INFORMATION TECHNOLOGY AUDIT APPROACH FOR THE ASSESSMENT OF SOFTWARE PATCH MANAGEMENT

by

Deon Oosthuizen

MINI DISSERTATION

Submitted in fulfilment of the requirements for the degree

MASTERS COMMERCII

in

COMPUTER AUDITING

in the

FACULTY OF ECONOMIC AND FINANCIAL SCIENCES

at the

UNIVERSITY OF JOHANNESBURG

Supervisor: Professor Ben Marx

July 2015

ACKNOWLEDGEMENTS

A special thanks to my study supervisor, Professor Ben Marx, without whose assistance this study would not have been possible.

To my family, for always standing by me during my academic career and encouraging me to undertake this study.

I would also like to express my appreciation to the internal auditors who assisted in providing feedback through the survey in this study. The overwhelmingly positive response received on the survey is an indication of the commitment of internal auditors to advancing the profession and assisting fellow auditors in their endeavours.

Finally, a word of thanks to my employer, FirstRand, for their assistance in enabling me to undertake this master's study.

ABSTRACT

Computer software is ubiquitous and is driven extensively by our information-based society. However, little consideration is given to the complex task of developing software, which may involve conflicting objectives.

Developing software that is free from material defects is the ultimate goal for software developers; however, due to its cost and complexity, 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. The task of patching software to rectify inherent flaws may be a simple operation on computer systems that are of low significance, but is far more complex and critical on high-risk systems. Software flaws that exist within computer systems may put confidential information at risk and may also compromise the availability of such systems. One of the environments that is extremely susceptible to software flaws is the South African banking system, where not only is confidentiality a critical imperative, but also where high system availability is expected by the banking public.

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 reasons for software patching, the discipline of risk management relating to IT and software patching are also identified as fundamental to the audit approach for assessing the process.

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. Despite these organisations being extremely mature from a risk management perspective, the auditors believed that the patching process may benefit from an increased focus on risk management.

Keywords:

Software patches

Software patch management

Software flaws

Risk assessment

Risk mitigation

Confidentiality

Integrity

Availability

Downtime

Information security

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION AND STUDY LAYOUT

1.1. INTRODUCTION

As recently as 15 to 20 years ago, computers were the preserve of a select few people. They were primarily used as business tools and were prohibitively expensive for the average consumer to purchase. While computer networking has existed since the 1980s, it was originally only used within large organisations to enable teamwork and collaboration. In fact, before the days of the Internet, many home computers were used as standalone word processing or spreadsheet systems and were not connected to any form of network. The IT landscape has changed substantially since then, and as a result there have been major changes in the way computers are used, particularly within the home setting. It is the rapid proliferation of computers and related devices, such as mobile smart phones and tablets that have led to today's connected existence, an existence that would have been unimaginable just a few decades ago.

1.2. RISE IN PREVALENCE OF COMPUTER AND INTERNET ACCESS

In the United States, computer adoption rates (based on census figures) have shown that households with computers have increased from 37% in 1997 to 77% in 2010 (Economics and Statistics Administration, 2011:1), while in South Africa the proportion of households with computers increased by a more moderate percentage – from 8.5% in 2001 to 21.4% in 2011 (Statistics South Africa, 2012:71). This figure may initially appear low, when considering that many mobile phones have the computing power that a full computer had just a few years ago, it is conceivable that mobile phones could be added to the equation. In contrast to computers, mobile phone penetration in South Africa has increased from 31.9% in 2001 to 88.9% in 2011 (Statistics South Africa, 2012:71). Such a huge percentage increase over a relatively short period of 10 years, together with the continued pace of technological advancement, have contributed to our society becoming ever more dependent on information technology in every aspect of our lives.

However, the rapid increase in computers and related devices is only one side of the equation. It is the greater connectivity between computers that was brought about by

the Internet in the 1990s that really changed the dynamics of how society operates today. This is evidenced by the United States' Internet penetration rate, which has increased from 19% in 1997 to 71% in 2010 (Economics and Statistics Administration, 2011:1). In the United Kingdom, the Internet penetration has increased from just under 10% in 1998 to 80% in 2012 (Office for National Statistics, 2012:3). In South Africa, the current Internet penetration was measured at 35.2% in 2011, which still lags well behind that of the USA and UK. However, it is 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. This information and interconnectivity has resulted in the shrinking of the globe. With a greater degree of shared knowledge and communication ability, individuals can now coordinate activities and share information more freely than ever before. While this is greatly beneficial for learning and collaboration, there is also a greater ability to inflict damage on systems and companies than ever before, whether this is with the intention to commit fraud, cause reputational damage or simply make computing services unavailable.

1.3. EMERGENCE OF SOFTWARE THREATS

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. In the early days of the Internet, website hacking and defacing started to become a serious issue, one that was often overlooked when organisations undertook the decision to create an Internet presence. 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. This is a sentiment echoed by Sommerville (2011:367), who noted that "the widespread use of the Internet starting in the 1990s introduced a new challenge for software engineers—designing and implementing systems that were secure." In effect, the greater interconnectivity of the 21st century now enables an individual or group in a different country to use a computer or even a mobile phone to attack any of your company's systems that are connected to the Internet.

In a 2014 Kindsight (2014:3) malware report, the following trends were identified 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.

As the difficulty of producing dependable systems increases, systems engineers are required to consider threats from malicious and technically skilled attackers, as well as problems resulting from accidental mistakes in the development process (Sommerville, 2011:367). It is therefore essential to design systems to withstand external attacks and to recover from such attacks. Without security precautions, it is likely that attackers could compromise a networked system. A successful attack may misuse the system hardware, steal confidential data, or disrupt the services offered by the system. System security engineering is therefore an increasingly important aspect of the systems engineering process (Sommerville, 2011:367).

Security engineering as a discipline is a relatively new concept that is concerned with the development and evolution of systems that can resist malicious attacks intended to damage the system or its data. This discipline is now seen as an extension of the existing area of software security engineering, which is part of the more general field

of computer security. The practices contained within these disciplines have become a priority for businesses and individuals as more and more criminals attempt to exploit networked systems for illegal purposes. Software engineers should be aware of the security threats faced by systems and ways in which these threats can be neutralised (Sommerville, 2011:367). The practice of security engineering can, however, only be effective to a point, as the development of software, no matter how well intentioned, will always be prone to errors and potential security issues. This is where the discipline of software patching comes into play, and this needs to be enforced to address security bugs which may be present in software that has already been released to the market.

1.4. PATCHING SOFTWARE FLAWS

The American National Institute of Standards and Technology (hereafter NIST) provides the following definition of a software patch: “Patches are additional pieces of code developed to address problems (commonly called ‘bugs’) in software. Patches enable additional functionality or address security flaws within a program.” A security flaw or ‘vulnerability’ is a weakness in software that an attacker would typically be trying to exploit in order to gain unauthorised access. NIST provides this definition of a vulnerability: “Vulnerabilities are flaws that can be exploited by a malicious entity to gain greater access or privileges than it is authorized to have on a computer system” (NIST, 2005:8).

Patch management is defined as the process of applying patches to software systems. The NIST’s definition of patch management is that “patch and vulnerability management is a security practice designed to proactively prevent the exploitation of IT vulnerabilities that exist within an organization.” The NIST advocates an approach to patch management where “proactively managing vulnerabilities of systems will reduce or eliminate the potential for exploitation and involve considerably less time and effort than responding after an exploitation has occurred” (NIST, 2005:8). What this means is that when a software patch is released, it should be applied to the affected system such that the documented flaw is then addressed. While NIST’s advocacy of a proactive approach may be well founded, it is important to determine what is meant by being proactive, particularly in terms of the timeframes within which patches should be deployed.

1.5. SOFTWARE ATTACKS AND THEIR IMPACT

In October 2013, the software company Adobe, maker of Adobe Reader, a widely used application for reading documents in the Portable Document Format (hereafter PDF), announced a hack had taken place on their network. The security breach is said to have involved information relating to 2.9 million Adobe customers, including customer names, encrypted credit or debit card numbers, expiration dates and other information relating to customer orders (Poeter, 2013:1). The scale of this exposure and the prominence of Adobe as a leading software company illustrate the real threat to computer systems posed by hackers.

In April 2011 the Sony PlayStation network was hacked and personal information from its 77 million user accounts was compromised. It is claimed that the Sony breach is expected to cost $1.5 billion (Basirico, 2011:1). While Sony has not officially divulged the method used to hack into their servers, it is proposed by Basirico (2011:1) that "Sony servers were running outdated software versions with documented vulnerabilities, including a service to encrypt data communication, but one that allows unauthorized access. The hackers identified that Sony was running software that was loaded with vulnerabilities." This case indicates that even a company with the stature of Sony may not be entirely proficient with the process of patch management.

In 2014, it was noted by McGregor (2014:1) that "cyber attacks and data breaches don't look like they're going to slow down. We've seen high-end data breaches of large companies, with data, personal records and financial information stolen and sold on the black market in a matter of days." This was after a number of high-profile hacks, one of which included that of the global online shopping giant eBay, when hackers managed to steal the personal records of 233 million users.

Most recently, Microsoft's February 2015 security update was released with a total of nine bulletins security updates, of which three were rated "critical". One of these has reportedly been used in the wild by attackers targeting financial services firms and US military and government networks. Microsoft's update comes after a turbulent month for information security professionals, with multiple zero-day vulnerabilities found in Adobe's Flash software. It is noted that while "all of the known issues have been very quickly addressed by Adobe, the number of flaws cyber criminals are

finding in software widely used on a daily basis is worrying, wrote Wolfgang Kandek, (Chief Technology Officer at Qualys), in a blog post" Ashford (2015:1).

The PSN Infrastructure Security & Cyber Defence Team's stance on patch management highlights the importance of timeframes on patch management. They note that "software patch management is now a critical security issue because the time between the discovery of a vulnerability in an operating system or application and the availability of an exploit is becoming increasingly shorter." PSN provide a warning that some vulnerabilities are discovered by organised crime and hacker groups, which then exploit them for a period of time before the general security community/vendors become aware of the vulnerabilities. Such a vulnerability is commonly referred to as a 'zero-day' vulnerability, as no patch has been developed or deployed for the problem yet. PSN provide a definition for an attack using a zero-day vulnerability as "one which exploits a previously unknown vulnerability in a computer application, meaning that the attack occurs on 'day zero' of awareness of the vulnerability" (PSN Infrastructure Security & Cyber Defence Team, 2012:11).

On 8 October 2013 Microsoft released their usual monthly software patches to their customers; this included two patches relating to 'zero-day' vulnerabilities in Microsoft's Internet Explorer web browser. What Microsoft neglected to disclose, however, is that these exploits had already been used against companies prior to the patches being released (Wang, 2013:1). In his article for *PC Magazine*, Wang goes on to conclude that "it's great that Microsoft addressed and patched these vulnerabilities, but we have to remember that attackers were actively using these threats up until the patch. Keep in mind the threats could still be used against unpatched systems." This comment underlines that even the most effective patch management cannot protect against such zero-day vulnerabilities.

1.6. THE ROLE OF AUDITORS IN ASSESSING SOFTWARE RISKS AND PATCH MANAGEMENT

Meyer and Lambert (2007:1) concur on the importance of the patch management process and specifically comment on the role that should be played by accountants and auditors in the patching process:

> Accountants and auditors, both internal and external, should be aware of how their companies and clients are managing the patch process, because it is a key control in securing a company's data, including financial data. 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 increased the emphasis on and responsibility of auditors for assessing risk.

Meyer and Lambert (2007:1) note that "information-system security is critical for all organizations and its key component is patch management. Patch management is essential to the proper functioning of the financial reporting process, and auditors should recognize its importance because corrupted financial data can cripple an organization".

A recent survey by the external auditor PricewaterhouseCoopers (hereafter PwC) notes that there is an increasing focus on data privacy and security in organisations. This is not surprising as automation in organisations is increasing rapidly. Many organisations use technology as a differentiator; one such example is First National Bank (hereafter FNB), who are constantly pushing the envelope in terms of mobile banking products, electronic wallets and customer self-service. All of these innovations bring about additional risks to data privacy and security. In addition, new legislation such as Protection of Personal Information (hereafter POPI) seeks to put stronger controls on personal information, which in turn requires additional emphasis on information security. The PwC survey (2012:26) identified:

> Data privacy and security to be the single most requested area for increased internal audit focus, with 46% of stakeholders asking for internal audit to add capabilities in this area. The reality is that this risk is evolving so fast that most organizations cannot keep pace. It is becoming more complex, driven largely by the proliferation of technology, the increasing amount of personal data stored by

companies, and the ever-growing sophistication of those seeking access.

1.7. THE STATED RESEARCH PROBLEM

With the rapid progression of information technology, the prevalence of software across various platforms and devices and the inherent security weaknesses that form an ever-present danger, one of the most significant controls is software patching. Being such an important control to protect confidential information, software patching as a discipline is an important business consideration; however, it seems to be regarded largely as an IT problem. This study will seek to evaluate the approaches to software patching from a business and audit perspective (as a control measure), in order to investigate the most appropriate approaches to software patching for ensuring its effectiveness. The research problem is noted as follows:

> *An investigation into the information technology audit approach for the assessment of software patch management* in a manner that appropriately mitigates software patching risks.

1.8. RESEARCH OBJECTIVE

The primary objective of this study is to add to the existing body of knowledge through obtaining a thorough understanding of current practices from the existing literature and empirical evidence relating to:

- IT risk management with reference to software development and software patching;
- Software engineering and the reason why software defects exist;
- Recommended practices regarding the patch management process; and
- The audit assessment of software patching approaches.

The above objective will be achieved through a literature review (Chapters 2 to 4) and an empirical study of the current practices relating to patch management in South Africa's largest banking institutions. The methodology followed in the empirical study and the findings and conclusion of the empirical study are discussed in Chapter 5. The empirical study will use a questionnaire completed by the heads of internal audit at each of South Africa's largest banking institutions. These

participants were selected for their thorough knowledge of the business practices within their respective organisations.

1.9. STUDY LAYOUT

The study will be divided into the following chapters:

CHAPTER 1: INTRODUCTION AND STUDY LAYOUT

In this chapter the background to the study is discussed. The research problem and objectives of the study are also explained.

CHAPTER 2: ASSESSING THE RISK OF SOFTWARE AND PATCH MANAGEMENT FROM AN AUDIT PERSPECTIVE

Assessing IT risk is essential to any audit effort relating to software patching. The concept of risk management forms the cornerstone of reducing unwanted and negative outcomes for an organisation and ensuring its continued success. This chapter reviews the process of risk management and IT risk in order to provide an overview to determining and reducing the risk to the organisation, which must form the basis of any audit assessment on the process of software patching.

CHAPTER 3: THE IMPACT OF SOFTWARE DEVELOPMENT ON PATCH MANAGEMENT

This chapter will provide background on the process of software development, software defects and the reasons why software patch management will remain an important process given the current regime of software development practices.

CHAPTER 4: THE PROCESS OF SOFTWARE PATCHING AND THE AUDITING THEREOF

In this chapter, the process of patch management will be explored. Specific focus will be placed on the requirements of an effective patch management process from a theoretical perspective and what auditors should consider when assessing the patching process.

CHAPTER 5: EMPIRICAL STUDY AND RESEARCH FINDINGS

Based on the literature study of Chapters 2 to 4, a questionnaire was designed and developed to obtain information on business continuity risks and procedures in organisations. In this chapter, the methodology followed for the empirical study, as well as the findings from this study, are explained and discussed.

CHAPTER 6: CONCLUSION

In this chapter, conclusions are drawn from the literature review and empirical study. Recommendations for possible areas for further research are also made.

1.10. CONCLUSION

The importance of patch management to address and protect against vulnerabilities has been noted. However, there are many aspects to the practice of software patch management that need to be considered. With this in mind, the decision was taken to embark upon additional research to gain clarity on the topic of patch management and how to ensure that any software patch management process is as effective as possible in achieving its objectives. In particular, the timeframes within which software patches need to be deployed are of critical importance to the success of any patch management programme. This requirement is, however, at odds with another important requirement of any patching process, namely the effective testing of a patch prior to deployment. Without this, organisations run the risk that an inadequately tested patch could result in downtime for their systems. Such downtime is usually as a result of incompatibility between the patch update and either the system configuration or other software currently installed on the system. While software vendors typically perform some level of testing on patches before releasing them to their clients, they are unable to test all possible system configurations and software interdependencies. This is why it is typically required that an organisation performs their own testing on any patch before deploying it to a production environment.

This study will analyse the topic of patch management and identify the factors which need to be considered and should form the basis of a robust process for ensuring an effective patch management approach, all of which should form the basis of auditors' assessments of the patching process. Specific focus will be given to the timeframes for patch deployment by considering how software patches are rated from a risk

perspective. This is done with the aim of allowing a more informed decision in terms of the conundrum of the speed to patch (to ensure vulnerabilities are addressed quickly) vs. time to test a patch (to ensure that the deployment of a patch does not unexpectedly result in a system outage due to compatibility issues).

In the next chapter, the focus will be on the topic of risk management as an important driver for ensuring business success as well as a cornerstone of the execution of audits, especially when implementing a risk-based audit approach.

CHAPTER 2
ASSESSING THE RISK OF SOFTWARE AND PATCH MANAGEMENT FROM AN AUDIT PERSPECTIVE

2.1. INTRODUCTION

The management of risk is a critical business activity that focusses on reducing unwanted effects on the achievement of a business objective. In this chapter, the practice of software patch management will be investigated from a risk management perspective, with the view to identifying how the patching process should be approached to ensure that risk is adequately addressed.

2.1.1. RISK AND INTERNAL AUDIT

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). This explanation of how risk should impact what is audited is an important concept and will become the key pivot point in this study. The concept of risk-based auditing highlights the approach that auditors need to follow when performing their duties. It is important for auditors to remain cognisant that risk is arguably the most significant factor to consider, and thus it is important to assess any potential audit area and any recommendations based on the principle of risk.

2.1.2. RISK AND PATCH MANAGEMENT

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 a 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) introduces the concept of security risk management and highlights that key risk management concepts already explained previously in this study are relevant to this area: “Security risk assessment and management is essential for effective security engineering. Risk management is concerned with assessing the possible losses that might ensue from attacks on assets in the system, 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.

Meyer and Lambert (2007:3) draw attention to the varying types of software, and notes that not all software is of equal importance. It is noted that software containing confidential information, such as accounting software, sales contact information, or personnel applications, is often targeted by computer hackers, making it a high priority. Somerville (2011:370) makes an important comment on where the responsibility for risk management pertaining to software should reside:

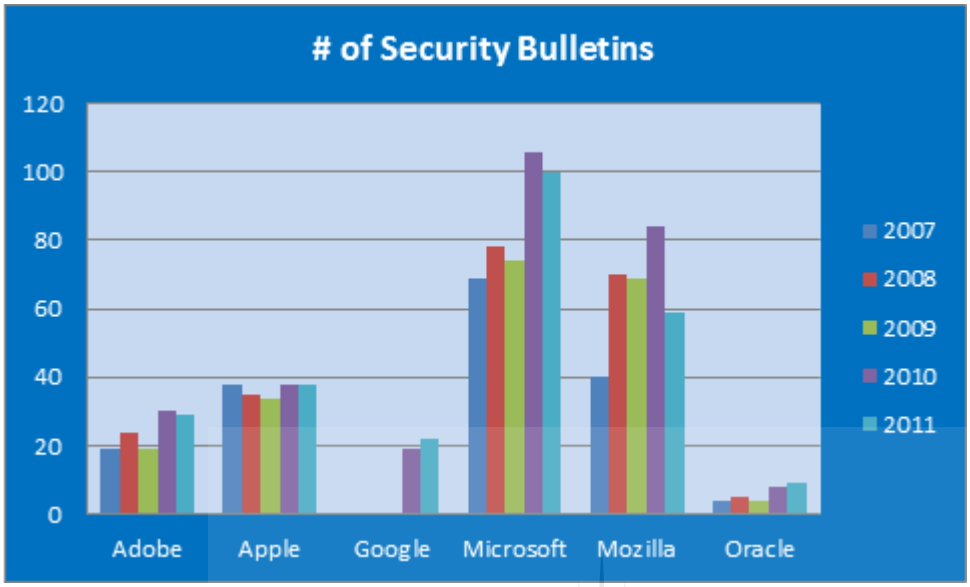
> Risk management is a business issue rather than a technical issue so software engineers should not decide what controls should be included in a system. It is up to senior management to decide whether or not to

> accept the cost of security or the exposure that results from a lack of security procedures. Rather, the role of software engineers is to provide informed technical guidance and judgment on security issues. They are, therefore, essential participants in the risk management process.

This indicates that software risk must be considered against the backdrop of all other risks and reviewed by management alongside other risks to ensure that there is no undue emphasis placed on either software or business risks.

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. It is noted that "security risk assessment should continue throughout the lifetime of the system to identify emerging risks and system changes that may be required to cope with these risks. This process is called operational risk assessment. New risks may emerge because of changing system requirements, changes in the system infrastructure, or changes in the environment in which the system is used" (Sommerville, 2011:375). The new risks, as mentioned by Sommerville (2011:375), may manifest themselves in an attack on a particular piece of software, either for malicious or criminal purposes. Such attacks may occur frequently, especially against software used extensively across the world. Microsoft, being one of the largest software providers in the world, is especially targeted in this area. Net Applications (Wikipedia, 2014:1) estimates Microsoft's operating systems' market share to be just over 90% of the world's computers. Microsoft has released around 100 security-related patches in both 2010 and 2011, which is showing an upward trend over previous years (GFI, 2012:1). This upward trend is evidenced in figure 2.1. which indicates the number of security-related bulletins for six major software vendors from 2007 to 2011. A security bulletin released by a software vendor is aimed at notifying users of a security vulnerability present within the software they provide. An applicable software patch to address this problem is typically provided as part of the security bulletin, however in certain cases it is possible that no security patch is yet available, in which case the vendor typically advises clients to disable or re-configure their software accordingly until such time a patch is made available.

Figure 2.1: Number of security-related bulletins released by major software companies



Source: GFI (2012:1)

2.2. THE DEFINITION OF RISK

The foundation of risk management is the concept of risk. The Oxford English dictionary defines risk as "a chance or possibility of danger, loss, injury or other adverse consequences", while the definition of "at risk" is "exposed to danger". This is, however, a very theoretical definition, and further research is required in order further the understanding of the term 'risk' from a practical viewpoint. Additional definitions relating to risk exist, which may shed more light on the practical side of risk. The identification of risks and what they mean to the business and internal audit will also be considered.

As Collier (2009:3) states, "we are all faced with a multitude of risks on a day-to-day basis, even if it is just crossing the road, driving our car, concern about school or university grades for ourselves or a member of our family, whether we will get the job we want or the salary increase or promotion we expect". This comment highlights the personal aspect of risk, and concludes that we inherently deal with risk on a daily basis and have a number of responses to it. Coleman's (2011:12) view on risk highlights a fundamental underlying assumption regarding risk, namely that it must have some negative consequence for an individual or organisation. If an individual

ventures onto a frozen lake, that person is seen to be taking a risk not just because the ice may break, but because if it does break the result may be bad. In contrast, in the case of a lake that no one is trying to cross, it may be noted that there is instead a 'chance' of the ice breaking. It follows that the term 'risk' is only used should the breaking ice have an impact on someone or something. This idea that risk should have some sort of impact is a key consideration in the understanding of what could possibly constitute a risk.

Hopkin's (2010:12) approach to understanding the term 'risk' is to collate the definitions of risk from various organisations and through comparison arrive at a universally accepted definition. In doing so, Hopkin (2010:12) provides an overview of the definitions used by a number of bodies and standards organisations:

Table 2.1: Definitions of risk

Organisation	Definition of risk
ISO Guide 73 ISO 31000	Effect of uncertainty on objectives. Note that an effect may be positive, negative, or a deviation from the expected. Also, risk is often described by an event, a change in circumstances or a consequence.
Institute of Risk Management (IRM)	Risk is the combination of the probability of an event and its consequence. Consequences can range from positive to negative.
"Orange Book" from HM Treasury	Uncertainty of outcome, within a range of exposure, arising from a combination of the impact and the probability of potential events.
Institute of Internal Auditors	The uncertainty of an event occurring that could have an impact on the achievement of the objectives. Risk is measured in terms of consequences and likelihood.
Hopkins own derived definition	Event with the ability to impact (inhibit, enhance or cause doubt about) the mission, strategy, projects, routine operations, objectives, core processes, key

	dependencies and / or the delivery of stakeholder expectations.

Source: Hopkin (2010:12)

Through the review of these rather varying definitions of risk, a few commonalities can be noted: that there exists some "uncertainty" which in turn may culminate in a "circumstance" or "consequence", the outcome of which may have an "impact" on "objectives", a "mission", a "project" or "processes".

The definition of risk, as noted by Hopkin (2010:12) provides a broad generic explanation. However, when relating risk specifically to an organisation in a business context, Griffiths (2005:17) arrives at a definition that is very similar to that of the Institute of Internal Auditors and which is related to the purpose for any business to exist: "The threat that an action or event will adversely affect an organisation's ability to achieve its objectives and execute its strategies successfully."

Many definitions of risk tend to focus on negative consequences, but the International Organization for Standardization (hereafter ISO) in their Risk Management Standard 31000 and the Institute of Risk Management (hereafter IRM) highlight that risk does not necessarily have to impact objectives in a negative way; it may also have a positive outcome (ISO, 2009). Griffiths (2005:17) also echoes this view by providing an expanded definition of risk as "The chance of something happening that will have an effect on business objectives." In this definition he omits the view the risk must have a negative connotation. This is a concept that does not appear to be widely shared when referring to risk as most definitions tend to dwell on the negative consequences.

Given the definition of risk identified, software patching may have two noticeable risks that could manifest if the patch management process is not effective. The first involves failure to apply a patch in a timeous manner such that an open security exposure is exploited, resulting in loss of confidential information and reputational damage (Stanton, 2005:13). The second relates to lack of effective testing before deploying a patch, where the deployment of the patch could result in a system crash resulting in downtime. The significance of the downtime would be dependent on the importance of the system and whether production is affected or sales are lost in the process (Stanton, 2005:6).

2.3. RISK CATEGORISATION

Subsequent to the identification of what constitutes risk, it is appropriate to investigate the broad categories of risk. This will enable the conceptualisation of the frame of reference and provide guidance on where to look when identifying risks. According to Collier (2009:60), at a high level, risks faced by businesses can be split into four key areas:

- Business or operational: relating to the activities carried out within an organization;
- Financial: relating to the financial operation of a business;
- Environmental: relating to changes in the political, economic, social and financial environment; and
- Reputation: caused by failing to address some other risk that may have an effect on the reputation of an organisation.

Given these high-level risk categories and classifications of risks, the next step may consider what potential factors or sources could give rise to these risk areas. These are referred to as risk drivers. Hopkin (2010:31) proposes six broad categories in organisations that can cause risks to occur and provides some examples relating to each category:

Table 2.2: Risk drivers

Category	**Examples of disruption**
People	Lack of people skills and/or resources. Unexpected absence of key personnel. Ill-health, accident or injury to people.
Premises	Inadequate or insufficient premises. Denial of access to premises. Damage to or contamination of

Category	Examples of disruption
	premises.
Assets	Accidental damage to physical assets. Breakdown of plant or equipment. Theft or loss of physical assets.
Suppliers	Disruption caused by failure of supplier. Delivery of defective goods or components. Failure of outsourced services and facilities.
Information technology (IT)	Failure of IT hardware systems. Disruption by hacker or computer virus. Inefficient operation of computer software.
Communications	Inadequate management of information. Failure of internal or external communications. Transport failure or disruption.

Source: Hopkin (2010:31)

As per Hopkin's (2010:31) risk categories, one identified driver of risk is information technology. This is the key category that will be relevant to the topic of software patch management. However, Collier's financial risk area may also be relevant, as there are a number of possible financial impacts related to deploying software patches. These could include: the actual cost of maintaining a patching process, the impact of a system outage due to a prematurely deployed patch or the impact of the loss of confidential information due to a security breach.

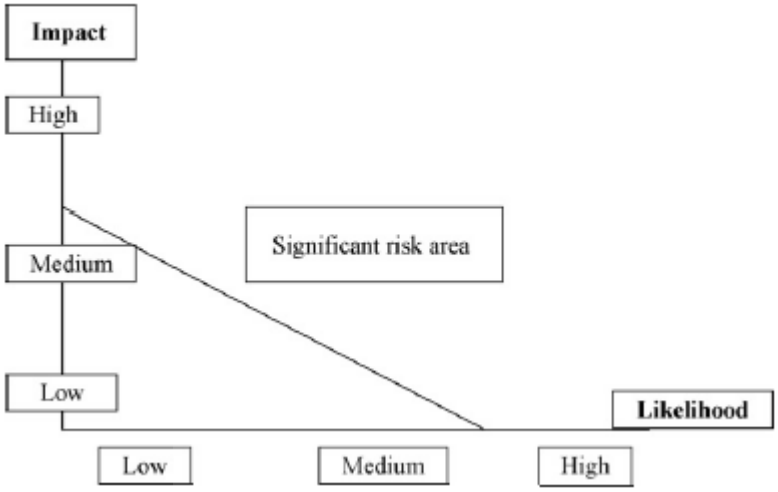
2.4. RISK ASSESSMENT

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.

The figure below represents a simple risk matrix on which risks can be plotted in terms of their impact and likelihood. The diagonal line indicates the organisational risk appetite. Those risks that manifest above the line are then deemed significant risks facing the organisation, which need some form of treatment by virtue of their being above the threshold. Risks below the threshold may well be considered tolerable or of a low priority. By considering the likelihood and impact (i.e. severity) of each of the risks (or risk categories) it is possible for organisations to prioritise the key risks (Collier, 2009:85).

Figure 2.2: Impact and likelihood matrix



Source: Collier (2009:85)

Quantitative risk analysis methodologies do, however, have some disadvantages. Notably, estimating the probability of damage or loss of each asset is often imprecise. In addition, the probability distribution of losses is highly skewed. Many circumstances can cause minor problems, but few circumstances can cause major problems. Quantitative risk analysis tends to average these events, thus blurring the differences between the extremes and implying similar solutions (Rainer et al, 1991:138).

It may be neither necessary nor desirable to spend the time and effort required to perform a quantitative risk analysis. Management may decide that only a quick evaluation of the firm's risk posture is needed. In such cases, qualitative risk analysis approaches may be used. Qualitative methodologies attempt to express risk in terms of descriptive variables, rather than in precise monetary terms. These approaches are based on the assumption that certain threat or loss data cannot be appropriately expressed in monetary or discrete events, and that precise information may be unobtainable (Rainer et al, 1991:138).

Qualitative risk analysis methodologies 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. Qualitative methodologies may, however, also be somewhat imprecise. The variables used (i.e. low, medium, and high) must be clearly understood by all parties involved in the risk analysis, including management. In general, it is noted that management may consider qualitative methodologies suspect because they do not provide monetary values and probabilities (Rainer et al, 1991:140).

Whether to use a qualitative or quantitative methodology for assessing risk exposures relating to software vulnerabilities would clearly be a decision based on each organisation's risk appetite and dependence on information technology. Nevertheless, it is difficult to apply a quantitative methodology because of the large number of unknown factors that need to be taken into consideration. For example, while the possible impact of any known software vulnerability may be known, any actual exposure is likely to be unpredictable due to the complexity of computer systems, networks and other manual processes involved in different organisations. Stiennon (2012:8) concurs by noting that it is impossible to assign value to all IT assets due to the sheer complexity involved. For example, e-mails cannot all be classified as the same, because they have varying importance.

2.5. 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. These concepts are sometimes confused, but there is some distinction between the two. 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).

ISACA's Risk IT framework defines two major factors that are important when considering how to set risk appetite levels for an enterprise. The first factor is the enterprise's capacity to absorb loss: i.e. how capable the organisation is to incur, for example, financial loss and reputational damage. The second factor is management's culture or predisposition towards risk-taking. This can either be cautious or aggressive, and defines the amount of loss the enterprise wants to accept in pursuing a return. Risk appetite can be defined in practice in terms of combinations of the frequency and magnitude of a risk, and it must also be noted that risk appetite can and will be different amongst enterprises. Each organisation

will be willing to accept a different level of risk based on its objectives, the market conditions and the possible profits that can be derived from accepting particular risk levels.

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).

Risk appetite and risk tolerance go hand in hand. Risk tolerance is typically defined at the enterprise level and is reflected in policies set by the executives. Any business initiative includes a risk component, so management should have the discretion to pursue new opportunities and risk. Enterprises in which policies are cast in stone rather than being more fluid could lack the agility and innovation to exploit new business opportunities. Conversely, there are situations where policies are based on specific legal, regulatory or industry requirements where it is appropriate to have no tolerance for failing to comply (ISACA, 2009:18). It is noted that different organisations will have differing risk levels and that the organisation's risk culture (shared attitudes, values and practices) will play a significant role in how much risk an organisation is willing to accept. While older and more established organisations, especially those in heavily regulated industries, may choose to accept less risk, newer and more agile organisations in dynamically changing industries may accept more risk (Collier, 2009:72).

ISACA (2009:18) notes that the cost of mitigation options can affect risk tolerance, as there may be circumstances where the cost of risk mitigation options exceeds an enterprise's capabilities/resources, thus forcing the organisation to accept a higher tolerance for one or more risk conditions. An example given is that "if a regulation says that 'sensitive data at rest must be encrypted', yet there is no feasible encryption solution or the cost of implementing a solution would have a large negative impact, the enterprise may choose to accept the risk associated with regulatory non-compliance, which is a risk trade-off." However, Hopkin (2010:236) notes that while an organisation may choose to accept a certain level of risk, the actual level of risk may be quite different to the anticipated level.

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.

2.6. RISK TREATMENT

Subsequent to the identification and assessment of risks, the next process is how to deal with the risks that are above the tolerance level for the organisation. Collier (2009:3) notes that "we are more aware of risks when we take out insurance policies on our lives, our homes or cars. We also face risk in our workplaces as occupational health and safety regulations are properly concerned with what we do and how we do it, so that we can return home safely at the end of each day".

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.

Collier (2009:89) defines avoidance as the action that is taken to exit the activities giving rise to risk, such as a product line, geographical market or a whole business unit. ISACA (2009:28) concurs and provides a definition of avoidance as a means of exiting the activities or conditions that give rise to risk. It is noted that risk avoidance often applies when no other risk response is suitable. ISACA (2009:28) further states that it may be necessary to practise risk avoidance when:

- There is no other cost-effective response that can succeed in reducing the frequency and magnitude below the defined thresholds that the organisation has defined for risk appetite;
- The risk cannot be shared or transferred; and
- The risk is deemed unacceptable by management.

Risk reduction means that action is taken to detect the risk (ISACA, 2009:28), followed by action to mitigate it in some manner (i.e. reduce the likelihood or impact of the risk, or possibly both). Generally, this is achieved by the use of internal controls. Collier (2009:90) notes that risks to be subjected to risk reduction are typically risks that occur more frequently but have less impact. ISACA (2009:28) notes that other ways of reducing risk could include strengthening overall risk management practices or introducing a number of control measures intended to reduce either the frequency of an adverse event happening and/or the business impact of an event should it happen.

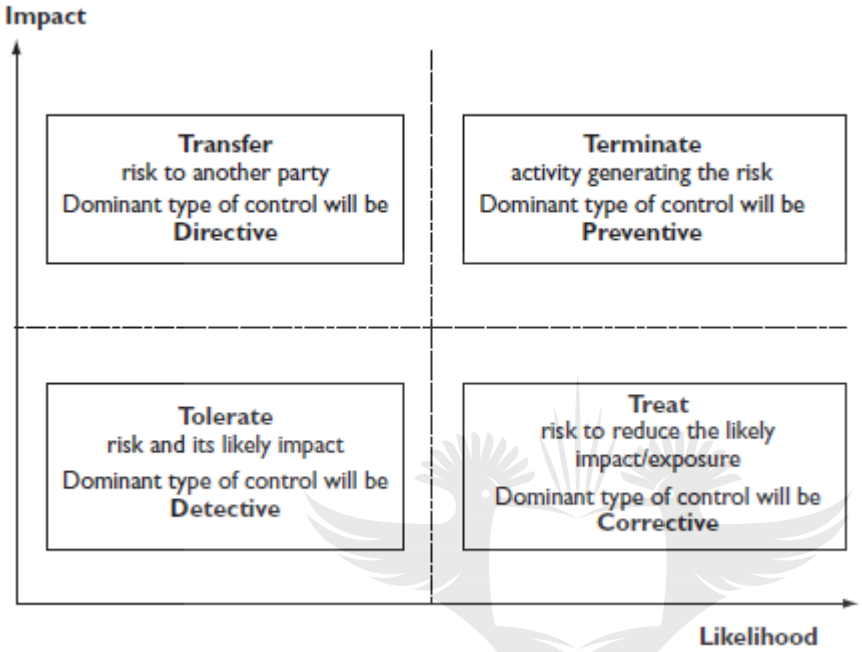
Risk sharing is an action taken to transfer a portion of the risk through, for example, insurance, pooling risks, hedging or outsourcing. Collier (2009:90) suggests that such risks are typically of a significant nature, though they usually occur less frequently. Sharing should reduce risk frequency or impact through transferring or otherwise sharing a portion of the risk (ISACA, 2009:28). ISACA (2009:28) notes possible risk-sharing initiatives as being insurance coverage, outsourcing part of the IT activities, or sharing IT project risk with the provider through fixed price arrangements or shared investment arrangements. It is, however, noted that while risk may be shared, in a legal sense these techniques do not relieve an enterprise of a risk. Nevertheless, they can involve the skills of another party in managing the risk and reduce the financial consequence if an adverse event occurs.

Risk acceptance is typically defined as taking no action to affect a risk's likelihood or impact (Collier, 2009:90). Thus, the resulting loss must be accepted if or when it occurs (ISACA, 2009:28). It is also noteworthy that risk acceptance is not the same as being ignorant of risk, but rather assumes that the risk is known and the possible outcome is accepted. Furthermore, an informed decision must have been made by management to accept the situation as such. If a particular risk is assessed to be extremely rare but very important, and approaches to reduce it are prohibitive, management may decide to accept it (ISACA, 2009:28). However, it is suggested by Collier (2009:90) that generally these types of risks have low impact even when they do occur, which may be more frequently.

Some suggestions in terms of linking the assessment of the risk (impact and likelihood) to a risk mitigation strategy are proposed by Hopkin (2010:246), and

drawing from this, Figure 3 below advises what possible action can be taken depending on the level of the risk.

Figure 2.3: Risk mitigation strategies

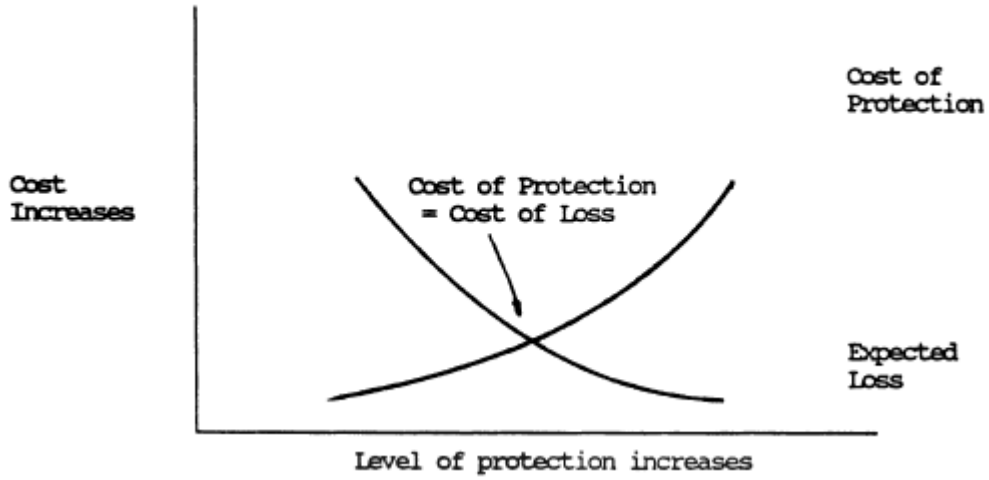


Source: Hopkin (2010:246)

While this figure may provide a guide for the most probable method of risk treatment, it may not be the most appropriate in all business circumstances. For example, in very new and innovative businesses, a risk with high impact and high likelihood may be at the core of the business and termination may not be the most appropriate course of action if the business is to remain viable.

Rainer et al (1991:132) considers the trade-off between the cost of protection and the cost of a loss. They suggest that an organisation would not typically incur a greater expense through addressing a risk than through the exposure that would result if the risk materialises. There is a crossover point after which it is not financially beneficial to address a risk based on the expected loss. This is graphically depicted in Figure 4 below.

Figure 2.4: Cost of protection vs. expected loss



Source: Rainer et al (1991:132)

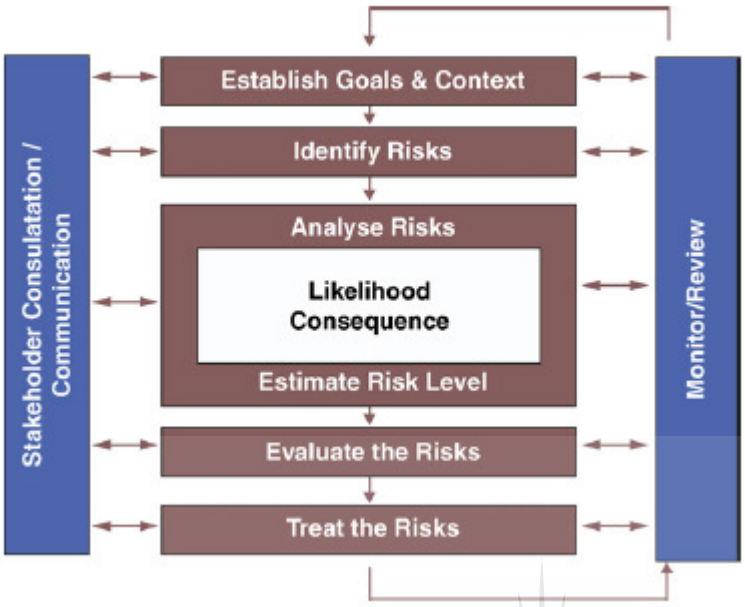
When considering whether to implement any software patch, it will be necessary to understand the impact and likelihood as mentioned, this would provide some suggestions on how to proceed. For instance, it may not be necessary to implement a software patch that has a low impact and likelihood due to the cost involved as well as the risk of downtime.

2.7. THE RISK MANAGEMENT PROCESS

The process of risk management encompasses all the areas already mentioned, namely: risk identification, assessment and treatment. Risk management, however, involves some additional steps that will be expanded on in this section.

Simply stated, risk management seeks to avoid or lessen loss. Loss implies injury to, denial of access to, or destruction of assets (Rainer et al, 1991:130). The opportunity for a threat to impact negatively on an asset is called a vulnerability. Risk is present when an asset is vulnerable to a threat. In general, the process of managing risk involves a number of individual steps. Each of these steps then contributes to the overall objective, which is ultimately to reduce all risks to a tolerable level. A number of steps may be needed to facilitate a risk management process, as indicated in Figure 5 below.

Figure 2.5: Risk management process



Source: Collier (2009:54)

The process of risk management typically comprises:

1. Establish of goals and context for risk management such that the objectives and expected outcome are understood (Collier, 2009:53). The first step in the risk management process is to establish the environment and the goals for the environment. It is against these goals that the risk assessment is undertaken (le Roux & van Waveren, 2008:80).

2. Identify any potential risks that may arise through the process (Collier, 2009:53). Risk identification, as noted by le Roux and van Waveren (2008:80), is "the process of determining what can happen, why and how". The quality of a risk assessment greatly depends on identifying and understanding hazards and unwanted events, as well as assessing the specific risks. The noted purpose of this stage is to identify the risks that are likely to affect the achievement of the goals of the established environment.

3. Analyse the identified risks by evaluating their likelihood and consequences in order to estimate the level of risk faced (Collier, 2009:53). As noted by le Roux and van Waveren (2008:80), this step in the risk assessment process requires that, for

each risk, the current controls and their effectiveness must be identified. The risk level must also be established by using a risk matrix that is specific to the company.

4. Evaluate and rank the identified risks such that focus can be given to the most severe risks first (Collier, 2009:53). While le Roux and van Waveren (2008:80) note that this step of the risk assessment process requires assessment of the level of risk as acceptable (risk is sufficiently low, and treatment is not considered cost-effective) or unacceptable (risk requires treatment). The categorisation of the risk according to the risk rating should therefore guide this decision.

5. Treatment of the risks by applying the most appropriate options (Collier, 2009:53). The objective of this stage of the risk assessment process is to develop cost-effective options for treating the risks. Treatment options are driven by outcomes that include avoiding, reducing, transferring, or retaining the risk (le Roux & van Waveren, 2008:80).

6. Regularly monitor the risk environment, as risks and their priorities do not necessarily remain constant. Existing risks need to be regularly monitored, and new risks and their impact should be included in the risk management plan of the business (le Roux & van Waveren, 2008:80).

7. Establish channels for communicating areas of high risk to internal and external stakeholders. Furthermore, processes should also be implemented that will escalate risks to the risk coordinator. This will ensure that the risk categorisation criteria are revised on a regular basis to portray the real risks (le Roux & van Waveren, 2008:80).

2.8. INFORMATION TECHNOLOGY RISK AS A DISCIPLINE

The importance of IT risk to an organisation is highlighted by Dhillon and Backhouse (1996:65): “Organizations are still trying to cope with the mystique that surrounds technology and the myriad benefits that could be derived from it. Furthermore there is reluctance on the part of users to deal with IT-related risks. Consequently, far less risk evaluation is done for computer-based systems than is the case for manual paper-based systems.” While risk management as a practice is well established and has been practised in some form or another for hundreds of years, how to treat

information technology risks is still somewhat in its infancy. This is borne out by the literature, as there is minimal focus on information technology risks specifically.

Westerman and Hunter (2007:1) provide insight on what IT risk encompasses by explaining that a "half century of adopting information technology at an astonishingly rapid rate has created a world in which IT is not just widely present but pervasively, complexly interconnected inside and outside the enterprise. As enterprises' dependence and interdependence on IT have increased, the consequences of IT risk have increased as well." IT risk involves the potential for an unplanned event involving a failure or misuse of IT to threaten the objectives of an enterprise. An IT risk incident has the potential to produce substantial business consequences that touch a wide range of stakeholders – "in short, IT risk matters now more than ever" (Westerman & Hunter, 2007:1). As is further noted by Westerman and Hunter, the rapid pace of information technology creates a major problem concerning risk management, given that the current pace of technology seems to show no signs of abating. A concerted effort will need to be made to enable risk management to 'play catch-up' if it is to be successful in the area of information technology. IT risk then follows the same definition of risk already detailed, but just focusses on risks that affect information technology in some way. It has already been noted that technology is a risk area, and the reason for creating a specialisation called IT risk is the additional complexity involved in assessing this type of risk.

IT risk has a very important role to play in organisations due to the prevalence of information technology, which enables business operations to achieve their objectives. According to Hopkin (2010:268), with increasing dependency on computer systems, it is important for organisations to identify the losses that could occur and take actions to manage the associated risks. It is generally considered that the main causes of loss associated with the IT systems are as a result of:

- Theft of computers and other hardware, as mentioned by both Hopkin (2010:268) and Rainer et al (1991:131);
- Potential unauthorised access into IT systems as alluded to by Hopkin (2010:268);
- The introduction of viruses into IT systems, which is suggested by both Hopkin (2010:268) and Rainer et al (1991:131);

- Hardware or software faults and failures, as noted by Hopkin (2010:268);
- User errors, which may include the loss or deletion of information, which is a concern according to Hopkin (2010:268);
- The possibility of IT project failures, as alluded to by Hopkin (2010:268);
- The potential for the theft of data, noted by Rainer et al (1991:131);
- The risk of disclosure, modification, and/or destruction of data per Rainer et al (1991:131); and
- The risk of hackers compromising IT systems (Rainer et al, 1991:131).

Delving further into the technical means of exploitation for the possible threat categories, Shimonski (2004:3) identifies the following steps that may be used by an attacker:

- **Reconnaissance**: This is when a potential attacker probes an IT network or systems ('knocking on your door') to map out the network and systems for a future malicious attacks. It may also be common for a potential attacker to identify vulnerabilities in software that could be used to gain unauthorised access to systems and data.
- **Denial of Service (DoS)**: The purpose of a DoS attack is to render computer systems inoperable. Such an attack can be very serious in nature and is often very simple to perform. Reasons for a DoS attack may vary, but it may be in an attempt to exact revenge, extort money or simply prove a technical ability to successfully execute such an attack. It is suggested that a DoS attack may be launched by using computers around the world that have been compromised through a security vulnerability. Shimonski (2004:4) notes that "many efforts have been made to patch systems that could either launch a DoS attack or be affected by one, but to think that top level hackers (the Elite) aren't constantly working on new ways to exploit systems is foolish" This highlights the need for appropriate patch management, a topic that will be further explored in this study.
- **Data Manipulation**: This type of attack may occur when an attacker is able to exploit a system weakness in order to illegally modify data. It is noted by Shimonski (2004:5) that data is the lifeblood of any organisation today, as we move further towards a paperless society. The risk of data manipulation is

significant to any organisation and could easily destroy a business that does not have appropriate safeguards in place to protect against this type of attack.

Harris (2013:22) notes that within the information security fraternity there are generally three fundamental high-level principles of information security, referred to as the "AIC Triad". This represents the availability, integrity and confidentiality of information assets. Confidentiality relates to the safeguarding of information so that it is not made available to unauthorised individuals. Whether it be company intellectual secrets or credit card information, confidentiality is an extremely important IT risk area. Integrity relates to ensuring that all information either stored or transmitted over computer networks is correct and not subjected to unauthorised modification. Integrity is said to be upheld when the assurance of the accuracy and reliability of information and systems is provided and any unauthorised modification is prevented. Availability ensures reliability and timely access to data and resources to authorised individuals.

Assessing IT risks can follow a similar process to that already identified for any other risk, focussing on impact and likelihood in the case of a quantitative approach. According to the Institute of Internal Auditors (hereafter IIA), certain aspects of impact and likelihood can be modified to be more IT-specific by assessing:

- The likelihood of an IT process failure occurring and its potential impact;
- The likelihood that the IT process could fail in such a way that it causes critical IT functionality to fail; and
- Whether it is at least reasonably likely that critical functionality could fail without prompt detection and result in the failure to achieve the business objective (IIA, 2008:15).

With regard to the identification and assessment of IT risks, Stiennon (2012:8) notes that "it is the changing nature of risk that is impacting risk regimes today. It is impossible to know beforehand which IT assets will be of interest to an attacker." This highlights the importance of selecting the correct values for the risk impact, as the likelihood may not be easy to determine. It may also be necessary to take a more pessimistic approach to the value used for the likelihood in such instances. Considering as an example interconnected systems (such as the Internet), the likelihood of any single person with the requisite skills taking a decision to attack the

IT systems of your company will not be an easy metric to determine with a great degree of certainty.

In order to bridge the gap between the theory of IT risk and its practice, the details contained in a survey conducted by the Economist Intelligence Unit may help to shed some light. The Economist Newspaper Limited, trading as The Economist Group, is a multinational media company headquartered in London, United Kingdom that specialises in international business and world affairs information. Its principal activities are magazines, newspapers, conferences and market intelligence. The *Coming to Grips with IT risk Report: a global survey of 145 senior executives*, which was facilitated by the Economist Intelligence Unit on behalf of SAP, aimed to gain a deeper understanding of how companies define and mitigate IT risk.

The key findings of this report included the following:

- **Complexity is largely to blame for current risk levels.** The sheer complexity of IT applications and system architectures is the main source of risk exposure. The survey also revealed that there is a shortage of skilled project managers who can handle unwieldy IT projects (Economist Intelligence Unit, 2007:2).
- **IT risk management structures were found to be largely inadequate.** Only 13% of executives claimed that their companies have a comprehensive IT risk management structure in place. Although many believed that senior management was aware of the financial risks associated with IT failure, only 11% described their company's handling of IT risk as "highly effective" (Economist Intelligence Unit, 2007:2).
- **Customer service was the area most affected by IT failure.** This results from companies' growing reliance on real-time online interaction. If IT systems fail, customers are only a click away from another company (Economist Intelligence Unit, 2007:2).
- **Following loss of customers, revenue loss is what executives fear most from IT failure.** When the system is down, customers will buy from other sites. Another feared consequence of IT failure is damage to brand and reputation, especially if customer information is compromised (Economist Intelligence Unit, 2007:2).

- **Unplanned downtime is considered the most damaging risk.** This is much more serious than other hazards such as viruses or the leaking of sensitive company data. The prospect of IT downtime is of particular concern in the manufacturing and financial services sectors (Economist Intelligence Unit, 2007:2).

2.9. CONCLUSION

An appropriate summary of risk is that of Gheorghe (2010:32), who states that:

> The risk on organization level cannot be eliminated; it will exist all the time; the management of the organization is responsible with minimizing it to an acceptable level. Risk management should be a continuous process which begins by assessing the level of exposure of the organization and identifying the main incident risks. Once identified, risks have to be minimized using control procedure and finally residual risk should be adjusted at acceptable level.

These comments highlight the important aspects that risk management is an ongoing process and that the objective is not to eliminate risk, but rather reduce it to an acceptable level.

The entire software development process, for any software, should be subjected to appropriate risk management to ensure that all key risks are identified and the necessary responses are implemented to reduce the impact of these risks to a tolerable level. The discipline of software patching should also be considered as part of the risk management function within any organisation, as it has a bearing on the risks faced by the organisation. As noted by the ISACA (2009:28), the patching of software can have a significant impact on operations and on security for an organisation.

Information security is an important consideration in IT operations, but often takes second place to systems availability. This is because when dealing with a security weakness, the likelihood of a security breach is often lower than the likelihood of a possible system outage caused by a deployed patch that has not been appropriately tested. Essentially, IT managers see a security exposure as a remote and unlikely possibility, while system downtime is a real and ever-present threat. This was also the finding of the Economist Intelligence Unit's (2007:2) research, which indicated a major concern around IT downtime.

The conundrum that will form the basis of the rest of the study is that of deploying a security patch as quickly as possible while at the same time having to deal with the possible ramifications of system downtime if insufficient testing has been performed. In the next chapter, the process of software development is reviewed with the intent to highlight the compromises within this process that drive the need for organisations to implement an effective software patching process.

CHAPTER 3
THE IMPACT OF SOFTWARE DEVELOPMENT ON PATCH MANAGEMENT

3.1. INTRODUCTION

In the previous chapter, it was identified that risk is a critical concept to be considered in relation to the assessment of software vulnerabilities. In this chapter, the focus is on the process of software development and the impact of this process on software patch management. In order to understand why patch management is so important, it is necessary to consider the process of software engineering as a discipline and software enterprises as businesses with the primary goal of generating revenue. Due to the very nature of software being that it is designed by humans, there is an inherent risk of errors being introduced into software code. When this is combined with the pressure to release software to market as quickly as possible, the likelihood that software defects exist is increased exponentially.

3.2. DEFECTS IN THE SOFTWARE DEVELOPMENT PROCESS

It is noted by Harris (2013:1085) that "programming code is complex – the code itself, routine interaction, global and local variables, input received from other programs, output fed to different applications, attempts to envision future user inputs, calculations, and restrictions form a long list of possible negative security consequences." This underlies a fundamental problem with creating software that functions perfectly and has no security loopholes. Harris (2013:1085) also points out that "as you limit the functionality and scope of an application, the market share and potential profitability of that program could be reduced. A balancing act always exists between functionality and security, and in the development world, functionality is usually deemed the most important". Programmers and application architects need to strike a balance between the functionality of the program and ensuring that security requirements are implemented.

Ahmad and Varshney (2012:9) concur with Harris by noting that software defects have a major impact on the software development life cycle. Software defects are expensive to detect and correct, and it is estimated that the cost of finding and correcting defects represents one of the most expensive software development

activities. This view is echoed by Jones (2010:555), who states that historically, large software projects have spent more time and effort on finding and fixing bugs than on any other activity. This is because software defect removal efficiency only averages around 85 percent, with the major costs of software maintenance being finding and fixing bugs accidentally released to customers.

As a result, there is an indication that for the foreseeable future, it will not be possible to eliminate all defects. While this may bring dissatisfaction to many customers, who are likely to be saddled with buggy and insecure software, Ahmad and Varshney (2012:9) suggest a silver lining of sorts, noting that while defects may be inevitable, it may be possible to minimise their number and impact on projects. The proposed solution to this is to ensure that development teams implement a suitable defect management process that focusses on preventing or catching defects as early in the process as possible, thereby minimising their impact.

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.

In 2008 and 2009, a study was performed that identified the 25 most common and serious software bugs or defects. The study was sponsored by the SANS Institute, with the cooperation of MITRE and about 30 other organisations. In spite of the fact that software engineering is now a major occupation and millions of applications have been coded, only recently has there been a serious and concentrated effort to understand the nature of bugs and defects that exist in source code. The SANS (2009) report is significant because the compilation of the list of the 25 serious problems identified in the report was facilitated by a group of some 40 experts from major software organisations. As a result, it is claimed that the problems cited are

universal programming problems and not simply issues for a single company (Jones, 2010:509).

It is noted that, as of 2009, these 25 problems may occur in more than 85 percent of all operational software applications. Furthermore, it is asserted that one or more of these 25 problems can be cited in more than 95 percent of all successful malware attacks (Jones, 2010:511). Given the complexity and pervasiveness of these 25 potential problems, it is easy to see why being able to design secure software that is immune to all these problems is exceedingly difficult.

The IBM severity scale, as referenced by Jones (2010:571) has four levels for measuring the severity of any particular software flaw, ranging from a level 1, which is catastrophic, to a level 4, which is deemed minor:

- **Severity Level 1**: Is the most severe possible outcome, which results in software which does not operate at all;
- **Severity Level 2**: Where major software features may be unavailable or incorrectly applied;
- **Severity Level 3**: Which may have minor software features unavailable or incorrectly applied; and
- **Severity Level 4**: Is the least severe level, in which there may be cosmetic errors that do not negatively affect operation of the software.

Interestingly, the IBM severity scale seems to only consider the impact of functionality issues and does not address security-related issues that compromise the integrity of the application and its associated data. This highlights the gaps relating to security – software functionality is what is deemed important, because this is what the customer experiences when interacting with the software. Security in this case seems to take on a lower significance.

3.3. 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).

There are typically three levels of testing used for software development projects, namely:

Unit testing: Sommerville (2011:211) defines unit testing as the process of testing program components, such as methods or object classes. Individual functions or methods are the simplest type of component. Dooley (2011:193) concurs and elaborates further to explain that with unit testing, the individual methods and classes are tested and not the larger configurations of the program.

Integration or component testing: Sommerville (2011:216) notes that software components are often made up of several interacting objects. Integration testing involves the testing of the various units that make up the complete component. Dooley (2011:193) notes that integration testing is normally done by a separate testing division. This is the testing of a collection of classes or modules that interact with each other; its purpose is to test interfaces between modules or classes and the interactions between the modules. Testers write their tests with knowledge of the interfaces, but not with information about how each module has been implemented. Integration testing can only be performed after unit tested code is integrated into the source code base.

System testing: System testing involves the integration of all components to create a full working version of the system, which can then be tested in its entirety. System testing verifies that all the individual components are

compatible, interact correctly and transfer the right data at the right time across their interfaces (Sommerville, 2011:219). Dooley (2011:194) notes that system testing is usually performed by a separate testing division. This is the testing of the entire program (the system). System testing is done on both internal baselines of the software product and on the final baseline that is proposed for release to customers. System testing can also include stress testing, usability testing, and acceptance testing. It is further noted that end users may also be involved in this type of testing in order to verify that the system performs per the user's requirements.

3.4. THE PROBLEM WITH TESTING

Software can be extremely complex. As functionality and features increase, so does the level of complexity. Dooley (2011:194) questions "if we can use testing to find errors in our programs, why don't we find all of them? After all, we wrote the program, or at least the fix or new feature we just added, so we must understand what we just wrote." It is generally the complexity of programs that makes the task of testing more challenging. There are two main reasons noted 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. It is also noted that reading static programs does not help to identify any dynamic interactions between modules and interfaces, and as a result, a combined approach may be needed to ensure appropriate testing. Both static (code reading) and dynamic (testing) techniques are required to find and fix errors in programs (Dooley, 2011:194). 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.

While these noted difficulties in testing are problematic, Sommerville (2011:207) introduces the concept of a verification and validation process, which seeks to establish confidence that the software system is 'fit for purpose'. This requires that the system be good enough for its intended use. It is further explained that the level of required confidence depends on the system's purpose, the expectations of the system users, and the current marketing environment for the system. There are three main factors that will impact on the required level of confidence, namely:

Software purpose: The more critical the software, the more important it is for it to be reliable. For example, the level of confidence required for software used to control a safety-critical system is much higher than that required for a prototype that has been developed to demonstrate new product ideas (Sommerville, 2011:207).

User expectations: Because of their experiences with buggy, unreliable software, many users have low expectations of software quality. They are not surprised when their software fails. When a new system is installed, users may tolerate failures because the benefits of use outweigh the costs of failure recovery. However, as software matures, users expect it to become more reliable, so more thorough testing of later versions may be required (Sommerville, 2011:207).

Marketing environment: When a system is marketed, the sellers of the system must take into account competing products, the price that customers are willing to pay for a system, and the required schedule for delivering that system. In a competitive environment, a software company may decide to release a program before it has been fully tested and debugged because they want to be the first into the market. If a software product is very cheap, users may be willing to tolerate a lower level of reliability (Sommerville, 2011:208).

3.5. SOFTWARE PATCHES

As has been noted earlier in this chapter, 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."

The importance and prevalence of security flaws as a software problem is highlighted by the CERT Coordination Center, which indicates that reported software vulnerabilities have grown from 171 in 1995 to 8,064 in 2006 (Meyer and Lambert, 2007:1). This suggests that software users must be vigilant in protecting their information systems. It is noted that an efficient system for implementing software security patches is critical, since the time it takes for a reported vulnerability to be exploited is decreasing to less than 24 hours in some cases. It is also noted that there is a recent trend of zero-day exploits, in which a vulnerability is exploited before or on the same day as the vulnerability becomes known to the software developer.

A patch management system, according to Stiennon (2012:8) seeks to ensure that every application has the latest software patches installed. The aim is to ensure that no software vulnerabilities exist in the software used; however, this is not always possible. In the event that software vulnerabilities cannot be eliminated, the total exposure to new vulnerabilities must be minimised as far as possible. However, regarding the feasibility of this concept, Stiennon (2012:8) notes that "organisations spend an inordinate amount of time and money on these protections and still they succumb to targeted attacks which use previously unknown vulnerabilities". This may be demotivating and emphasises the large amount of effort required to simply provide a reasonable level of protection against security breaches.

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). It is noted that patching may be a risky operation for a number of reasons. For example, 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. When sufficient patch testing and planning is not done, the “patch and pray dilemma” invariably appears (Taylor et al, 2005:18). Both availability and security are noted as critical elements in any consideration of a patch deployment. That is, the impact of not deploying a patch (and thereby running a security risk) or deploying a patch without sufficient testing (potentially resulting in an availability issue) need to be considered. Taylor et al (2005) note that high-performing IT organisations often patch far less frequently than other IT organisations, and yet they still achieve their desired security posture. It is claimed by Taylor et al (2005:19) that such high-performing IT organisations are able to effectively manage residual risk and use compensating controls instead of patching, which is another important consideration when deciding whether or not to patch.

As a guide, Taylor et al (2005:19) identify a number of questions that need to be evaluated before an organisation attempts to apply any patch to a production system:

- Is this a material threat to our ability to deliver safe and reliable service to the business?
- Can we mitigate this threat without applying the patch or update?
- Can we test the impact of the update and feel confident that our tests will predict the outcome on our production systems?
- When is the next release cycle? Can we package this update with other tested updates?
- If we have to do this now, how can we minimize the risk of unexpected consequences?
- If we cannot reduce the risk of exposure through testing, and we cannot bundle this with any other releases, then can we get the stakeholders and IT management to sign off on the risk?

3.6. AUDITING AND PATCH MANAGEMENT

Auditors, both internal and external, should be aware of how their companies and clients are managing the patch process, because it is a key control in securing a company's data, including financial data. This is the view on the role of auditors given by Meyer and Lambert (2007:1). It is also noted that 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 increased the emphasis on the responsibility of auditors for assessing risk. Trent (2004) notes that information-system security is critical for all organisations, and its main component is patch management. Meyer and Lambert (2007:1) discuss the financial aspect by noting that: "Patch management is essential to the proper functioning of the financial reporting process, and auditors should recognize its importance because corrupted financial data can cripple an organization." These views on patch management and the increasing focus on risk and due professional care seek to catalyse the importance of this topic to auditors. This will require heightened attention during audits in future, as well as a detailed understanding of the topic by the audit team to ensure that an appropriate audit opinion can be reached.

3.7. CONCLUSION

Historically, there was no consideration for information security built into the software design process. It may not have been a problem, seeing as many systems were not networked and very few individuals understood how these systems operated, so there was a very low probability for security attacks. As time has progressed and systems have become more networked and connected to public networks like the Internet, information security has become more of a concern. Further to this, there are now many more individuals with access to computers, and with time available and an inquisitive mind these can become new hackers.

In terms of the commercial aspect of creating software, there is now much more competition, and a fast entry to the market can make the difference between a product's success and failure. As a result, development timeframes are being reduced to gain a faster speed to market, but the result of this speed has seen not

only security being neglected, but testing also being placed under strain. This has all contributed to software being released to the market with a significant number of flaws or bugs. The commercial model suggests that it is more beneficial to release a product to the market with possible flaws and to release patches as and when these issues are detected. In effect, the software companies are transferring some of their testing effort to their users, which saves them both time and money.

It can be argued that any approach to software testing may be inherently flawed, and it is impossible for a vendor to thoroughly test software before it is released to the market, as the cost and time requirements would be prohibitive. The deficiencies in software design, testing and commercial pressures mean that software patch management is certainly here to stay for the foreseeable future. In the following chapter, the processes that are recommended in implementing a successful patching initiative are investigated and reviewed.

CHAPTER 4
THE PROCESS OF SOFTWARE PATCHING AND THE AUDITING THEREOF

4.1. INTRODUCTION

The previous chapter highlighted that the need to deploy software patches may be inevitable for most computer software for the foreseeable future. The criticality of this process is highlighted by Meyer and Lambert (2007), who note that it is possible that many Internet-connected information systems will be subject to constant attack. While this might sound rather alarmist, figures to substantiate this claim can be found: the United Kingdom experienced 44 million cyber attacks in 2011 (Whitehead, 2013:1), while the United States military noted an alarming 10 million cyber attacks per day in 2012 (Fung, 2013:1). Even private companies are not exempt, with British Petroleum CEO Bob Dudley indicating that his company suffers 50,000 cyber attacks a day (CNBC, 2013:1).

The importance of patch management is highlighted by Meyer and Lambert (2007:7), who note that it is a key control in any information system. Patch management can be linked to SOX requirements in that documentation and internal testing is required to be evaluated as part of financial statement and SOX section 404 audits. The suggestion is that patch management should be considered as part of a larger enterprise risk management (hereafter ERM) system, which would provide valuable input on event identification, risk assessment and risk response. Meyer and Lambert (2007:7) conclude that “without a systematic process of patch management, companies are essentially leaving the doors to their information systems wide open to potential troublemakers.” It is clear that a formalised and efficient process for deploying software fixes or updates, commonly referred to as software patches, is required for any organisation to be effective at mitigating their risk in this area. In order for auditors to provide assurance on the patching process, the approach taken will need to be assessed. The required processes will be further investigated throughout this chapter.

4.2. THE PATCH MANAGEMENT PROCESS

According to Sommerville (2011:243), there are three distinct types of software maintenance operations that are typically facilitated by means of software updates or patches. The first, and most common, are fault repairs, involving coding errors that are noted as being relatively cheap to correct; design errors, which are more expensive, as they may involve rewriting several program components; and requirements errors, occurring when the initial software requirements were not properly understood or implemented. Such requirement errors are identified as being the most expensive to repair, because of the extensive system redesign that may be necessary. Second is environmental adaptation, where software maintenance is required when some aspect of the system's environment (such as the hardware, operating system or other support software) is subjected to a change that impacts the application. As a result, the application must be modified to adapt it to cope with these environmental changes. Finally, there is functionality addition, which may not necessarily result in any deficiency in the original software, but is triggered when the system requirements change as a result of an organisational or business change. It is noted that such a change is generally of a much greater scale than for the other types of maintenance (Sommerville, 2011:243).

4.2.1. PREREQUISITES FOR A SOFTWARE PATCHING PROCESS

The deployment of software patches, as noted, is an important task. As a result, a formalised and systematic approach is required to ensure the best possible outcome. An initial assessment of the computing environment may be necessary before embarking on the development of a patching programme. Trent (2004:22) notes that the level of security knowledge of the IT staff that will be performing the patching process is an important consideration to determine whether suitable skills exist within the organisation. Secondly, the amount of resources required to ensure that the task of patching can be effectively executed needs to be understood, as in certain organisations it may be necessary to have dedicated individuals or teams to perform this process, depending on the size of the organisation. Finally, a determination needs to be made as to whether the IT infrastructure is suitably configured to allow an automated patching process to take place, or whether there will need to be a

large amount of manual intervention by IT technicians to perform the patching process.

When embarking on the establishment of a formal patch management process, a number of prerequisites may be required before the actual patching process can commence. The following dependencies are noted from various sources in the literature:

End user education and training: A prerequisite for implementing a patch management process is to determine the level of expertise of the end user population. Dependent on the level of user education, appropriate communication will need to be developed to inform end users of the process and their responsibly during the process. Users will also need to be made aware of the patching policies and procedures implemented by the organisation. Without appropriate user knowledge and awareness, the patching process is unlikely to be successful (Trent, 2004:23). Meyer and Lambert (2007:3) concur and also note that there should be consequences if users do not abide by the company policies and procedures with regard to the patching of their computers.

The formal assigning of responsibilities: It is recommended that formal assignment of responsibilities for patch management be made (Trent, 2004:23). Meyer and Lambert (2007:2) also recommend that a team should be assembled to oversee patch management. It is suggested that this team should be multidisciplinary, in that it should include not only individuals from IT (who are responsible for implementation) but also a liaison to the board of directors, internal auditors and corporate accountants as part of the team. The representatives from the IT department should have specific knowledge in system administration, system security, operating systems, and any hardware or software used within the organisation. This team will be responsible for working quickly to identify vulnerabilities, understanding the organisation's exposure, determining the necessity of installing any available patch, testing the patch to ensure that it will not interfere with any existing systems, and implementing the patch by following appropriate change control.

The development of a chain of communication: In order to ensure the success of any patching process, the importance of effective communication to everyone involved in the process is noted, Trent (2004:24) suggests that there are three groups that need to be taken into account when communicating the pending deployment of a patch. These are the patching team that will be deploying the patches; the company's business management, in order to ensure support; and the end users, who need to be aware of the fact that the computers they are using are company property and that failure to patch will put company information at risk.

The setting of baselines for computer systems: A baseline is identified as a standard set of software programs that individuals have available on their computers. This baseline should include the most current versions of programs installed with all vendor-released and tested patches (Meyer & Lambert, 2007:2). Furthermore, Trent (2004:25) notes: "Adhering to your baseline is important because a single, unpatched, nonstandard computer can make an entire environment vulnerable to attack."

The need for management support: It is suggested that one of the most important aspects of any patch management policy is the need for appropriate senior management support. It is important to ensure that management is included in all aspects of the patch management planning and policy development. Trent (2004:26) notes that "if the end users know that management is behind you in your quest to secure the environment, the end users will be less likely to challenge the methods you utilize to accomplish the task." Meyer and Lambert (2007:2) further suggest that there should be not only general management support, but also that of the board of directors and upper management. An important consideration may also be the willingness to fund and otherwise assist in the patch management process. The consequence of a lack of management support is noted to be a patch management process that will fail to achieve the level of security expected.

A risk-based approach to patching: As has already been seen, it is important to always have a risk-based view to ensure that the appropriate focus is placed on patching and to ensure system uptime. Meyer and Lambert

identify two types of risk that are of a primary concern when setting priorities for patching: organisational risk (if the application is not available), and exposure risk (if the software is exposed to the Internet or an external network). When dealing with these two very different risk types, it is suggested that firstly, software should be prioritised in terms of which programs would present the greatest risk to an organisation if they were not available. It is then recommended that the highest-priority applications should be tested to ensure that the installation of the patch will not disrupt the software's functioning. Furthermore, a high priority should be given to those applications that are most at risk of attack because they are exposed to the Internet or other offsite access (web browsers and e-mail applications), as well as security software, including firewall and virus protection (Meyer & Lambert, 2007:2).

The need for automated patch deployment software: An important aspect of any patch management process is how the patches are managed and deployed. Without formal patching software it would be difficult and time consuming to deploy all patches manually to the affected systems (servers, desktop computers, laptops, etc.). A patch management software package will automate the deployment of patches and thus ensure a greater overall success in the patch deployment process (Meyer & Lambert, 2007:3).

Patch management system metrics: The ability to measure the effectiveness of any patching process is necessary to identify weaknesses, so that improvements to the process can be made. NIST (as quoted by Meyer & Lambert, 2007:5) recommend the collection of three categories of metrics. Firstly, the susceptibility to attack, which includes the number of patches and the number of vulnerabilities. Secondly, the mitigation response time, which includes response time for vulnerability and patch identification, as well as patch deployment. Finally, cost metrics, such as the patch management group, system administrator support, patch software, and system failure costs.

Formalised policies and procedures: As patch management is an important internal control, an organisation should fully document its policies and procedures. The policies should identify the obligations of the patch

management group, the expected level of cooperation by system administrators related to security issues, and the hierarchy of software/hardware. Procedures should spell out exactly what is to be done to identify vulnerabilities, test patches, deploy and install patches, and determine patch compliance. The procedures should also require that all steps in the patch management process be documented to facilitate the auditing of this important internal control (Meyer & Lambert, 2007:5).

The development and maintenance of an IT asset inventory: A key step in the development of a comprehensive patch management process involves creating a detailed inventory of all IT assets, including all hardware and software. The inventory serves as a means to identify which systems need patches. In order to ensure ongoing success, it is critical to ensure that the IT asset inventory is kept regularly updated. Some additional benefits may also be derived from this process of identifying and recording IT assets, such as financial reporting, tax reporting, data security and asset disposal management (Meyer & Lambert, 2007:4).

The creation of a contingency plan for zero-day attacks: In the area of IT security, the worst-case scenario is a zero-day vulnerability attack on critical systems. Organisations should have plans in place to determine what they will do if a vendor patch is not available and their critical systems are vulnerable to an imminent attack. It is noted that planning for an eventual zero-day attack is prudent and will reduce the panic that often accompanies such an attack (Meyer & Lambert, 2007:6).

The IIA (2005:5) note that when auditors are assessing the process of software patching, it is important to note whether it is part of a defined, repeatable change-and-release process or if it is ad hoc, informal, and emergency based. The more formalised and controlled the process is, the greater the level of assurance can be provided in terms of the process.

4.2.2. EXECUTION OF SOFTWARE PATCHING

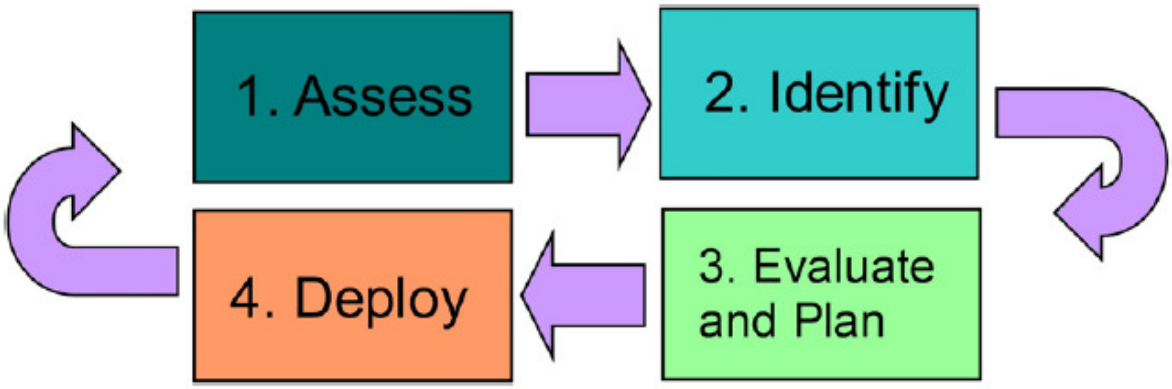
In the literature, it is generally accepted that any effective patching process should follow a number of predetermined steps. Various authors provide a differing number

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 diagram below graphically depicts the suggested process based on the literature. The details within these phases will now be further investigated by assessing the approach taken by various authors.

Figure 4.1: Software patching process



Source: Trent (2004:27)

Assessment phase

The patch management process begins with an assessment of the organisation's production environment, including the security threats and vulnerabilities present and whether the organisation is able to suitably deploy new software updates. Trent (2004:8) also notes that the patching process may begin before any new patch is released. This step involves the organisation's preparation before embarking on any potential deployments. The gathering of information about the IT environment is required in order to successfully deploy software patches. Andrew (2003:2) provides further insight by suggesting particular research that may be needed during this initial process. It is noted that there may be a necessity to investigate, assess the impact, review application dependencies, identify targets, and assess hardware and software requirements during this phase. Furthermore, Roberge (2004:7) broadly concurs with Trent and Andrew and adds that the discovery phase may also involve locating assets (workstations and servers) on the company network and categorising them.

Identification phase

During the identification phase, the goal is to discover any new software updates in a reliable manner and then to determine whether those software updates are relevant to the organisation's production IT environment. Thereafter, a determination should be made as to whether the update would require a normal deployment process or emergency deployment. Trent (2004:29) suggests that the most appropriate method

to identify new software updates may be to sign up to vendor security bulletins, such as Microsoft's TechNet security bulletin in the case of Microsoft products. Roberge (2004:7) recommends that through an analysis process, the current patch levels must first be determined and a minimum baseline policy should be defined.

Trent (2004:31) notes that a large part of patch evaluation is determining the importance of deploying a particular patch, while Meyer and Lambert (2007:5) also highlight the importance of assessing the impact of any new vulnerabilities. It is suggested that once a vulnerability is identified, the patch management team should assess the potential impact an exploitation of this vulnerability would have on the IT systems. Subsequent to the impact assessment, a determination may be made as to whether the patch should be deployed. It is noted that it may not always be necessary to deploy every available patch. If the problem that a patch addresses is of an insignificant nature or the targeted system is of a low importance, it may be suitable to forego the deployment of that particular patch. The following factors are mentioned by Meyer and Lambert (2007:5) as being important when considering whether a patch should be deployed:

- The type and delivery of possible attack;
- Severity of the vulnerability; and
- Criticality of the system.

In the event that there is no available patch or a patch cannot be deployed in a suitable timeframe, it may be necessary to disable a particular resource or function of the system until such time as a patch can be applied. It is also necessary to consider possible compensating controls (such as firewalls) during the assessment. These compensating controls may also negate the need to deploy a patch (Meyer & Lambert, 2007:5). In addition, Trent (2004:31) notes that while some patches may only apply to a limited area of the IT environment, others may affect the entire organisation.

It is suggested that the two most significant considerations during this phase are cost and risk. Types of costs relating to patching may include: costs for manpower, costs to the organisation, and planned downtime. On the other side of the cost equation are costs associated with not applying patches. These costs typically include investigative and corrective work to restore systems and unplanned downtime. The

cost of updating software could be compared to that of not updating it, and a resulting cost threshold for a 'go/no-go' decision could be set. However, cost is just one factor – risk must also be considered. If the risk of unplanned downtime is high and the cost is not prohibitive, the software should generally be updated. If the risk is low, the cost too high, or both are true, the software should generally not be updated (Sun Microsystems, 2004:17).

There are various risks to applying a patch, as well as risks to not applying one. The risk of applying a patch may be the possibility of introducing unexpected consequences as a result of the change. The risk of not applying a patch may be the possibility of not being able to use an application that the patch fixes. The decision to update software should be based on sound data and the requirements particular to the software installation under consideration (Sun Microsystems, 2004:17). This conundrum of weighing up the risk of security vs. availability is at the heart of any successful patching process.

Evaluation and planning phase

It is suggested that the typical goals during the evaluation and planning phase should include:

- A go/no-go decision as to whether to deploy the software update (Trent, 2004:31), which may also include a risk analysis for all missing patches Roberge (2004:7)
- A determination of what resources are needed to deploy the update. (Trent, 2004:31; Andrew, 2003:2)
- Testing of the software update in a testing environment that mirrors the actual production environment in order to confirm that the update does not compromise business critical systems and applications (Trent, 2004:31). Furthermore, Andrew (2003:2) states that this testing should include the development of a test plan. Meyer and Lambert (2007:4) emphasise the importance of testing by noting that a patch may have an unwanted negative consequence and could cause a system to fail. It is also noted that the importance of testing patches is often ignored in the pressure to patch systems when a known exploit exists for a particular high-risk vulnerability. The United States General Accounting Office (2004:20) note that examples of unintended consequences include patches that force other

applications to shut down and patches that undo the effects of previously applied patches. Andrew (2003:2) also suggests that the testing process may be repeated in a number of iterations.

Deployment phase

The goal during the deployment phase is to successfully roll out the approved and tested software updates into the production environment in a manner that there is no disruption to the required IT service levels. Trent (2004:32) proposes three distinct activities during this patch deployment phase:

• Deployment preparation;

• Deployment of the patch to targeted computers; and

• Post-implementation review.

Roberge (2004:7) refers to this phase as remediation, as its purpose is to 'remedy' the vulnerabilities found by bringing systems up to date. While appropriate testing should have already taken place before deployment, there may still be a possibility of problems arising following the deployment of the patch into the production environment. For such an eventuality, a 'safety net' is recommended, which may entail the ability to roll back (undo) a patch should the need arise.

Meyer and Lambert (2007:5) also suggest a verification of all patch installations. This is seen as important in determining whether all deployed patches have been successfully applied. After completing the assessment of patch success or failure, it will be necessary to update the IT asset inventory, which will then highlight if there are any systems that have not yet been patched. This verification process is considered vital in ensuring that all systems are appropriately patched, as security is only as strong as the weakest link and a single unpatched system could disrupt other systems.

A final step suggested by Roberge (2004:7) is that of reporting, which also encompasses Meyer and Lambert's verification step, but goes further to require that reporting should also include enough review, analysis, and adjustment to close the loop and begin the cycle again automatically.

As can be seen, the proposed enterprise approach is high-level, quite comprehensive and time-consuming, erring on the side of caution so as not to introduce instability into systems. This approach is can thus be viewed as extremely rigorous. In the next section further investigation will be performed to compare various deployment strategies for patches based on their severity ratings.

4.3. VENDOR PATCH SEVERITY RATINGS

In order to perform an appropriate assessment on the impact of any particular security patch released by a software vendor, it is necessary to perform a risk assessment (as mentioned earlier in this study). A critical input into this process is the potential impact of the exposure resulting from a successful attack on that particular software vulnerability. Performing such an assessment would be difficult for organisations, since in most cases software is purchased from a vendor and the requisite experience to determine a particular exposure would not be present within the organisation. With this in mind, Microsoft (2012:1) states that “not all vulnerabilities have equal impact.” This is the reason they have introduced the security bulletin severity rating system. This system, based on customer feedback, is intended to help their customers decide which updates they should apply under their particular circumstances, and how rapidly they need to take action. Microsoft states that “customers have encouraged us to include this information in our bulletins to help them assess their risk” (Microsoft, 2012:1).

Based on Microsoft’s industry experience, they claim that attacks that impact customers' systems are rarely a result of attackers' exploitation of previously unknown vulnerabilities. Attacks typically exploit vulnerabilities for which patches have long been available, but not applied (Microsoft, 2012:1). This view is shared by many industry and security experts, because it is relatively easy to reverse engineer a released vendor patch to identify a security weakness in comparison to performing exhaustive work to try to identify a new vulnerability. This is the reason why it is so important to ensure that high-risk patches are deployed as soon as possible after they are released to the public.

Microsoft’s severity rating system provides a rating for each vulnerability per component or platform. This rating represents the worst theoretical outcome if the vulnerability is exploited on a given component or platform. However, the severity

rating does not indicate the likelihood of that outcome (Microsoft, 2012:1). This seems to be a serious flaw, given that to effectively assess risk the likelihood needs to be considered. To that end, Microsoft introduced the Microsoft Exploitability Index, which is designed to provide additional information to help customers better prioritise the deployment of Microsoft security updates.

The Microsoft Exploitability Index helps customers prioritise security bulletin deployment by providing information on the likelihood that a vulnerability addressed in a Microsoft security update will be exploited within the first 30 days of the update's release (Microsoft, 2008:1). The Exploitability Index uses one of three values to communicate to customers the likelihood of functioning exploit code having been developed, based on vulnerabilities addressed by Microsoft security bulletins. The levels are as follows:

Level 1: Exploit code likely

This rating indicates that Microsoft's analysis has shown that exploit code could be created in such a way that an attacker could consistently exploit that vulnerability. The result is that customers are at a high risk of attack based on this particular vulnerability. As a result, customers who have reviewed the security bulletin and determined its applicability within their environment should treat this with a higher priority (Microsoft, 2008:2).

Level 2: Exploit code would be difficult to build

This rating indicates that Microsoft's analysis has shown that exploit code could be created, but an attacker would likely have difficulty creating the necessary code, as it would require expertise and sophisticated timing, and/or it would have varied results when targeting the affected product. As a result, the likelihood of successful exploitation of this vulnerability is less than that of level 1. As a result, customers who have reviewed the security bulletin and determined its applicability within their environment should treat this as a material update. When prioritising against other highly exploitable vulnerabilities, this could rank lower in the deployment priority (Microsoft, 2008:2).

Level 3: Exploit code unlikely

This rating indicates that Microsoft's analysis shows that successfully functioning exploit code is unlikely to be released. This means that while it might be possible for exploit code to be released for the particular vulnerability, the likelihood of this is fairly low. Therefore, customers who have reviewed the security bulletin to determine its applicability within their environment could prioritise this update below other vulnerabilities within a release (Microsoft, 2008:3).

A further rating for the possibility of a denial of service (hereafter DoS) attack has also been included by Microsoft. A denial of service attack differs from a security breach in that instead of compromising confidential information and systems, it simply makes the systems unusable by the organisation or end user. This causes business disruption and customer frustration. This rating may be used to inform the impact value during the risk assessment.

Table 4.1: Microsoft DoS types

DoS Exploitability Assessment	Short Definition
Temporary	Exploitation of this vulnerability may cause the operating system or application to become temporarily unresponsive, until the attack is halted, or to exit unexpectedly but automatically recover. The target returns to the normal level of functionality shortly after the attack is finished.
Permanent	Exploitation of this vulnerability may cause the operating system or application to become permanently unresponsive, until it is restarted manually, or to exit unexpectedly without automatically recovering.

Source: Microsoft (2008:3)

While the exploitability index may help customers to assess the possible likelihood of a security issue, it may not take all relevant factors into account. Situational factors relevant to the client's unique environment may still need to be factored into any risk calculation. These may include, inter alia:

- The client risk profile to being targeted by an attack. PwC notes in their 2015 Global Information Security Survey that for financial services institutions, information security incidents are continuing to rise, as are the costs of these intrusions (PwC, 2015:1).
- Mitigating or compensating controls that have been implemented, such as application firewalls, network segmentation and application gateways (Meyer & Lambert, 2007:5).
- The risk tolerance level of an organisation to an attack. This will largely be determined by the type of business the client is involved in. For example, a bank would have a much lower tolerance for an attack than, for example, a mining operation (Collier, 2009:69).

Subsequent to collating all the required inputs to determine the overall risk rating of a particular patch, the next step would be to determine the deployment timeframe. Trent (2004:31) provides the following recommended deployment timeframes based on the risk rating:

Table 4.2: Patch ratings vs. recommended deployment timeframes

Classification (Risk Rating)	**Recommended Deployment Timeframe**
Critical	Start deployment within 24 hours
High	Start deployment within 1 week
Medium	Start deployment within 1 month, or opt for a service pack or update rollup
Low	Opt for a service pack or update rollup

Source: Trent (2004:31)

Microsoft advocate a similar approach, where it is suggested that by taking the exploitability index (mentioned earlier), a customer should be able to arrive at a recommended timeframe within which to deploy any particular patch. Microsoft does,

however, not provide the specific timeframes as Trent does, which could allow a large degree of interpretation by the client.

Table 4.3: Microsoft patch severity rating scale

Rating	Definition
Critical	A vulnerability whose exploitation could allow code execution without user interaction. These scenarios include self-propagating malware (e.g. network worms), or unavoidable common use scenarios where code execution occurs without warnings or prompts. This could mean browsing to a web page or opening email. **Microsoft recommends that customers apply Critical updates immediately.**
Important	A vulnerability whose exploitation could result in compromise of the confidentiality, integrity, or availability of user data, or of the integrity or availability of processing resources. These scenarios include common use scenarios where client is compromised with warnings or prompts regardless of the prompt's provenance, quality, or usability. Sequences of user actions that do not generate prompts or warnings are also covered. **Microsoft recommends that customers apply Important updates at the earliest opportunity.**
Moderate	Impact of the vulnerability is mitigated to a significant degree by factors such as authentication requirements or applicability only to non-default configurations. **Microsoft recommends that customers consider applying the security update.**
Low	Impact of the vulnerability is comprehensively mitigated by the characteristics of the affected component. **Microsoft recommends that customers evaluate whether to apply the security update to the affected systems.**

Source: Microsoft (2012:2)

As can be seen from the table above, Microsoft recommends that critical updates be applied “immediately”. However, this course of action may not be practical in many instances because it provides no time to properly plan for and test the patch before deployment. The requirement to apply a patch immediately is also in contrast to the recommended approaches indicated in Chapter 4 of this study, as well as Sun Microsystems’ (2004:17) statement that in respect of risk, “different companies have adopted various strategies for applying security-specific patches. Companies try to balance the need to avoid possible problems caused by applying a new, but potentially problematic, patch too quickly with exposure to a security flaw while waiting for the patch to demonstrate stability.” This further highlights the importance of an effective risk assessment before taking a decision on the timeframe within which to deploy a particular patch. To compound matters, it is noted that that the average time lag between the announcement of a security vulnerability and its exploitation by hackers is shrinking, thus putting pressure on the speed of deployment, even in highly controlled environments (Sun Microsystems, 2004:17).

4.4. AUDITING REQUIREMENTS FOR PATCH MANAGEMENT SYSTEMS

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 both corporate accounting and internal auditing, as well as external auditors, take an active role in the patch management system. It is suggested that there should 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 is often seen as purely an IT problem rather than as a key internal control protecting the financial information of a company. This is inaccurate, as corrupted financial data can cripple an organisation. “While the patch management system protects the entire information system, its role in protecting the financial information, especially where organizations employ ERP systems, is unassailable” (Meyer & Lambert, 2007:6).

The ISACA CobiT 4.1 framework defines a number of IT processes and controls that auditors should assess in their assignments. The “acquire and implement” domain provides details on audit requirements regarding the process of managing changes. The control objectives in this section require that standardised procedures be implemented for all changes to IT systems, and this includes the implementation of patches. This highlights the importance of a controlled process to deploy patches and ensure a successful outcome for this process (ISACA, 2007:94). The CobiT framework also mentions the importance of patching in the “delivery and support” domain, where it is noted that patching should form a key part of the process for protecting organisations against malicious software such as viruses, worms, spyware and spam (ISACA, 2007:118).

The evaluation of internal controls over financial information is of vital interest in the course of a financial statement audit, and is required by auditing standards. Because patch management is a key internal control, auditing the effectiveness of this control is necessary under the Sarbanes-Oxley Act (SOX) section 404. A lack of an adequate, functioning patch management system could be a material weakness in internal control, possibly resulting in an adverse section 404 opinion on internal control or a negative management assessment of internal controls. Auditors should be able to test whether these controls are working as designed. The key attributes that make a patch management system auditable include adequate documentation of policies and procedures, a significant set of meaningful metrics updated on a regular basis, and patch management software that can provide auditable information regarding the current status of any and all computers within an organisation (Meyer & Lambert, 2007:6).

4.5. CONCLUSION

From a hacking perspective, the concept known as a ‘zero-day vulnerability’ is the ‘Holy Grail’ for hackers. A zero-day (or zero-hour or day zero) attack or threat is an attack that exploits a previously unknown vulnerability in a computer system, meaning that the attack occurs on ‘day zero’ of the awareness of the vulnerability. As a result, the developers have not yet had any time to address the vulnerability and release a patch, as they do not yet know about the issue. Zero-day attacks can be very dangerous and hackers highly prize these types of vulnerabilities. There are

even websites on the Internet that sell zero-day vulnerabilities to the highest bidder. The days of hackers only being interested in gaining fame through their exploits are over, and the lines between hacking and organised crime are starting to blur. When a software vendor releases a patch to fix a serious security issue, it is generally easy for a novice hacker to reverse engineer what the patch is fixing. This would allow the hacker to determine what the initial security problem was and enable him to write code to exploit this vulnerability. It is in this scenario where the time to deploy a patch after release by the vendor is critical for organisations.

It is critical vulnerabilities that pose the greatest challenge to any patch management process, since they require urgent resolution. This makes following a formal change management and patch evaluation process more difficult, due to the time this type of patch takes to complete. From an audit perspective there is a requirement to deploy a patch in as fast a timeframe as possible, but only after having done appropriate testing in a test/development environment prior to deploying to a production environment.

Software vendors generally advocate a neutral stance and suggest that sufficient testing be done before a patch is deployed. They give ratings to their patches and generally recommend that 'critical' patches be deployed in a shorter timeframe. While some guidance is given, there may be little specification of recommended timeframes; it is thus the responsibility for each organisation to determine the timeframes for themselves. From a purely information security perspective, where the primary goal is to protect the confidentiality and integrity of systems, it is imperative to deploy patches as soon as possible after release by the vendor. However, this is often at odds with the requirement to perform sufficient testing.

CHAPTER 5
EMPIRICAL STUDY AND RESEARCH FINDINGS

5.1. INTRODUCTION

The objective of this study 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.

In Chapter 1, the importance of software patching in the current technological landscape was demonstrated. Chapter 2 reviewed the aspects of risk management relating to software and patch management. Risk was noted to be a critical driving factor that is to be considered throughout the organisation and a cornerstone of any successful patching strategy. In Chapter 3, the discipline of software development was considered. Since software patching is an activity closely related to software development, the various factors in software development that impact on software patching are assessed. Chapter 4 is the culmination of the literature study, in which the recommended processes of software patching are reviewed and assessed. The results of the literature study in Chapters 2 to 4 provided the basis for the aspects that were tested through the empirical study. The methodology followed for the empirical study consists of the analysis of a questionnaire sent to South Africa's largest financial institutions.

In this chapter, the approach followed in the empirical study will be explained and the findings will also be presented and discussed.

5.2. APPROACH TO THE EMPIRICAL STUDY

5.2.1. SELECTION OF THE 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. As was previously noted, 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.

5.2.2. METHOD APPLIED IN THE EMPIRICAL STUDY

5.2.2.1. DESIGN AND POPULATION

As stated above, the empirical study employed questionnaires that were sent to the head of internal auditing at each of the big four South African banks. The questionnaires were sent electronically to the relevant parties under a covering message from the research study supervisor (a copy of the covering letter is recorded as Annexure 1). Follow-up reminders were given after 10 days to all parties who had not yet responded.

The information in the questionnaire mainly covered the following areas (a copy of the questionnaire is recorded as Annexure 2):

- 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.
- Question 2 was aimed at identifying the patching considerations that are deemed to be important to the auditors within the big four banks.
- Question 3 seeks to gauge whether the auditors of the big four banks believe that the patching process within their organisation(s) are suitably risk focussed.
- Question 4 seeks to identify the factors that the auditors of the big four banks deem to be important when developing a software patching programme.
- Question 5 seeks identify the approach followed by the big four banks with regard to the testing of software patches.

- 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.

5.2.2.2. 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.

The questionnaire was sent to the chief audit executives of the four big South African banks on 25 May 2015, and follow-up was performed on a regular basis, with the final response being received on 8 June 2015.

5.2.2.3. LIMITATIONS TO THE SCOPE OF THE SURVEY

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, it was expected that this feedback would not necessarily represent other industries in South Africa or abroad. As previously mentioned, 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.

5.3. THE RESPONSE RATE

For the questionnaires, a 100% response rate was achieved. All of the questionnaires were completed and received.

Table 5.1: Response rate of questionnaires

	Questionnaire to big four South African banks	
	Number	**Percentage**
No response	0	0%
Completed and usable questionnaire	4	100%
Total percentage of questionnaires sent out	**4**	**100%**

5.4. RESEARCH FINDINGS

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

5.4.1. SOFTWARE PATCHING RISKS

i) 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.

ii) Findings

Table 5.2: Software patching risks

	Number	Percentage
Which of the following risks resulting from the process of software patching do you deem more significant?		
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)

In section 3.5 of Chapter 3 it was noted that there are two risk considerations relating to the deployment of software patches. The first is the security exposure threatened by a potential vulnerability present in the software which the patch is intended to correct and the second being the possibility of the patch causing unexpected consequences that may result in system downtime. 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.

5.4.2. DEPLOYMENT OF SOFTWARE PATCHES

i) Objective of the question

In Chapter 3 of the study, 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.

ii) Findings

Table 5.3: Deployment of software patches

	Total	Number			Percentage		
		To what extent			To what extent		
Which of the following factors do you deem to be important in assessing the need to deploy a software patch:		**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)

Various literature sources suggest a number of patching factors, as discussed in Chapter 3. 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 a 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.

5.4.3. SOFTWARE PATCHING RISK FOCUS

i) Objective of the question

Chapter 2 of the study deals with risk management, and it is noted that the assessment of risk should form the basis of any decision regarding the patching of software. Question 3 seeks to gauge whether the auditors of the big four banks believe that the patching process within their organisations are suitably risk focussed.

ii) Findings

Table 5.4: 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)

Chapter 2 of the study discusses the process of risk management, a fundamental imperative in ensuring successful business operations. The process of identifying, quantifying and then mitigating risk is an arduous task, and may often be neglected as a result. In this regard, 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 a comment received on the survey, which indicated the following:

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

5.4.4. SOFTWARE PATCHING PROGRAMME

i) Objective of the question

In the literature discussed in Chapter 4 of the study, 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.

ii) Findings

Table 5.5: Software patching programme

	Total	Number			Percentage		
		To what extent			To what extent		
Which of the following factors do you deem to be important when assessing a software patching programme:		**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)

As noted in section 4.2.1 of Chapter 4 of the study, based on the literature, there are 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.

5.4.5. TESTING OF SOFTWARE PATCHES

i) Objective of the question

Based on the literature discussed in Chapter 4 of this study, it is noted that there are two distinct classifications for software patches based on the results of a formal risk assessment (discussed in Chapter 2). The 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.

ii) Findings

Table 5.6: Testing of software patches

	Number	Percentage
Based on your audit assessments, are patches tested prior to deployment in a formal testing environment within your organisation?		
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)

In section 4.2.2 of Chapter 4, the literature notes that there are two deployment methods, namely a normal process and an emergency process, depending on the criticality of the patch. An emergency process typically aims at a faster deployment timeframe through a reduced level of testing and approvals. In the survey, however, 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.

5.4.6. PATCH DEPLOYMENT TIMEFRAMES

i) Objective of the question

Chapter 2 of the study highlights the importance of performing a thorough risk assessment 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.

ii) Findings

Table 5.7: Patch deployment timeframes

	Number	Percentage
Based on your audit assessments, which of the following does your organisation use to determine the target timeframes for deploying patches?		
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)

The process of undertaking a risk assessment, as discussed in Chapter 2, 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. This is especially true for a patch that is developed and supplied by an external vendor, as there may not be the requisite skills within the organisation to make an assessment on these factors. In such cases, reliance may need to be placed on the generic risk assessments provided by the software developer, who typically risk ranks the patches by criticality. 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.

Summative Findings

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.

5.5. CONCLUSION

The findings of the empirical study indicated that the major South African banks are generally aware of the risks posed by software patching and that most of the best practices cited by the literature are noted by the survey respondents as being important to consider as part of their audit assessments. One area that did not receive much focus, according to the responses, is user awareness and support for the patching process. This is a potential area that may provide further assurance on the patching process during an audit. As per the literature, users may serve as an additional control measure, specifically for the computers that they use on a daily basis. Such users may alert IT support in the event of known issues with patching on

their devices. The vast majority of respondents were of the view that the most significant risk is the security exposure, as opposed to the potential risk that prematurely deploying a patch could cause. This is probably due to the huge customer backlash and potential legal and regulatory scrutiny and fines that may result from a system's security being compromised. Finally, despite the fact that the respondents generally displayed a high level of risk and software patching maturity, they nevertheless all indicated that they believed there was room for improvement within their organisations with respect to the risk focus currently being placed on the patching process.

In the next chapter, the conclusions of the literature study as well as the empirical study are given, along with potential recommendations for further study.

CHAPTER 6
CONCLUSION

6.1. INTRODUCTION

In this chapter, the significant findings from the literature study contained in Chapters 2 to 4, as well as the significant findings from the empirical study in Chapter 5 will be summarised. Furthermore, potential areas for future research will also be identified.

6.2. DEDUCTIONS

6.2.1. FROM THE LITERATURE STUDY

The process of software patching involves a number of risks, such as the risk of system compromise by a hacker due to the patch not being applied in a timely manner, or the risk that the premature deployment of a patch may cause unexpected consequences or downtime for the organisation. Risk management as a discipline aims to provide organisations with a systematic method to identify, analyse, and then treat risk. It is therefore important for risk management principles to be considered during the development and execution of any patching process. As an auditor, it is vital to ensure that the appropriate risk management principles are embedded within the patching process to ensure a successful outcome. The patching process itself can be an extremely time intensive process, with a number of steps required.

The significant findings from the literature study are the following:

- The risks to computer software have increased exponentially in the last decade and show no signs of abating, due to the increasing interconnectivity between systems and the pervasiveness of system access across different devices and jurisdictions.
- Software patches are a consequence of imperfect coding practices, which result in flaws within applications that may be exploited by hackers to cause system downtime (known as a denial of service attack) or a system security compromise resulting in the theft of confidential information.
- The discipline of risk management has as its objective the aim of enabling organisations to better manage risks that they face. The process of software

patching has a number of risks, and as a result, a formal approach to the assessment of these risks is required.

- The outcome of risk ranking software patches to be deployed may not necessarily result in all available patches being applied. A low-risk patch or one where the risk can be more easily mitigated without a patch deployment are two examples cited in the literature.
- An effective patching process should typically consist of the following four phases:
    - The first phase involves the assessment of the organisation's environment for any potential security threats and vulnerabilities. This phase may also involve gathering information about the IT environment as well as identifying all relevant IT equipment such as workstations, laptops and servers.
    - The second phase involves identification of all relevant software patches that may be required in the organisation's IT environment. Following the identification of available patches, this phase also requires that the patches be reviewed based on their criticality for deployment.
    - The third phase is evaluation and planning, where the patches are considered prior to being deployed. This typically involves a determination of the resources required to deploy the patch, as well as the testing of each patch in a suitable testing environment in order to limit the risk of unintended consequences after deployment in the production environment.
    - The final phase is the deployment phase. This is where the patch(es) are released into the production environment. Also part of this phase is a review of whether the patching process has been successful and the potential roll-back of problematic patches.
- Software vendors typically risk rate their patches as a means to provide organisations with a metric to determine the urgency with which a particular patch should be deployed. It is, however, suggested that organisations perform their own risk ratings on patches, as the vendor ratings are generic in

nature and may not take into consideration how a particular application is used or configured.

6.2.2. FROM THE EMPIRICAL STUDY

Based on the literature study and the results from the feedback provided in the empirical study, the following conclusions have been noted:

- The majority of the internal audit respondents indicated that they were more concerned with the security risk posed by not deploying a patch than the consequences for availability that deploying a patch prematurely could have through system downtime. This underlines the huge reputational and financial impact a potential security breach could cause, especially for the banking sector. While system downtime is still a big concern, it is apparently less important.
- All of the internal audit respondents indicated that their organisations could possibly improve their risk management focus relating to software patching.
- Internal auditors at the big four banks indicated that they do not believe that the end user awareness of the patching process is an important consideration. This could be an area for improvement, as the literature suggests that end user awareness may benefit a patching process.
- All of the internal audit respondents indicated that their organisations make use of a combination of the risk ratings provided by the software vendor and internal risk assessments when considering the need to deploy any particular software patch. This is in line with the best practice recommendations given in the literature.
- All the big four banks emphasise the importance of testing software patches prior to deployment. This is evidenced by their requirements for thorough testing even on emergency patch deployments. Formal testing is a time-consuming process, and this could mean that emergency patches take longer to deploy, potentially leaving security vulnerabilities open for longer.
- There may be insufficient consideration given to whether the threat can be mitigated without applying a patch or update, as not all respondents indicated this to be an important consideration for them. The literature indicates that

compensating controls such as firewalls or disabling certain software features may be considered instead of a patch deployment.

6.3. AREAS FOR FURTHER RESEARCH

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.

6.4. 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.

BIBLIOGRAPHY

Ahmad, K & Varshney, N. (2012). On Minimizing Software Defects during New Product Development Using Enhanced Preventive Approach. *International Journal of Soft Computing and Engineering (IJSCE)*, ISSN:2231-2307, Volume 2, Issue 5, November 2012.

Andrew, C. (2003). Patch Management: An Effective Line of Defense for UNIX and Linux. *Information Systems Audit and Control Association Journal*, Volume 5, 2003.

Ashford, W. (2015). *Microsoft fixes bugs exploited to hack military and financial firms*. Available from: http://www.computerweekly.com/news/2240240189/Microsoft-fixes-bugs-exploited-to-hack-military-and-financial-firms (Accessed on 6 April 2015).

Banking Association of South Africa. (2012). *South African Banking Sector Overview*. The Banking Association South Africa. Johannesburg, South Africa.

Basirico, J. (2011). *Lessons of the Sony PlayStation hack*. Available from: http://www.scmagazine.com/lessons-of-the-sony-playstation-hack/article/207788 (Accessed on 13 October 2012).

CNBC. (2013). *BP Flights Off 50,000 Cyber-Attacks a Day: CEO*. Available from: http://hereisthecity.com/en-gb/2013/03/07/bp-fights-off-50000-cyber-attacks-a-day-ceo (Accessed 4 February 2015).

Coleman, T.S. (2011). *A Practical Guide to Risk Management*. The Research Foundation of CFA Institute. New York, NY, USA.

Collier, P.M. (2009). *Fundamentals of Risk Management for Accountants and Managers*. Elsevier Ltd. Great Britain.

Dhillon, G. & Backhouse, J. (1996). Risks in the Use of Information Technology Within Organizations. *International Journal of Information Management*, Volume 16, No 1:65-74.

Dooley, J. (2011). *Software Development and Professional Practice*. Springer Science & Business Media. New York, NY, USA.

Economics and Statistics Administration. (2011). *Exploring The Digital Nation: Computer and Internet use at home*. Available from: http://www.esa.doc.gov/sites/default/files/reports/documents/exploringthedigitalnation-computerandinternetuseathome.pdf (Accessed on 7 October 2013).

Economist Intelligence Unit. (2007). *Coming to grips with IT risk: A report from the Economist Intelligence Unit sponsored by SAP*. The Economist.

Fung, B. (2013). *How many cyberattacks hit the United States last year?* Available from: http://www.nextgov.com/cybersecurity/2013/03/how-many-cyberattacks-hit-united-states-last-year/61775/ (Accessed 4 February 2015).

GFI. (2012). *Security Patching Trends for Major Software Vendors*. Available from: http://www.gfi.com/blog/security-patching-trends-for-major-software-vendors (Accessed on 5 November 2014).

Gheorghe, M. (2010). Audit Methodology for IT Governance. *Informatica Economica*, Volume 14, No 1/2010:32-42.

Griffiths, P. (2005). *Risk-Based Auditing*. Gower Publishing Limited. Hants, England.

Harris, S. (2013). *CISSP Exam Guide 6th Edition*. McGraw Hill. New York, USA.

Hopkin, P. (2010). *Fundamentals of Risk Management*. Kogan Page Limited. London, United Kingdom.

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.

Institute of Internal Auditors (IIA). (2008). *GAIT for Business and IT Risk*. Institute of Internal Auditors. Florida, USA.

Information Systems Audit and Control Association (ISACA). (2007). CobiT 4.1. Information Systems Audit and Control Association. Rolling Meadows, IL, USA.

Information Systems Audit and Control Association (ISACA). (2009). *The Risk IT Framework*. Information Systems Audit and Control Association. Rolling Meadows, IL, USA.

International Organisation for Standardization (ISO). (2009). *ISO 31000:2009*. International Organisation for Standardization. Switzerland.

Jones, C. (2010). *Software Engineering Best Practices*. McGraw-Hill. New York, NY, USA.

Kindsight. (2014). *Kindsight Security Labs Malware Report – H1 2014*. Available from: http://resources.alcatel-lucent.com/?cid=180437. (Accessed on 25 September 2014).

le Roux, A.J. & van Waveren, C.C. (2008). Risk categorisation of products used for the construction of electricity distribution assets. *South African Journal of Industrial Engineering*, Volume 19(2):77-91.

McGregor, J. (2014). *The Top 5 Most Brutal Cyber Attacks of 2014 So Far*. Available from: http://www.forbes.com/sites/jaymcgregor/2014/07/28/the-top-5-most-brutal-cyber-attacks-of-2014-so-far/ (Accessed on 30 October 2014).

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).

Microsoft (2008). *Microsoft Exploitability Index*. Available from: http://technet.microsoft.com/en-us/security/cc998259 (Accessed on 9 September 2013).

Microsoft. (2012). *Security Bulletin Severity Rating System*. Available from: http://technet.microsoft.com/en-us/security/gg309177.aspx (Accessed on 21 May 2013).

National Institute of Standards and Technology (NIST). (2005). *Creating a Patch and Vulnerability Management Program*. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology. Gaithersburg, MD.

Office for National Statistics (2012). *Internet Access – Households and Individuals, 2012*. Available from: http://www.ons.gov.uk/ons/dcp171778_275775.pdf (Accessed on 7 October 2013).

Poeter, D. (2013). *Adobe Hacked, Data for Millions of Customers Stolen*. Available from: http://www.pcmag.com/article2/0,2817,2425215,00.asp (Accessed on 13 October 2013).

Pressman, R.S. (2010). *Software Engineering:A Practitioner's Approach, Seventh Edition*. McGraw-Hill. New York, NY, USA.

PriceWaterHouseCoopers (PwC). (2012). *Aligning internal audit: Are you on the right floor?* PriceWaterHouseCoopers.

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).

PriceWaterHouseCoopers (PwC). (2015). *Global State of Information Security Survey 2015*. Available from: http://www.pwc.com/gx/en/consulting-services/information-security-survey/download.jhtml (Accessed 28 February 2015).

PSN Infrastructure Security & Cyber Defence Team. (2012). *Technical Standard: Common Standard for Patch Management: Public Services Network Programme*. Available from: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/81647/Patch_Mgt_Std_v1-0.pdf (Accessed on 23 May 2013).

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.

Roberge, C. (2004). *Patch Management Best Practises*. Ecora Software Corporation Available from http://www.cressidatechnology.com/pdfs/whitepapers/bestpractices.pdf (Accessed on 22 October 2014).

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).

Sommerville, I. (2011). *Software Engineering, Ninth Edition*. Addison-Wesley. Boston, Massachusetts.

Stanton, A. & Bradley, S. (2005). Patch Management Best Practises: A "How to" Guide for Securing the Enterprise. Available from: http://www.ecora.com/ecora/ebook/cpm/TheCompletePatchManagementBook.pdf (Accessed 23 May 2013).

Statistics South Africa. (2012). *Census 2011:P0301.4*. Available from: http://www.statssa.gov.za/Publications/P03014/P030142011.pdf (Accessed on 7 October 2013).

Shimonski, R.J. (2004). *Threats and your Assets – What is really at Risk?* Available from: http://www.windowsecurity.com/articles-tutorials/misc_network_security/Threats-Assets.html (Accessed on 23 July 2014).

Stiennon, R. (2012). Why risk management fails in IT. *Network World, Proquest Business Collection*, Volume 29:8.

Sun Microsystems. (2004). *Solaris Patch Management: Recommended Strategy*. Sun Microsystems. Santa Clara, CA, USA.

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.

Trent, R. (2004). *The Administrator Shortcut Guide to Patch Management*. New Boundary Technologies. Realtimepublishers.com

United States General Accounting Office. (2004). *Information Security: Continued Action Needed to Improve Software Patch Management*. United States General Accounting Office. Washington DC, USA.

Wang, A. (2013). *What Microsoft Didn't Say on Patch Tuesday*. Available from: http://securitywatch.pcmag.com/security/316773-what-microsoft-didn-t-say-on-patch-tuesday (Accessed on 13 October 2013).

Westerman, G. & Hunter, R. (2007). IT Risk: Turning Business Threats into Competitive Advantage. Harvard Business School Press. Boston, Massachusetts.

Whitehead, T. (2013). *Britain vulnerable from cyber attacks for at least 20 years*. Available from: http://www.telegraph.co.uk/news/uknews/law-and-order/9863041/Britain-vulnerable-from-cyber-attacks-for-at-least-20-years.html (Accessed 4 February 2015).

Wikipedia. (2015). *Big Four Banking*. Available from: http://en.wikipedia.org/wiki/Big_Four_(banking) (Accessed on 8 March 2015).

Wikipedia. (2014). *Usage share of operating systems*. Available from: http://en.wikipedia.org/wiki/Usage_share_of_operating_systems (Accessed on 5 November 2014).

ANNEXURE 1: COVERING LETTER FROM THE STUDY SUPERVISOR

20 May 2015

Dear Chief Audit Executive

Vulnerabilities in operating system and application software can expose business operations to data manipulation, theft and disruption of business activities. In order to reduce these risks, an effective software patching process must be implemented. Your opinion is essential in providing feedback on the current approach to the patching of operating systems and applications within the banking sector.

The research is being undertaken by Deon Oosthuizen, who is a senior audit manager at FirstRand, under the supervision of Professor Ben Marx. It forms part of a master's study on the approaches to software patch management which are needed to ensure that organisations protect themselves against security vulnerabilities and also reduce system downtime to a minimum.

This short questionnaire should not take longer than five minutes to complete, and your response as part of the population of the big four South African banks is critical to the success of the research. Your input will be of immense value. All information will be treated as confidential and will only be used to produce aggregate results. If you have any objection to completing the questionnaire, please state your reason and return the questionnaire for control purposes.

We thank you in anticipation of your co-operation.

Professor Ben Marx
B Compt, B Compt (Hons), M Compt, D Com (Auditing), CA (SA), ACCA (UK)
Study Leader

ANNEXURE 2: QUESTIONNAIRE: TO INTERNAL AUDIT AT THE BIG FOUR SOUTH AFRICAN BANKS

INSTRUCTIONS:	
1.	This questionnaire can be completed either electronically or manually.
2.	For electronic completion:
	a. Mark your answer in the appropriate box with a cross (x) [by clicking on the appropriate box with your mouse] or type in the relevant information as requested. If you need to navigate between blocks, use the TAB function or cursor arrows.
	b. Should you wish to change your answer in any of the check blocks, simply click on that box again to clear it, and re-select your answer.
	c. Once completed, **please save the file** and send the attachment to **deon.oosthuizen@firstrand.co.za**
3.	For manual completion:
	a. Print and complete the questionnaire. Please mark your answer in the appropriate box with a cross (x) or supply the relevant information as requested.
	b. Please return the completed questionnaire by fax or by emailing the scanned document: **e-mail: deon.oosthuizen@firstrand.co.za** **fax: 086 738 4568**
4.	Please do not hesitate to supply additional information that might be of relevance to this research.
5.	The return date for the completed questionnaire is **29 May 2015.**
6.	Should you wish to contact Deon Oosthuizen, you can do so on **083 419 4414** or **deon.oosthuizen@firstrand.co.za**

All information will be treated as confidential and will only be used to produce aggregate results.

Your co-operation is greatly appreciated.

Thank you in anticipation.

1. Which of the following risks, resulting from the process of software patching, do you deem more significant?

		Select one option
1.1	Potential security breaches which may result from a software vulnerability	☐
1.2	Unexpected downtime as a result of the patching process, due to a patch that breaks functionality or causes systems to be unavailable	☐

Comments:

.....

.....

2. Which of the following factors do you deem to be important in assessing the need to deploy a software patch?

		To a large extent	To a lesser extent	Not at all
2.1	Where a patch cannot be deployed or is not available, will relevant stakeholders and IT management be asked to sign off on the risk?	☐	☐	☐
2.2	For critical patches which need to be deployed as a matter of urgency, are unintended consequences considered?	☐	☐	☐
2.3	Consideration is given to the next release cycle and, where possible, patches are packaged and tested with other updates	☐	☐	☐
2.4	Sufficient testing is performed to ensure confidence and predictability for patches deployed to production systems	☐	☐	☐
2.5	Consideration is given to whether the threat can be mitigated without applying the patch or update	☐	☐	☐

2.6	The materiality of the threat is considered in terms of the ability to deliver safe and reliable service to the business	☐	☐	☐

Comments:

.....

.....

	To a large extent	To a lesser extent	Not at all
3. Do you believe that the software patching process within your organisation is suitably risk focussed?	☐	☐	☐

Comments:

.....

.....

4. Which of the following factors do you deem to be important when assessing a software patching programme:

		To a large extent	To a lesser extent	Not at all
4.1	The level of security knowledge of the IT staff who will be performing the patching process	☐	☐	☐
4.2	The amount of resources required to ensure that the task of patching can be effectively executed	☐	☐	☐
4.3	The level of end user knowledge and awareness of software patching	☐	☐	☐
4.4	IT infrastructure is suitably configured to allow an automated patching process to	☐	☐	☐

take place

Comments:

5. Based on your audit assessments, are patches tested prior to deployment in a formal testing environment within your organisation?		Select one option
5.1	Only for emergency (critical) patches	☐
5.2	Only for normal scheduled patches	☐
5.3	For both emergency and normal scheduled patches	☐
5.4	Other? Please specify	

Comments:

6. Based on your audit assessments, which of the following does your organisation use to determine the target timeframes for deploying patches		Select one option
6.1	Using vendor patching recommendations, such as severity ratings	☐
6.2	Through a risk assessment conducted internally	☐
6.3	Combination of vendor recommendations and an internal risk assessment	☐

Comments:

General Comments:

Thank you for sparing the time to complete the questionnaire.

Remember to save and return the questionnaire per email or fax.

e-mail: deon.oosthuizen@firstrand.co.za **fax: 086 738 4568**