programmers and application architects to strike a balance between the functionality of the program and ensuring that security requirements are implemented. With respect to software defect removal, Jones (2010:555) notes that there are two distinct processes, the first being development defect removal (when defects are found and removed during software development) and the second being maintenance defect removal (when defects are corrected after the development of the software). It is also noted that the major cost driver for the total cost of ownership (hereafter TCO) of software is that of defect removal (both development and maintenance defect removal). It is claimed that between 30 and 50 percent of every dollar ever spent on software has gone to finding and fixing bugs. Being a significant percentage, it is clear that this area of spending will be placed under huge pressure when software development organisations are seeking to reduce costs.

3.2.2. Software testing

When conducting software testing, the intention is to show that a program does what it is intended to do and to discover any program defects before it is put into use. Normally, programmers carry out some testing of the code they have developed during the programming process. This often reveals defects that must be removed from the program. This is commonly called software debugging. Defect testing and debugging are seen as different processes. While testing establishes the existence of defects, debugging is generally concerned with locating and correcting these defects (Sommerville, 2011:41).

3.2.3. The problem with testing

Software can be extremely complex and as functionality and features increase, so does the level of complexity. There are two main reasons noted in the literature as to why not all the errors in program code are discovered and rectified during testing. The first reason given by Dooley (2011:194) is humans not being perfect. Dooley questions that "if we made mistakes when we wrote the code, why should we assume we won't make some mistakes when we read it or try to test and fix it?" While this problem can happen for even small programs, it may be particularly prevalent for larger programs that have upwards of 50,000 lines of code. This is a significant amount of code to review, and as a result, the likelihood of missing a problem is high. The second reason why problems are missed during testing is that, due to their complex nature, programming errors can escape from one testing phase to another and ultimately reach the user. Even small programs have many pathways through the code and many different types of data errors that can occur (Dooley, 2011:194). This is also borne out by the large number of possible programming errors that can manifest themselves, as indicated in Section 3.2 of this study.

3.2.4. Software patches

As has been noted earlier, the software engineering process is not perfect and many different issues of varying severity could still manifest themselves in the final product that is released to the customer. This is where the development of patches by software developers begins. Meyer and Lambert (2007:1) provides a description of what a patch is: "No software program is perfect. As problems and bugs are discovered, the developer or a third party may fix them. This fix is what is referred to as a software patch."

While critical software patches typically need to be deployed within a short timeframe, it is nevertheless important for organisations to follow a formal process to deploy these patches – this process should aim to ensure that adequate testing has been performed before deployment (Taylor, Allen, Hyatt & Kim, 2005:18). Patching may be a risky operation for a number of reasons such as the fact that patches tend to affect many critical systems libraries and other software used by numerous applications. Patches can also often be significant changes, many times with little documentation describing what they change. Patches also tend to be large and complex operations with even small configuration variances that can cause drastically different results. These factors can make the success rate for patch changes much lower than other changes, thus requiring more comprehensive testing (Taylor et al, 2005:18).

3.3. Software patch management

The literature generally accepted that any effective patching process should follow a number of predetermined steps. Various authors provide a differing number of steps to be executed, but the overall approaches share much similarity. The recommendations will be reviewed and compared in order to determine the best practice requirements for an effective patching process.

Sun Microsystems (2004:6) provides five practices to be considered for any patching strategy. These include: analysing the need to apply patches or update software based on risk, cost, availability and timing; minimising change to the IT environment whenever possible; addressing alert notifications and other critical issues as soon as possible; only making other changes to the IT environment to address known problems; and maintaining the IT environment as currently as is appropriate for the business and application needs. As per these recommended practices, it is evident that there are likely two conflicting factors: firstly, the need to keep the software environment updated, and secondly, to minimise changes to the environment as far as possible. This highlights the complexity in deploying patches in an effective manner.

As the need for a robust patching approach has been established, the focus is now on the required steps to ensure such a process is fully effective. Trent (2004:7) notes that there may be four main steps involved in deploying software patches, namely: assess, identify, evaluate, and plan and deploy. It is noted that these steps may be repeatedly executed as part of the patching process. The IIA (2005:6) suggest that internal auditors should keep up to date on leading IT change and patch management processes and recommend that the organisation adopts these processes. The details within these phases will now be further investigated by assessing the approach taken by various authors.

3.4. Auditing patch management

Auditors, both internal and external, should be aware of how their companies and clients are managing the patch process as this is a key control in securing a company's data, including financial data. The recent AICPA Statements of Auditing Standards (SAS 104, Due Professional Care in the Performance of Work, and SAS 109, Understanding the Entity and Its Environment and Assessing the Risks of Material Misstatement) as well as the COSO Integrated Framework for Enterprise Risk Management have also increased the emphasis on the responsibility of auditors for assessing risk.

Aside from working toward best practices, organisations should fully engage accountants and auditors in the patch management process, as they may play an important role in ensuring that this key internal control is effective (Meyer & Lambert, 2007:6). It is noted to be especially important that accountants within an organisation, in corporate accounting and internal auditing, as well as external auditors, take an active role in the patch management system. There should also be at least one representative from corporate accounting and internal auditing in the patch management group. Furthermore, external auditors should provide their expertise in designing and evaluating controls to help the patch management group improve the patch management system. Patch management should also not be seen as purely an IT problem, but rather as a key internal control protecting the financial information of a company. (Meyer & Lambert, 2007:6).

4. METHODOLOGY

The literature study provided the foundation for the aspects that were tested empirically through the questionnaires sent to the Internal Audit departments of the big four South African banks.

4.1. Population

Banks are by their nature highly susceptible to fraudulent activity, and they also require a high level of system availability, as customers expect 24/7 availability. With these two factors in mind, the banking industry is an ideal research environment for assessing the approach used to deploy software patches. There is often a large disconnect between deploying patches quickly to avoid security vulnerabilities being exploited and ensuring minimum system downtime resulting either from deploying the patch or from outages relating to insufficient testing of a patch. The Banking Association of South Africa (2012:3) identifies four major banks in South Africa, and it is noted that these four banks represent about 84% of total banking assets. These banks are Standard Bank, which is said to be the largest in terms of assets, with 31%, followed by ABSA (26%), FirstRand (20%) and Nedbank (23%). PwC (2014:30) also name the four largest banks in South Africa as Barclays Africa Group (Previously ABSA), FirstRand, Nedbank and Standard Bank. This is further confirmed by Wikipedia (2015). The survey relating to the software patching practices within their respective organisations was sent to the head of internal audit for each of the four big South African banks. These participants were selected for their thorough knowledge of business practices at their respective organisations.

4.2. Questionnaire design and testing

The questions in the questionnaire were based on the information obtained from the literature study and other internal audit practitioners. The questionnaire was designed to ensure that participants could easily complete the questions. All the questions also provided the opportunity for the participant to provide his/her comment if desired. The participant was able to complete the questionnaire either electronically or manually. Before the questionnaire was sent out it, was tested by a selected group of people consisting of academics and audit practitioners. Testing the questionnaire ensured that the questions were unambiguous and set out logically and that the questionnaire was easy to complete. It was also determined that it would take on average no more than five minutes to complete the questionnaire.

5. RESEARCH FINDINGS AND INTERPRETATION

The objective and findings of each question in the questionnaire will be explained and discussed below:

5.1. Software patching risks

5.1.1. Objective of the question

Two major patching risks are identified in the literature. The first is related to confidentiality or security, which is the risk that manifests when software is compromised by a hacker and confidential information is leaked, as typically occurs when customer or credit card information is compromised. The second risk is that of availability: this risk could manifest either through an attack on software that renders it intentionally unavailable by a hacker, or through a software patch being deployed with insufficient testing such that there are unexpected errors in the software code which cause the software to be unavailable. Question 1 was aimed at identifying which of the two significant patching risks was deemed to be of a higher significance to the big four banks.

5.1.2. Findings

From the responses received, it is evident that the internal auditors of the major South African banks deem the security risk to be more significant in general, as stated by three of the four respondents. While certain South African banks have experienced downtime that has caused customer frustration, it is conceivable that customers would be even more irate and the likelihood of litigation far higher if there were to be a security breach that resulted in customer information being compromised.

Table 1. Software patching risks

Which of the following risks resulting from the process of software patching do you deem more significant?	***Number***	***%***
Potential security breaches which may result from a software vulnerability.	3	75%
Unexpected downtime as a result of the patching process, due to a patch that breaks functionality or causes systems to be unavailable.	1	25%

Source: Questionnaire (own calculation)

5.2. Deployment of software patches

5.2.1. Objective of the question

From the literature, a number of patching considerations are mentioned that can impact the decision-making processes relating to software patching. Question 2 was aimed at identifying the patching considerations that are deemed to be most important to the auditors within the big four banks when they assess the adequacy of a software patching process.

5.2.3. Findings

Various literature sources suggest a number of patching factors. The first is whether management signoff on the risk exposure is required in the event that a patch cannot be deployed or is not available. The vast majority of respondents indicated that this would be an important concern when assessing the adequacy of a software patching process. The second factor is the need to consider any possible unintended consequences when critical patches are rapidly deployed – the most noted consequence is unexpected downtime. Here too, the majority of respondents indicated that it is an important factor to assess from an audit perspective. The third factor is whether, if possible, patches are packaged, tested and then released as part of a formal release cycle. This process is aimed at reducing unexpected consequences of releasing patches outside of formal releases. All of the respondents indicated this to be an important consideration in the assessment of the patching process. The fourth factor is whether sufficient testing is performed to provide a level of confidence for successful patching on production systems. On this factor, all of the respondents indicated that it is a major consideration in their assessments. The fifth factor is whether consideration is given to whether a threat can be mitigated without the need to apply a software patch, as in certain cases it may be possible to employ a compensating control such as a firewall to mitigate a particular exposure. The responses were mixed on this factor, with two of the four respondents indicating that it is an important consideration, one respondent indicating it as a minor consideration and another as not at all important. The final factor is whether any threat posed by software vulnerability is considered against the ability to provide reliable services (i.e. avoiding downtime). Regarding this factor, all of the respondents indicated it to be of a major consideration in their assessments.

Table 2. Deployment of software patches

Which of the following factors do you deem to be important in assessing the need to deploy a software patch:	*Total*	*Number*			*Percentage*		
		To what extent			*To what extent*		
		Large	*Lesser*	*Not at all*	*Large*	*Lesser*	*Not at all*
Where a patch cannot be deployed or is not available, will relevant stakeholders and IT management be asked to sign off on the risk?	4	3	1	0	75%	25%	0%
For critical patches that need to be deployed as a matter of urgency, are unintended consequences considered?	4	3	1	0	75%	25%	0%
Consideration is given to the next release cycle and where possible, patches are packaged and tested with other updates.	4	4	0	0	100%	0%	0%
Sufficient testing is performed to ensure confidence and predictability for patches deployed to production systems.	4	4	0	0	100%	0%	0%
Consideration is given to whether the threat can be mitigated without applying the patch or update.	4	2	1	1	50%	25%	25%
The materiality of the threat is considered in terms of the ability to deliver safe and reliable service to the business.	4	4	0	0	100%	0%	0%

Source: Questionnaire (own calculation)

5.3. Software patching risk focus

5.3.1. Objective of the question

In the literature study, with respect to risk management, it was noted that the assessment of risk should form the basis of any decision regarding the patching of software. Question 3 seeks to explore whether the auditors of the big four banks believe that the patching process within their organisations are suitably risk focussed.

5.3.2. Findings

All the responses indicated that the internal auditors at the four major South African banks believed that the software patching process within their organisations could be improved to be more risk focussed. This is supported by the following comment received:

"There is a high priority set on 'doing patching', rather than performing a risk assessment and then patching."

Table 3. Software patching risk focus

	Total	*Number*			*Percentage*		
		To what extent			*To what extent*		
		Large	*Lesser*	*None*	*Large*	*Lesser*	*None*
Do you believe that the software patching process within your organisation is suitably risk focussed?	4	0	4	0	0%	100%	0%

Source: Questionnaire (own calculation)

5.4. Software patching programme

5.4.1. Objective of the question

In the literature a number of prerequisites for any patching programme are suggested. Question 4 seeks to identify which of these factors the auditors of the big four banks deem to be most important when developing a software patching programme.

5.4.2. Findings

The literature study identified four important factors that likely contribute significantly to the success of any patching process. The internal audit respondents surveyed were not all of the view that these factors were important to assess during an audit. This may indicate potential gaps in the auditing of the patching process. The majority of respondents were in agreement that the level of security knowledge of IT staff and the IT infrastructure allowing for automation are important considerations that would yield a successful outcome for the patching process. The level of user awareness was not seen to be as important in general. The amount of resources required to perform the patching operation was seen to be of lesser importance by the majority of respondents.

5.5. Testing of software patches

5.5.1. Objective of the question

In the literature it is noted that there are two distinct classifications for software patches based on the results of a formal risk assessment. These two classifications are 'normal' or routine patches and 'emergency' patches. It was noted that emergency patches should typically be deployed within a shorter timeframe than that of a routine patch. Question 5 seeks to identify the approach followed by the big four banks with regard to the testing of these two categories of software patches.

Table 4. Software patching programme

Which of the following factors do you deem to be important when assessing a software patching programme:	*Total*	*Number*			*Percentage*		
		To what extent			*To what extent*		
		Large	*Lesser*	*Not at all*	*Large*	*Lesser*	*Not at all*
The level of security knowledge of the IT staff who will be performing the patching process	4	3	1	0	75%	25%	0%
The amount of resources required to ensure that the task of patching can be effectively executed	4	1	3	0	25%	75%	0%
The level of end user knowledge and awareness for software patching	4	2	1	1	50%	25%	25%
IT infrastructure is suitably configured to allow an automated patching process to take place	4	3	1	0	75%	25%	0%

Source: Questionnaire (own calculation)

5.5.2. Findings

In the survey all the respondents indicated that despite the need to deploy emergency patches in a faster timeframe, the requirement for formalised testing is not reduced for these emergency-type patches. This clearly underlies the importance in the banking sector of ensuring that there is limited risk for unexpected results or downtime from the patching operation for all types of patches.

Table 5. Testing of software patches

Based on your audit assessments, are patches tested prior to deployment in a formal testing environment within your organisation?	*Number*	*Percentag e*
Only for emergency (critical) patches	0	0%
Only for normal scheduled patches	0	0%
For both emergency and normal scheduled patches	4	100%
Other	0	0%

Source: Questionnaire (own calculation)

5.6. Patch deployment timeframes

5.6.1. Objective of the question

The results from the literature study clearly indicate the need to perform a thorough risk assessment in order to determine the importance of deploying any particular software patch, as well as the urgency with which the patch should be deployed. Question 6 seeks to identify the method(s) used by the big four banks to determine the timeframe within which a particular patch should be deployed.

5.6.2. Findings

In the literature, the process of undertaking a risk assessment involves the assessment of two key factors, namely the impact of a particular risk and the likelihood of this risk materialising. Assessing these two factors for any given software patch can prove difficult for an organisation. Based on the responses received, a hybrid approach is used for all the big four South African banks, whereby the vendor assessment, as well as an internal risk assessment, forms the bases of risk rank and determine the timeframes for patch deployment. This indicates a level of risk maturity higher than what may be seen at other organisations and is likely indicative of the importance that the South African banks place on the discipline of risk management.

Table 6. Patch deployment timeframes

Based on your audit assessments, which of the following does your organisation use to determine the target timeframes for deploying patches?	*Number*	*Percentage*
Using vendor patching recommendations, such as severity ratings	0	0%
Through a risk assessment conducted internally	0	0%
Combination of vendor recommendations and an internal risk assessment	4	100%

Source: Questionnaire (own calculation)

5.7. Summative Findings on the empirical study

The empirical findings indicated that the process of patch management is complex and that there are a number of requirements and considerations that need to be taken into account when building an effective patching process. The recommended approaches provided by the literature would form the basis for the assessments that internal auditors would need to consider when evaluating their organisation's patching process.

The findings indicated that the big four South African banks generally followed a risk-based approach to the assessment of software patching and that most of the recommendations as set out in the literature are seen as important for the internal auditors during their assessments. It was however noted that there may still be scope for an improved risk focus relating to the process of software patch management. Furthermore, the importance of end users in the patching process was not generally regarded as being important which is contrary to the recommendations contained in the literature. This could indicate a potential gap in the audit approach to the assessment of the software patching process.

6. RECOMMENDATIONS AND AREAS FOR FUTURE RESEARCH

The survey results indicated that effectively managing the process of software patch deployment is a complex task which involves a number of factors which need to be evaluated in order to increase the likelihood of a successful outcome. The trade-off of increased security through rapid patch deployment needs to be balanced with the time required to effectively test a patch for system stability. This trade-off is likely to vary depending on the risk rating of the particular application and the risk appetite of the organisation concerned. A thorough risk management process is therefore advocated based on the findings of the study.

The following areas have been identified where further research may prove useful:

• An analysis of the extent of testing required before the deployment of software patches, based on their criticality rating. This would investigate how to strike a balance between performing sufficient testing and still ensure that a critical patch is deployed quickly.

• Expansion of the survey population to include various industries. Due to the varying levels of risk maturity and risk tolerance across different industries, it is likely that patching requirements and approaches may be vastly different.

• An analysis on the costs versus benefits of undertaking in-house risk assessment of software patches, especially for smaller organisations where there may not already be established risk management capability.

7. CONCLUSION

The study investigated the need for and the recommended approach to the deployment of software patches. It was found that risk management should play an important role in the assessment of any software patch prior to its possible deployment within a production environment. While software vendors may provide a risk rating with each patch released, it is also important for organisations to perform their own assessment of each patch, as their usage profile or configuration may result in a risk rating different to that of the software vendor. Furthermore, there are a number of requirements suggested in the literature for ensuring a successful patching programme. It is important for auditors to be aware of these suggestions during their audits of the software patching process within their organisations.

The empirical study found that within the big four South African banks, auditors were generally in agreement with most of the suggestions made in the literature, with the exception of the role of the end user in the patching process and the need to ascertain which resources are required for deployment of patches. All the respondents also indicated that within their respective organisations, the approach to software patching could benefit from an enhanced risk management focus.

The risk posed by software flaws shows no sign of abating in the near future. As a result, organisations will be required to continually deploy software patches in response to these flaws. A successful patching process is one that is able to patch the vulnerability in the shortest possible timeframe while preventing unnecessary downtime due to an insufficiently tested patch. To achieve this balance, any successful patching process must be suitably risk focussed.

REFERENCES

1. Banking Association of South Africa. (2012). *South African Banking Sector Overview*. The Banking Association South Africa. Johannesburg, South Africa.
2. Collier, P.M. (2009). *Fundamentals of Risk Management for Accountants and Managers*. Elsevier Ltd. Great Britain.
3. Dooley, J. (2011). *Software Development and Professional Practice*. Springer Science & Business Media. New York, NY, USA.
4. Griffiths, P. (2005). *Risk-Based Auditing*. Gower Publishing Limited. Hants, England.
5. Harris, S. (2013). *CISSP Exam Guide 6th Edition*. McGraw Hill. New York, USA.
6. Hopkin, P. (2010). *Fundamentals of Risk Management*. Kogan Page Limited. London, United Kingdom.
7. Institute of Internal Auditors (IIA). (2005). *Global Technology Audit Guide: Change and Patch Management Controls: Critical for Organizational Success*. Institute of Internal Auditors. Florida, USA.
8. Information Systems Audit and Control Association (ISACA). (2009). *The Risk IT Framework*. Information Systems Audit and Control Association. Rolling Meadows, IL, USA.
9. Jones, C. (2010). *Software Engineering Best Practices*. McGraw-Hill. New York, NY, USA.
10. Kindsight. (2014). *Kindsight Security Labs Malware Report – H1 2014*. Available from: http://resources.alcatel-lucent.com/?cid=180437. (Accessed on 25 September 2014).
11. Meyer, M.J. & Lambert, J.C. (2007). Patch Management: No Longer Just an IT Problem. *The CPA Online Journal*. Available from: http://www.nysscpa.org/cpajournal/2007/1107/essentials/p68.htm (Accessed on 8 June 2013).
12. Pressman, R.S. (2010). *Software Engineering:A Practitioner's Approach, Seventh Edition*. McGraw-Hill. New York, NY, USA.
13. PriceWaterHouseCoopers (PwC). (2014). *Stability amid uncertainty: South Africa – Major banks analysis*. Available from: http://www.pwc.co.za/en_ZA/za/assets/pdf/major-bank-analysis-september-2014.pdf (Accessed 8 March 2015).
14. Rainer, R.K., Snyder, C.A. & Carr, H.H. (1991). Risk analysis for information technology. *Journal of Management Information Systems*, Volume 8, No 1:129-147.
15. SANS. (2009). *2009 CWE/SANS Top 25 Most Dangerous Programming Errors*. Available from: https://www.sec.in.tum.de/assets/lehre/ws0809/sire/2009_cwe_sans_top_25.pdf (Accessed on 31 July 2014).
16. Sommerville, I. (2011). *Software Engineering, Ninth Edition*. Addison-Wesley. Boston, Massachusetts.
17. Statistics South Africa. (2012). *Census 2011:P0301.4*. Available from: http://www.statssa.gov.za/Publications/P03014/P030142011.pdf (Accessed on 7 October 2013).
18. Sun Microsystems. (2004). *Solaris Patch Management: Recommended Strategy*. Sun Microsystems. Santa Clara, CA, USA.
19. Taylor, R., Allen, J.H., Hyatt, G.L. & Kim, G.H. (2005). *Global Technology Audit Guide, Change and Patch Management Controls: Critical for Organisational Success*. Institute of Internal Auditors. Altamonte Springs, FL, USA.
20. Trent, R. (2004). *The Administrator Shortcut Guide to Patch Management*. New Boundary Technologies. Realtimepublishers.com
21. Wikipedia. (2015). *Big Four Banking*. Available from: http://en.wikipedia.org/wiki/Big_Four_(banking) (Accessed on 8 March 2015).