LUND UNIVERSITY

Department of Informatics

Software Asset Management (SAM) and IT risks

An examination of how SAM affects IT risks in organizations

Bachelor thesis 15 hp, course SYSK16 in Information Systems

Authors: Joakim Lindvall
Viktor Löfqvist
Martin Ollinen

Supervisor: Björn Svensson

Examiners: Björn Johansson
Christina Keller

Software Asset Management (SAM) and IT risks: An examination of how SAM affects IT risks in organizations

AUTHORS: Lindvall, Löfqvist och Ollinen

PUBLISHER: Department of Informatics, Lund University

PRESENTED: May 2019

DOCUMENT TYPE: Bachelor Thesis

NUMBER OF PAGES: 59

KEY WORDS: Software Asset Management, Software Assets, IT risks, Risk Management, Organizations

ABSTRACT:
The existing literature suggests that organizations have difficulties managing and controlling their software assets. A number of IT risks are often associated with the poor management of these assets. At the same time, firms around the world claim that Software Asset Management (SAM) can be utilized to mitigate the IT risks, but there is no precise study based on the correlation between the two. This presents a distinct problem to researchers and cyber-security practitioners attempting to mitigate IT risks with the help of SAM.

This paper addresses this issue by presenting data from a series of interviews with cyber-security practitioners and IT managers. The paper generalizes the empirical findings to make broader theoretical points over the most appropriate usage of SAM against IT risks. The results suggest that whilst well-established methods like SAM remain important in managing software assets, it has a limited capacity to help organizations manage various IT risks.

Acknowledgments

We would like to express our immense gratitude to Assistant Professor Dr. Miranda Kajtazi for her teachings in Information Systems Security which helped inspire us to write about this topic. We also thank our supervisor Björn Svensson for his continued support and feedback at different stages throughout the writing of this paper. That said, the shortcomings of this paper are our own.

In addition, we would like to extend a very special thanks to Professor John Bew and Dr. Rod Dacombe from King's College London.

Table of contents

Figures

Tables

1 Introduction

1.1 Background

Organizations are facing an increasing difficulty of managing their software assets in recent years. As an example, a study by KPMG demonstrated that nearly 10 percent of all software were unaccounted for in large-scale organizations (KPMG, 2013). This claim is supported in the academic literature as similar scenarios have been observed by e.g. Henttinen (2018), Swartz and Vysniauskas (2013), and Varela, Méxas, and Drumond (2018). Indeed, the lack of effective management and control of software assets has become a source of financial and managerial woes for organizations (Ben-Menachem, 2005). Many organizations tend to undervalue their software assets and the primary methods for tracking software usage on networks are often inadequate (Albert, Santos, & Werner, 2013; Swartz & Vysniauskas, 2013; Varela et al., 2018). This ultimately leads to a succession of IT risks, including increased IT costs and an inability to identify what software is being used and where (Swartz & Vysniauskas, 2013).

Software Asset Management (SAM) provides organizations with the tools and processes to not only ascertain their software assets, but also their usage, where they are located, and how they are configured (Varela et al., 2018). It ensures that organizations are license compliant and closes the information asymmetry gap by accounting for all software assets that are deployed on the networks (International Organization for Standardization, 2015).

Nevertheless, there is still a number of acute problems which organizations need to address in order to implement an effective SAM policy. Increased usage of tablets and smartphones has become more commonplace amongst employees in organizations and subsequently negates the principles of comprehensive SAM (Varela et al., 2018). Additionally, organizations are constantly challenged by redundant and unknown software. In several studies, as discussed by e.g. Swartz and Vysniauskas (2013) and Varela et al. (2018), programs which could not be identified were secretly running on networks pertaining to an organization. This lack of insight into which modules are deployed, coupled with the lack of information on software dependencies, becomes increasingly problematic for patch management and security coordination (Swartz & Vysniauskas, 2013). Similarly, given that software assets are naturally dynamic in nature, not only are they transitory and unstable within organizations, but they are also subject to a high rate of change, this increases the complexity of implementing a security framework that reconciles the business needs with the actual infrastructure deployment (Albert et al., 2013; Swartz & Vysniauskas, 2013; Vion, Boyer, & De Palma, 2017).

It has in recent years, therefore, been argued that the propensity of SAM has evolved and aside from licensing control and compliance now also includes control of costs, competitive advantage, and to some extent IT risk management (Varela et al., 2018). SAM has further been touted by firms around the globe for its ability to ward off cyber-attacks due to more

complete control and supervision of the software assets that run on organizations' networks (see e.g. KPMG, 2018; Deloitte, 2018; Flexera, 2016).

Yet whilst prior research shows that that the benefits of SAM are highly correlated to licensing compliance, it is important to bear in mind that it does not delve into an organization's information security policies, as discussed by Albert, Santos, and Werner (2013). Instead, its primary contribution to organizations is the enablement of governance frameworks and processes for improved technological decisions and better supplier contracts (Albert et al., 2013). According to this argument, the concept of SAM boils down to an architecture for managing changes and complexity that allows organizations in turn to keep themselves updated when it comes to software on their networks (Albert et al., 2013).

1.2 Problem statement

SAM is utilized by organizations to reduce risks associated with non-licensing compliance (Albert et al., 2013). It can be utilized according to product-specific families (e.g. IBM or Microsoft product-catalogs) or it can be utilized as an organization-wide policy whereby all software assets are accounted for, including those not authorized by IT management (Varela et al., 2018; Vion et al., 2017). For illustrative purposes, the management and control of software assets through licenses costs approximately 21 percent of an organization's IT expenditure (Deloitte, 2012; KPMG, 2016).

It is claimed by firms that some of the IT risks mitigated by SAM include unauthorized or rogue software (see e.g. Deloitte, 2015; Flexera, 2016). It is also contended that SAM has the ability to detect aging software assets and obsolete security end-points (Flexera, 2016; KPMG, 2018; McAfee, 2013; Microsoft, 2018; Symantec, 2019; Red Hat, 2016). It is further claimed that SAM can mitigate IT risks pertaining to data integrity and software security through its access control and authorization management capabilities (Deloitte, 2015; Gartner, 2011; KPMG, 2009; Microsoft, 2017; Red Hat, 2016).

However, the problem with the claims above is that the precise relationship between SAM and these IT risks has not been established in a research context. In particular, SAM is not typically associated with more comprehensive IT risk management methods. It has to an extent been explored by Dempsey, Eavy, Goren, and Moore (2018) and Boyes, Norris, and Watson (2014). SAM has similarly been praised in its ability to act as a governance framework for managing and controlling software assets (Albert et al., 2013) and its ability to account for all software assets (Varela et al., 2018). But beyond this, no research has been carried out to investigate the effect of SAM on the IT risks above. There is, therefore, a gap regarding the claims by the firms above in the utility of SAM to mitigate IT risks.

1.3 Research question

How are the following IT risks affected in organizations when utilizing SAM?

- Unauthorized or rogue software (Shadow IT)
- Outdated software assets (Patch management)
- Data and software integrity
- Access control

1.4 Purpose

This paper presents cyber-security practitioners and IT managers' perception of how SAM affects the IT risks outlined in the research question. We know already depending on the literature referenced that access control, authorization management, and control of software are important when it comes to securing software assets (see e.g. Dempsey et al., 2018; Kondakci, 2006; Stoneburner, Goguen, and Feringa, 2002). These components are also key to understanding what SAM brings to the table in terms of our research question. In particular, it is hoped that the critical discussion of how SAM affects these IT risks will have a wider resonance.

1.5 Delimitations

In the context of this paper, SAM utilization is discussed as an organization-wide policy. The paper does not deal with SAM in terms of specific technologies, e.g. Flexera or IBM tools. Instead, it is a method of accounting for all software assets in an organization, including those not authorized by IT management. Incidentally, the way these software assets are accounted for go beyond the discussion of this paper.

Similarly, scholarly discussion of the management of software assets in organizations has undergone something of a theoretical renaissance in recent years, see e.g. Dempsey et al. (2018), Boyes et al. (2014), and Mbowee, Zlotnikova, Msanjila, and Oreku (2014). Management in this paper is delimited to the ability to control and authorize the use of software by employees. We will not look at e.g. the hardware aspects of IT asset management, the broader business discipline that SAM is part of.

In terms of data and software integrity, this paper does not deal with the manipulation of software assets as part of e.g. internal fraud. This is an IT risk that SAM is unable to manage. Outdated software assets (Patch management) in the research question is understood as software assets which have not received the latest security updates. The paper will not include a discussion of whether these updates are inherently insecure to begin with, as it would imply another thesis on its own.

2 Theoretical background

2.1 Software assets

A software asset is described as the information, resources, and software applications that support the business operations of an organization (International Organization for Standardization, 2015). An example of a software asset is either program-specific software (e.g. Microsoft Office, Microsoft Teams, SQL Server) or operating systems (e.g. the OS that runs on a computer, the OS that runs on a virtual machine). Above all, software assets are referred to in the literature as encompassing "everything concerning all corporate software" (Ben-Menachem, 2005).

In the context of this paper, an organization refers to business firms or government entities with an organized purpose. They vary from small-scale to large-scale operations and can consist of hundreds of employees and numerous software assets (Ben-Menachem & Marliss, 2004). In essence, organizations participate in the open market and rely on competitive advantages in the form of software assets to improve their business activities (Albert et al., 2013). They do so in order to improve customer relationships, outperform competitors, and ensure that supplier and business processes correspond to each other (Albert et al., 2013; Swartz & Vysniauskas, 2013).

According to Rotella (2018) there are three different types of software assets: Third-Party Software (TPS), Open-Source Software (OSS), and Commercial-Off-The-Shelf Software (COTS). TPS and COTS constitute the majority of software assets in organizations and they also present the most significant bugs and IT risks (Rotella, 2018). The reason for this is that organizations often fail to carry out proper quality checks before integrating them with existing software components (see e.g. Rotella, 2018, Voas and Hurlburt, 2015, or Xu, Lu, and Zhang, 2017). The problem with OSS, on the other hand, is that anyone can write and distribute the source code, and this can lead to severe IT risks such as corrupt data or the introduction of malware (Voas & Hurlburt, 2015).

More importantly, software assets are under constant threats from external attackers, such as hackers, who attempt to bring damage to the software by exploiting their vulnerabilities (Xu et al., 2017). Effectively managing these software assets has subsequently become a more pressing issue recently. But Voas and Hurlburt (2015) claim that measuring software is a difficult task. It can range from few lines of code to run-time behavior which, if left unchecked, can cause drastic harm to an organization's networks (Voas & Hurlburt, 2015). Indeed, the vulnerability of software assets on networks is often the leading cause of information security concerns. In order to protect themselves and their networks, organizations should conduct effective analysis and be able to identify all of their software assets as part of their risk management strategies (Islam & Falcarin, 2011; Xu et al., 2017).

It is argued in the literature that the foundation for ensuring secure software assets should be based on a top-down approach (see e.g. Xu et al., 2017). This allows the organizations to manage the presence of software and control them through e.g. legal certificates (Xu et al., 2017). This notion is reinforced by Rotella (2018) who argues that there is a need for effective managerial methods to quantify vulnerability levels and severities in software assets. This also

extends itself to specific releases of those products, such as ensuring that they have received the latest security updates (Mbowe et al., 2014; Rotella, 2018). An additional problem related to software assets is the existence of bugs. These are often the direct result of out of date updates and patches, and they play a major role in information security (Rotella, 2018).

In many cases, IT managers fail to address the above issues. They lack the proper framework to conduct threat assessments and vulnerability analysis of software assets on their networks (Dempsey et al., 2018). Based on a study, less than 20 percent of organizations have the capability to manage and control all software assets (Mbowe et al., 2014). A similar contention is raised by the academic literature (see e.g. Albert et al., 2013, Dempsey et al., 2018, or Stoneburner et al., 2002) and it is frequently brought up by industry and consulting firms (see e.g. KPMG, 2018, Microsoft, 2018, or Red Hat, 2016). The point they make is that organizations fail to implement standardized methods to eliminate the presence of vulnerabilities in their software assets, which in turn leads to a series of IT risks.

2.2 IT risks

In the information security context, an IT risk is defined as a function of the likelihood of a given threat-source exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization (Stoneburner et al., 2002). An example of an IT risk is an outdated software asset that contains vulnerabilities that may subsequently be exploited by external threats, such as hackers.

As argued by Islam and Falcarin (2011), secure software is all about mitigating risks from software assets to achieve business goals. Security is highly dependent on the context of where the software is deployed (Islam and Falcarin 2011). But measuring software security within a specific context is still not a mature practice (Islam and Falcarin 2011). Software security defines that only authorized parties can access and use software in an authorized way (Islam and Falcarin 2011). However, ensuring security is challenging because the software becomes more complex day by day (Islam and Falcarin 2011). It is continuously reported to be vulnerable to attacks and compromises despite using the latest security techniques and protocols (Islam & Falcarin, 2011).

2.2.1 Unauthorized and rogue software (Shadow IT)

Rogue software are applications which purport to perform some function, but, although appearing to, do not perform the stated function - often prompting the user to purchase the product (Pickard & Miladinov, 2012). Unmanaged and unauthorized software is a target that attackers can use as a platform from which to attack components on a network, this software is vulnerable because the software files may be forgotten or unidentified (Dempsey et al., 2018). Moreover, when vulnerabilities are discovered on such software, the responsibility to respond to the consequent risk is not assigned (Dempsey et al., 2018). According to a survey made by Flexera (2016), only 29 percent of organizations continually monitor their systems for security purposes to identify unlicensed and unauthorized software. The rest do so only periodically or not at all (Flexera, 2016). A key attack vector is to place (or replace) software on a device in order to perform malicious activities (Dempsey et al., 2018). Such software, called malware, can support exfiltration of data (compromising confidentiality), changing of

data (compromising integrity), disruption of operations (compromising availability) and/or establishment of remote command and control over the device to more flexibly perform malicious activity at the will of the attacker (Dempsey et al., 2018).

2.2.2 Outdated software assets (Patch management)

Dacey (2003, as cited in Cavusoglu, Cavusoglu, & Zhang, 2008) estimates that there are as many as 20 flaws per thousand lines of software code. According to Computer Emergency Response Team/Coordination Center, around 95% of security breaches could be prevented by keeping systems up-to-date with appropriate patches (Dacey, 2003, as cited in Cavusoglu et al., 2008). As vulnerabilities appear, software vendors periodically release patches in response (Dacey, 2003, as cited in Cavusoglu et al., 2008). For large organizations, with tens or even hundreds of thousands of network devices, the deployment of patches is a costly exercise impacting significantly on system availability, with consequences for properties of business processes, for credibility and revenue (Dacey, 2003, as cited in Cavusoglu et al., 2008). Failure to deploy a patch, however, risks exposing the host organization to the exploitation of vulnerabilities (Ioannidis et. al., 2012). The importance of timely patching in networks in the presence of externalities has been addressed by August and Tunca (2006, as cited in Cavusoglu et al., 2008), in which they develop a set of incentive structures for users to implement effective patch management when their actions impact upon the welfare of other users. They show that software vendors can offer rewards to encourage timely patching when vulnerabilities occur in both proprietary software and freeware (Ioannidis et al., 2012).

Cavusoglu et al. (2008) explain that vulnerabilities are signs of insecure software and that patch management is an effort to deal with the effects of that weakness. In that sense, patch management is not an effort to fix the root cause of the problem. Only software vendors can fix the problem by improving the security of their software programs. Cavusoglu et al. (2008) further cite Schneider (2004) that vendors currently do not have any incentive to fix their software because the cost of insecurity is not borne by the vendors. Costs of patch management, both damage and update costs, fall on the shoulder of the firms that are using the defective software (Cavusoglu et al., 2008).

2.2.3 Data and software integrity

The UK Medicines Healthcare products Regulatory Agency (MHRA) defines data integrity as “the degree to which data are complete, consistent, accurate, trustworthy, reliable and that these characteristics of the data are maintained throughout the data lifecycle” (MHRA, 2018 as cited in Kelleher & Greene, 2018). Kelleher and Greene (2018) continue by arguing that assuring data integrity requires effective quality and risk management systems which enable consistent adherence to sound scientific principles and good documentation practices. The international regulators have also defined an acronym (ALCOA) as the five elements necessary to assure data integrity throughout the data life-cycle. The following describes the basics of the five elements:

Attributable:	Data should be recorded by the subject who performs the task; it is important to document this action to enable full transparency and traceability.
Legible:	Means that all records must be readable and retainable on durable media for the duration of the records retention period.
Contemporaneous:	All activities must be recorded at the same time the action takes place.
Original:	The original record is the first initial capture of the data. Regardless of whether the data is recorded on paper or electronically, information should be available for review in the same state as originally collected.
Accurate:	Data must be accurately recorded. Therefore, education staff about the importance of follow integrity.

Table 2.1: Data and software integrity (Keller & Greene, 2018)

MHRA (2018, as cited in Kelleher & Greene, 2018) have additionally included that data governance measures should also ensure that data is complete, consistent, enduring and available throughout the lifecycle, where:

- Complete - data must be whole; a complete set
- Consistent - the data must be self-consistent
- Enduring - durable; lasting throughout the data lifecycle
- Available - readily available for review or inspection purposes

Furthermore, mobile devices and tablets (Bring Your Own Device) is an emerging trend where employees bring and use personal computing devices, such as mobile phones and laptops, on the company's network to access applications and sensitive data like emails, calendar and scheduling applications and documents (Yeboah-Boateng, 2016). Bring Your Own Device and mobile devices has become a real struggle for organisations to manage, control and the protection of their network (McAfee, 2013). A compromised mobile device with access to the organization's network could serve as susceptible entry points for nefarious activities within the network and possibly with access to sensitive information (Yeboah-Boateng, 2016).

The major harm from e.g. device theft shifts from data loss to data compromise, since it is much easier to recover data from a backup than reliably purge data from a media before unauthorized replication (Oleg & Ekaterina, 2017). It is further described by Oleg and Ekaterina (2017) that, as a result of the malware applications installed on these devices, there is an increased risk of user data being compromised. These applications execute themselves either automatically, search device by template or provide clandestine remote control over the device. This malware is usually distributed on various software depot cites and assume the guise of well-known applications with license protection removed. Nevertheless, it is not uncommon for a proper user to get a malware from official stores, as it slips through security screening (Oleg & Ekaterina, 2017). Yeboah-Boateng agrees that malware is one of the

leading threats to mobile devices and that any vulnerability leading to or making it plausible for malware to be installed on a mobile device needs to be addressed (Yeboah-Boateng, 2016)

Breach of data storage systems is the most dangerous threat, according to Oleg and Ekaterina (2017), since it simultaneously compromises massive amount of user accounts. According to Gemalto (2014), as cited by Oleg and Ekaterina (2017), the publicly disclosed number of breached user records around the world exceeded over 1 billion. The breaches occurred at the server-end, which makes the user unable to prevent data compromise even by ideally protecting his client-end device (Gemalto, 2014, as cited by Oleg and Ekaterina, 2017).

2.2.4 Access control

Secure identification of users, programming agents, hosts and networking devices is considered the core element in reducing IT risks (Benantar, 2006). To every unit of computing in modern systems with a relative level of security is attached an authenticated identity (Benantar, 2006). Benantar (2006) continues to explain that every identity in computing reflects real-life entities in that its level of granularity can be coarse (such as representing an organization; a group of people) or can represent a specific individual or a particular computing device. “Assurance in identity, referred to as *identity trust*, is established through authentication” (Benantar, 2006).

Passwords are still the most widespread means for authenticating users, even though they have been shown to create huge security problems (Jacomme & Kramer, 2018). This motivated the use of additional authentication mechanisms used in so-called multi-factor authentication protocols as cited in (Jacomme & Kramer, 2018). Jacomme and Kramer (2018) explain the typical multi-factor authentication protocols that typically includes additional devices, such as a mobile phone, or a dedicated authentication token. Two popular protocols are *Google 2-step* and FIDO’s U2F, which is supported by many websites, including Google, Facebook and GitHub (Jacomme & Kramer, 2018).

Benantar (2006) writes that access to objects including systems and network resources should depend on more than one condition being satisfied. Every system user or program acting on behalf of a user should operate using the least set of privileges necessary to complete a designated task, such as installing software assets. Every privilege assigned to a subject should be relevant only to the processing being performed. Extra privileges open the door for misuse and exploitation through human errors or malicious intents (Benantar, 2006).

2.3 Risk Management

Risk management is a method of identifying and assessing IT risks in organizations as well as taking steps to reduce their impact. Impact simply meaning the potential loss or negative effect which vulnerabilities in software may expose to said organizations (Stoneburner et al., 2002). Heavily tied to the concept of risk management is information security, which is defined as the protection of software assets against possible threats. Information security seeks to mitigate the IT risks which organizations face in order to prevent a loss of integrity or availability of IT systems (Chen, 2009; Stoneburner et al., 2002).

An integral aspect of risk management is to identify the boundaries of the IT systems, including all software assets that operate on the organization's networks. Yet effective oversight of software assets and dependencies on networks are often eluded by organizations, and significant IT risks may ensue in the form of failures of backup systems, misconfigured firewalls, or an external actor (i.e. hacker) gaining unauthorized access to networks (Alpcan & Bambos, 2009). As part of this effort, is it recommended that organizations use automated scanning tools which collect system information quickly and provide details pertaining to all software assets, such as their system criticality and when they were last updated (Stoneburner et al., 2002). Automated scanning tools are discussed by Kondakci (2006) who argues that they are essential in order to ensure the latest security updates throughout the life-cycle of a software asset. He contends that timely monitoring of software vulnerabilities and threats is a key concept when it comes to effective risk management (Kondakci, 2006). Chen (2009) further writes that these scanning tools can detect software threats, such as malware, and that they may even be used to gauge overall IT risks on an organization's networks and operating systems.

There is an additional benefit derived from the usage of automatic scanning tools for risk management that is discussed in the literature (e.g. Tvrdikova, 2008). On the one hand, the result of the analysis may give rise to an information security policy which can describe, amongst other things, the organization and its processes, identification of software assets, security infrastructure, and description of the present status and security measures (Tvrdikova, 2008). On the other hand, it may also lead to a system security policy that can define a list of pre-approved software assets in a specific system of an organization, that in turn operates with the direct approval and knowledge of IT management (Tvrdikova, 2008).

Stoneburner et al. (2002) similarly write that risk management should encompass a vulnerability analysis that details which software assets could be exploited by potential threats, e.g. by disgruntled employees or unauthorized users. In terms of protecting software assets, it is further important that preventive technical controls such as authorization and access controls exist in order to maintain an organization's security policy (Stoneburner et al., 2002). These controls make sure that software assets on organizations' networks fall in line with a list of pre-approved software. They additionally protect against vulnerabilities associated with rogue software, which operate without the knowledge of IT management and could potentially be exploited by external actors and cause a chain of compromises, as discussed by Alpcan and Bambos (2009). Indeed, the latter aspect is often overlooked by organizations when dealing with risk management of software assets. Alnatheer (2015) expands on the idea and describes how regulating software assets on networks can influence the behavior of the employees to comply with the official information security policies.

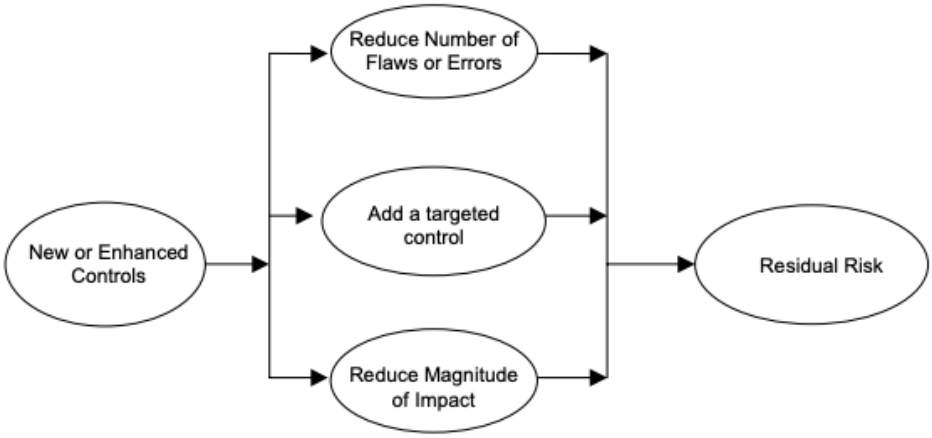

Figure 2.1: Implemented Controls and Residual Risk (Stoneburner et al., 2002, p. 40)

However, as discussed by Chen (2009), it is important to note that IT risk mitigation also requires other technical tools like cryptography, intrusion detection systems, antivirus software, audit trails, and backups. It is not absolved by authorization and access controls alone. Chen further writes that "risk management is more of an art than a science" and that there exists more than one way to form a comprehensive risk management policy in organizations (Chen, 2009). Moreover, an organization's software assets operate in an environment where both technical and social factors play a role. Risk management should for this reason not ignore another important element when it comes to mitigating IT risks, namely the level of awareness amongst employees (Giorgini & Paja, 2017).

The concept of information security awareness as part of risk management efforts is widely discussed in the literature. For example, employees are often seen as posing the biggest risk to ensuring secure software assets and organizations have to prioritize not only effective IT management but also an overall security culture (Alnatheer, 2015). In a similar vein, Tvrdikova (2008) argues that it would be erroneous to rely solely on technical tools to mitigate IT risks. Risk management must also deal with the end-users of the system as well as the effective management of software usage on networks; it entails a whole lot more than access control and authorization from a purely technical perspective (Tvrdikova, 2008).

But whilst impressions may be that technical tools, for the above reasons, are insufficient they still form an integral part in mitigating IT risks. Fenz, Ekelhart, and Neubauer (2011) argue, for instance, that technical solutions as part of risk management should be viewed in terms of how well they contribute to managerial insight and control of software assets. This can later be used to complement social factors, such as end-users, as discussed by Fenz et al. (2011). The key to understanding effective risk management is, therefore, that organizations must have a thorough knowledge of the system in question, potential threats, and corresponding vulnerabilities in software assets. This in turn provides organizations with the framework to deal with both overall and software-specific IT risks (Fenz et al., 2011).

A similar point is raised by Datta (2010) who argues that it is nearly impossible to eliminate all the IT risks in an organization. Through appropriate technical controls, though, these can be reduced. In this endeavor, it is the responsibility of IT management to utilize the most cost-efficient and effective tools in order to minimize the negative impact of IT risks as a result of software assets. This includes the notion that software assets on networks must be continually upgraded and that security architectures are constantly reviewed and monitored (Datta, 2010).

2.4 Software Asset Management

Software Asset Management (SAM) is a method of tracking software assets that enables organizations to maintain and optimize their use of software (Albert et al., 2013). It aims to reconcile software usage with the correct license rights, which are provided by software suppliers, in order to mitigate IT risks associated with licensing non-compliance (Albert et al., 2013; Vion et al., 2017). An example of SAM for managing software assets could be the right to use some specific software, which is in turn documented in software contracts, license documentations, and receipts (Henttinen, 2018; Varela et al., 2018).

The International Organization for Standardization (ISO) further defines SAM as the complete control and supervision of software assets throughout their life-cycles in an organization, whether they are mobile, premise-based, cloud-based, or hosted (International Organization for Standardization, 2015). Nevertheless, the complexity of managing software assets has grown in recent years, especially as suppliers have shifted toward more complex software license schemes following the dissolution of traditional IT architecture (i.e. both hardware and software) in favor of cloud environments (Vion et al., 2017).

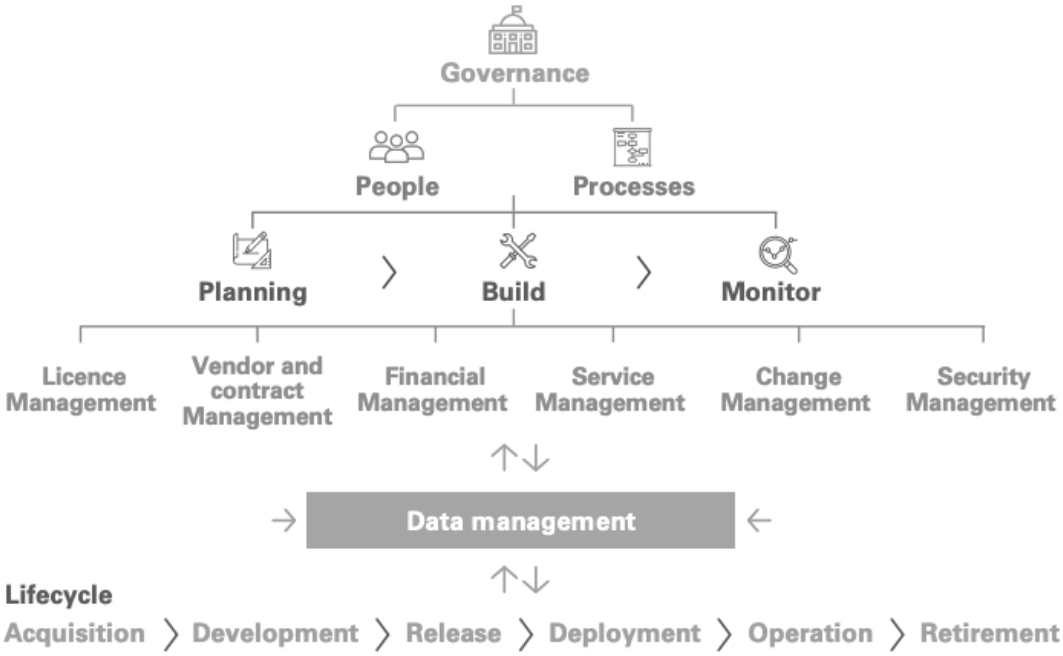

Figure 2.2: An example of Software Asset Management processes (KPMG, 2018, p. 2)

Whilst the effect of SAM on mitigating IT risks lacks thorough academic research, it is widely discussed in the available literature that its primary significance lies in steering software assets on networks and managing their availability to end-users. The latter, it is argued, might even serve as the basis for a comprehensive risk management policy in large-scale organizations (Dempsey et al., 2017; Swartz & Vysniauskas, 2013; Varela et al., 2018). It is equally important to note that the literature contends that SAM could possibly act as a framework for control of software assets throughout their life-cycles, including from their conception to the latest security updates (Ben-Menachem, 2005).

2.4.1 Software Asset Management in organizations

Organizations face an ever-increasing difficulty in trying to account for software usage according to the contractual rules established by software suppliers (Swartz & Vysniauskas, 2013). Many organizations tend to undervalue their software assets and they fail to allocate enough resources to manage them properly (Swartz & Vysniauskas, 2013). At the same time, software suppliers are stepping up their auditing activities in order to confirm that the license numbers of their clients are accurate. A lack of SAM in this regard could result in significant and unexpected costs for organizations (Varela et al., 2018; Ben-Menachem, 2005).

The need for SAM was first suggested by Holsing and Yen (1999) when they argued that it would mitigate technical, legal, managerial, financial, and even ethical risks in organizations. They state that these risks are pivotal in driving the need for SAM and that they are based on the best interests of IT management, end-users, and software suppliers (Holsing & Yen, 1999). Depending on the literature referenced SAM additionally enables organizations to gain an accurate view of installations and usages of software assets which can help complement cohesive risk management policies (Dempsey et al., 2018; Vion et al., 2017).

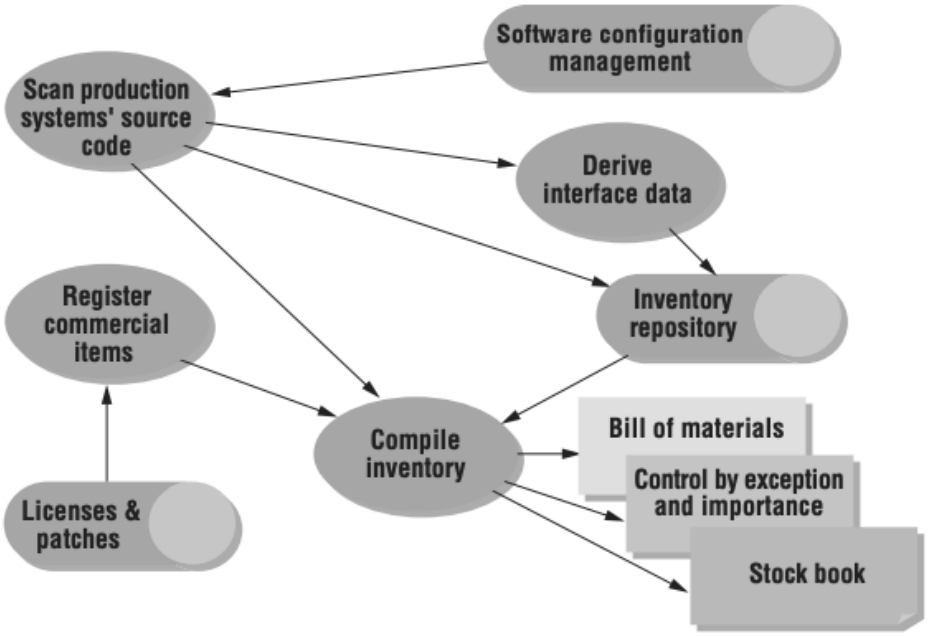

Figure 2.3: Creating an inventory for software assets (Ben-Menachem & Marliss, 2004, p. 37)

Klint and Verhoef (2002) describe e.g. in their study how the lack of a software asset inventory can hamper an organization's insight into total software usage. They further discuss that many organizations fail to implement basic software asset tools and that this puts the organization at great risk (Klint & Verhoef, 2002). This sentiment is echoed by other studies and it is discussed at great lengths by Ben-Menachem (2005) and more recently by Varela et al. (2018). Ben-Menachem and Marliss (2004) go one step further and write that "one of the most significant failures [by organizations] … is the absence of systems to gather, support, and supply information for managing software assets". They contend that this could result in a plethora of IT risks including the presence of software assets that operate without the knowledge of management. Viewed in this way, SAM is not merely a method to track

software assets, but also a managerial process to gauge and adapt to the constantly changing technological and business landscape (Ben-Menachem & Marliss, 2004).

2.4.2 Theoretical results: Software Asset Management and Risk Management

Traditionally speaking, organizations that follow a centralized information security policy do not permit their employees to install any software to their computers without the direct approval of IT managers (Jakubicka, 2010). Lately, though, the trend has been somewhat reversed and employees now often have more leeway when it comes to installing and using software assets on their own devices (Jakubicka, 2010; Swartz & Vysniauskas, 2013).

Ben-Menachem (2005) writes that SAM actively tracks and keeps an up-to-date inventory of all software assets that operate on an organization's networks. SAM further encompasses different person roles, authorization, access control, and support throughout the entire software license life-cycle (Ben-Menachem, 2005; Jakubicka, 2010). Equally important to note is that the inclusion of SAM in an organization requires the end-user to go through the IT manager before any software is installed. As an example, it ensures that the deployment of a software asset goes through the proper steps of user verification and authentication, and managers in turn can see to it that it falls within the scope of the business processes of the organization (Albert et al., 2013; Dempsey et al., 2018; Vion et al., 2017).

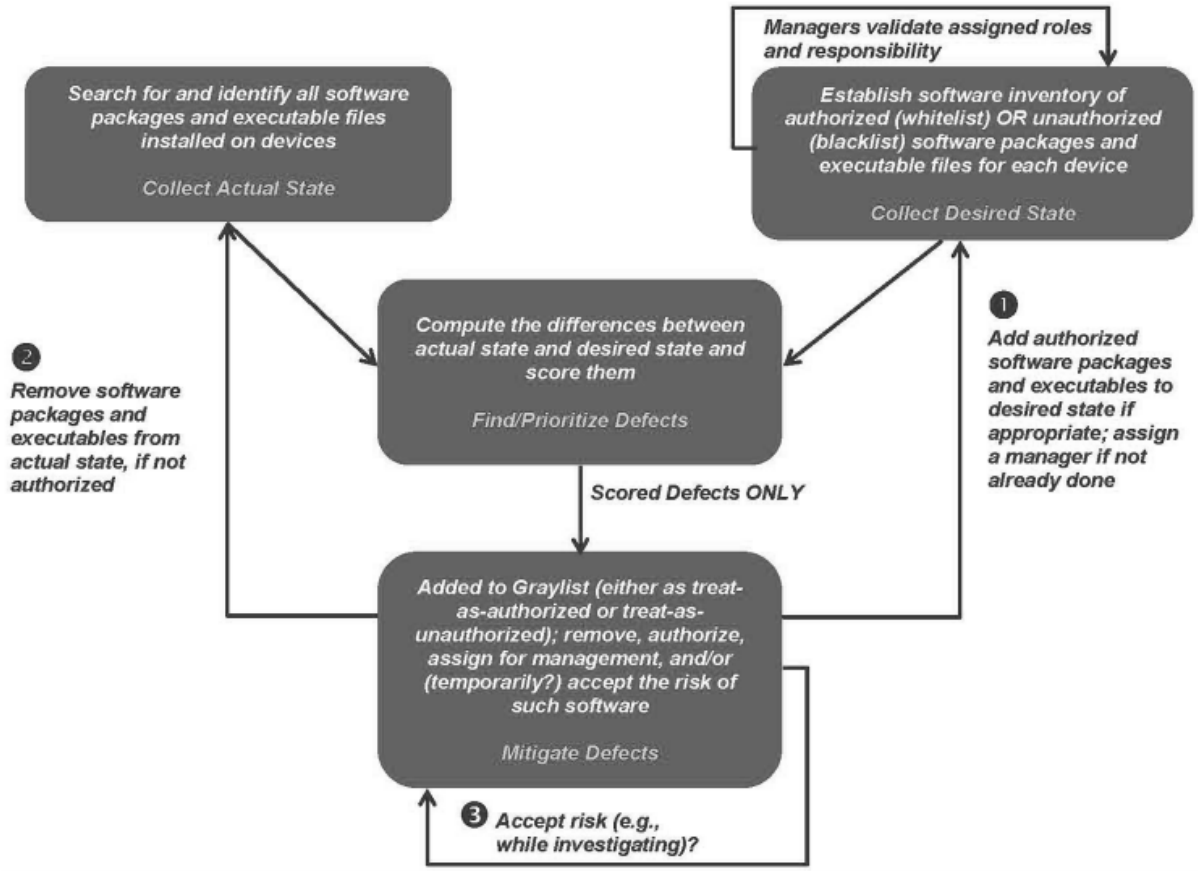

Figure 2.4: An example of Software Asset Management operations (Dempsey et al., 2018, p. 16)

Boyes et al. (2014) argue that by ensuring that only authorized software assets are permitted on an organization's networks, SAM limits the software usage to those necessary to carry out the business operations of the organization. They contend that the access control and configuration management aspect of SAM make it easier for IT managers to identify unauthorized changes as well as to detect threats and vulnerabilities of their software assets (Boyes et al., 2014).

In addition, by maintaining and keeping track of software assets through SAM organizations can ensure that the management objectives are achieved (Henttinen, 2018). It also sets out to identify all software assets that operate on a network (Swartz & Vysniauskas, 2013). Through verification and compliance steps SAM can detect all exceptions to policies, processes, and procedures pertaining to licensing usage rights (Swartz & Vysniauskas, 2013). The inclusion of an up-to-date inventory of software assets is as beneficial for contract management as it is for organizations to monitor and control IT risks associated with undetected software deployments (Henttinen, 2018).

Related to this is the discussion of SAM in the literature based on its merits as a way to gauge software assets which are redundant, unused, and at the end of their life-cycle (Ben-Menachem, 2005). SAM tools provide organizations with an overview of all current software deployments, even in virtual machines and the resources allocated to these (Vion et al., 2017). It additionally includes an inventory of a list of software assets and their dependencies that management can use to quickly observe the state of systems, networks, as well as estimate the highest IT risk areas (Ben-Menachem & Marliss, 2004).

In fact, Boyes et al. (2014) claim that there is a strong relationship between information security and asset management in organizations. They write that an up to date inventory of software assets allows organizations to analyze their systems in order to identify and assess any IT risks that may be present on their networks (Boyes et al., 2014). Mbowe et al. (2014) similarly argue that, as part of an organization's security policy, it is imperative that there exist effective processes which can document software assets. Whilst they are not directly discussing SAM, they write e.g. that a software asset inventory is crucial in mitigating IT risks since it strengthens the control of security over various software artifacts (Mbowee et al., 2014).

3 Methodology

3.1 Qualitative study

This paper aims to explore how SAM affects the IT risks outlined our research question. A good insight into how organizations work with both risk management and SAM is necessary in order to answer the research question at hand. These two areas are not common knowledge among employees in most organizations. Jacobsen (2002) claims that qualitative studies, with a hand full of conversational interviews, provide detailed data that allow for a deeper dive into a subject matter, unlike quantitative studies. Based on this, we quickly realized that we needed to perform a qualitative rather than a quantitative study. In order to perform such a study in a way that would generate usable data, a set of questions needed to be prepared and used to keep the interviews on track and within the specific range of topics that are of interest. This is a practice supported by Backman (2016).

Backman (2016) describes qualitative research methodology as the use of verbal communication which generates observations and results. This differs from quantitative research methodology, which makes use of measurements, mathematics, and quantification (Backman, 2018).

3.2 Interview participants

We used two main methods to get in contact with qualified interview participants. The first method was by finding people with the desired experience on platforms such as LinkedIn and contacting them by phone. The second approach was by contacting companies through public communication email addresses that went to their customer service or service desk. In the email, we stated that we would like to be forwarded to someone they thought could help us. The interview participants we intended to get in touch with were those with experience in working with SAM, system architecture and IT security. This could be either by working directly with the above-mentioned areas or holding a managerial position over people who work with them. We aimed to find interview participants with at least five years of experience in working with this or associated areas of expertise. The reason for reaching out to these individuals is because we assumed they would have insight into how SAM works and how it is utilized within their organizations while also having an understanding of how SAM affects IT risks and IT security. Another requirement was that the organization in question had to use SAM or be heavily involved in SAM implementation. We mainly chose to focus on organizations in the IT industry, preferably those who produce IT services or software, since we assumed that these organizations would have the highest chance to offer interviews that fulfill the requirements mentioned above.

The topic of the thesis does to some extent handle essential parts of an organization's security. We, therefore, offered all interview participants the opportunity to be anonymous. In order to be able to ensure that all parties involved would have a mutual view of anonymity we always sent our pseudonymous description of the organization and the transcription of the interview to the interview participant for approval before we used data from the interview.

Appendix	Organization	Position	Experience with SAM	Interview		
				Date	Method	Location
C	Omegapoint Malmö AB	Chief Executive Officer	Low	May 9th, 2019	IPI	Malmö, Sweden
D	Anonymous cyber-security venture firm	Chief Operating Officer	Medium	May 11th, 2019	SI	Tokyo, Japan
E	Anonymous defense and cyber-security firm	Cyber-security consultant	Low	May 16th, 2019	SI	Sydney, Australia
F	ATEA AB	Senior IT-Security consultant	Medium	May 16th, 2019	SI	Malmö, Sweden
G	Anonymous cyber-security practitioner	CSIRT handler	High	May 19th, 2019	EC	London, England
H	Anonymous transport and logistics firm	IT asset manager	High	May 17th, 2019	SI	London, England
I	Anonymous energy production firm	Head of Commercial IT and IT-security	High	May 21hst, 2019	EC	Stockholm, Sweden

Table 2.2: Interview participants

Note: IPI = In person interview SI = Skype interview EC = Email correspondence

3.3 Interview process

We created the interview questions (see Appendix) in such a way as to be able to answer the research question of the thesis. One aspect that dictated how we structured the interview questions was the need for honest answers with high validity. Jacobsen (2002) proposes that open questions mean that an interview maintains high internal validity. We, therefore, tried to create questions that are as open as possible. SAM and IT risks are however two subjects that entail multiple possible interpretations and points of discussion. To make sure that the interviews generated relevant inquiries that can be used to answer the research question, we crafted the interview questions based on the literature presented in chapter 2.

The interviews were performed at the interview participants' premises if they were in the Skåne region, otherwise, they were performed using Microsoft's communication tool Skype. According to Jacobsen (2002), you cannot always notice certain physical signals that the interview participants shows, for example, if they feel threatened, uncomfortable or bored. According to Jacobsen (2002), this could affect the result in a qualitative approach. We believe that there is a difference between a video interview and a telephone interview since you actually see the person in question and their body language. There were however two instances where the interview was conducted in the form of email correspondence. This was done by emailing the interview participants the definitions of the terms used in the interview. If there were no disagreement regarding the definitions, the set of questions were emailed to the interview participants. If there were any need for clarification, the interview participant emailed the interviewer before answering the question and the interviewer provided an explanation.

Initially, the interviews started by us asking the interview participants for approval to record the interview. One of the authors acted as an interviewer during all the interviews and led the discussion while the others recorded. To complement the audio recordings, physical notes were written during the interview. The notes were not really needed because we recorded the interviews, but due to experience related to problems with audio files in previous projects, we chose to minimize the risks.

The first thing we strived to do during the interview was giving the interview participants a clear picture of who we were, our background and the purpose of the essay. We then wanted the interview participant to start talking and warm up a bit by answering simple questions about the organization and what work the interview participant performs. We also wanted the interview participant to present his view and definition of the main concepts that the paper examines, SAM and IT risks. If it differed from the definitions provided in the theoretical chapter (2.2 or 2.4) we explained our definition in order to avoid any misunderstandings.

We were flexible about in which order we asked a question, whether exactly as it was written or in differing order. For example, if we asked a question and the answer also answered a different question that would be asked later in the interview, in some cases we chose not to ask the latter question to minimize unnecessary repetition. Nevertheless, we could sometimes repeat a question we indirectly received an answer for the purpose of avoiding misinterpreting answers.

We chose to keep the interviews short and with a focus on the company that the interview participants represented rather than the participants themselves. This was to ensure that our questions were answered as truthfully as possible and to keep the quality of the answers high by maintaining focus during the interview.

3.4 Transcription

It was determined at an early stage that the transcription would only contain what was relevant to the interview and to the subject at hand. Issues or chatter that did not in any way affect the interview questions that were written down in the interview template (see Appendix) were not included in the transcription. Temporary stuttering, filling words such as

"ehh" and taking time to reflect are some phenomena that we chose not to include in the transcription since we believe it does not affect the answers and improves readability.

If a sound recording were to be lost, the idea was that the interview should be recreated to the extent of our abilities, but not verbatim, based on the notes taken during the interview.

The goal of the transcription is to give the collected data appropriate and interpretable form in order to be able to answer the original issue (Backman, 2016). According to Backman (2016), this can be facilitated by a certain rough structuring or categorization of the collected data before data collection begins. The challenge lies in steering away from simple descriptions and capturing an overall picture of the essential reasons for its expression (Backman, 2016). We intend to achieve this by categorizing the results we present in accordance with the categories we have identified in our survey model.

3.5 Selection of literature

Appropriate literature was gathered in order to provide context to the problem statement and research question. We mainly focused on finding relevant literature in the form of books and research papers. Initially, we tried to limit our search results to a time span between 2010-2019. The reason being that we wanted to encompass scholarly discussions of recent developments in software assets, such as those offered by cloud services. We discovered, though, that a lot of the literature and theoretical results written about SAM were out of date. We consequently had to expand our criteria and include older literary submissions as well. In this endeavor, we utilized academic search engines such as Primo (King's College London), LUBsearch, LUBcat, IEEE Xplore Digital Library, Basket of 8 and Google Scholar. We used the following search terms: "Software Asset Management"; "IT Risks"; "Software assets"; "Software Assets and Security"; and "Software Asset Management and IT Risks".

3.6 Reliability

Reliability is defined as "the extent to which a measurement yields the same answer" (Miller, Miller and Kirk, 1986). There are three commonly used methods for assessing reliability; *test re-test, internal consistency* and *alternative form* (Mitchell, 1996). All three were considered for ensuring reliability in our study, but only internal consistency was ultimately deemed an effective approach as explained below.

Testing re-testing is assessing reliability by performing the same test with the same subjects at a different time. The conditions for the re-test should be the same or as close to the first test as possible (Mitchell, 1996). This method should however only be used as a supplement to other reliability assessment methods because it might prove difficult to get interview participants to agree to a second interview (Saunders, Lewis and Thornhill, 2012). This, combined with the fact that it would not be viable due to time constraints, made us choose not to use this method of assessing reliability.

Alternative form assesses reliability by including the same or similar questions more than once while changing the wording (Mitchell, 1996). This lets you compare the answers of

similar questions to check for potential inconsistencies. The participant may, however, refer back to a similar question or give an insufficient answer to effectively compare the two (Saunders et al., 2012). It should also be noted that this approach might significantly increase the interview process by having to re-ask the same open questions more than once. We, therefore, deem this second alternative to be insufficient.

Internal consistency is a method that assesses reliability by comparing the consistency between answers to different, but related questions (Mitchell, 1996). A possible inconsistency found by this method could, for example, be: "There are no security risks that can be prevented by using SAM" being given as an answer to one question, but then getting "the greatest risk preventable by SAM is unauthorized software" as an answer to a later question. The overall consistency can then be calculated using formulas such as Chronbach's alfa (Saunders et al., 2012). This method seemed the most viable to perform and is a good way of indicating if our qualitative interviews have a high level of reliability.

Internal consistency as well as thoroughly detailing our methods for further research and potential re-testing were therefore chosen as measures to ensure reliability in our research.

3.7 Validity

There are two different types of validity that are often discussed (Saunders et al., 2012; Jacobsen, 2002): internal validity and external validity. Internal validity refers to the validity of the result (Saunders et al., 2012). One way to check the internal validity is to carry out a data submission validation, which is when the transcription of a completed interview is sent to the interview participant afterward in order to examine to what extent they recognize themselves in the answers that have been transcribed (Jacobsen, 2002). This is a method we used to ensure that our data was correct. Above all, we thought this was important for those interview participants who chose to be anonymous since we did not want to risk publishing anything that they thought could be the information used to identify their organization or the interview participants themselves. The external validity regards the extent to which the result of a study can be generalized (Saunders et al., 2012). This is difficult when it comes to a qualitative approach since the goal is not to generalize in a larger context (Jacobsen, 2002). According to Jacobsen (2002), the goal is to develop a more general theory based on the data obtained from a smaller selection of interview objects.

3.8 Ethics

According to Jacobsen (2002), there are three basic requirements that a survey must meet in order to be ethically performed; informed consent, right to privacy and correct reproduction.

Informed consent is comprised of several smaller requirements that should be met to the greatest possible extent (Jacobsen, 2002). On one hand, those who are examined must freely choose to participate in the survey, which may seem obvious, but it is important that the data did not come to be due to any underlying cause or pressure from an external operator directed at the person participating in the survey (Jacobsen, 2002). In addition, those being interviewed

must according to Jacobsen (2002) get full access to information about the purpose of the survey as well as information about any benefits and disadvantages of their participation.

The right to privacy is intended to consider the protection of information relating to the participant being interviewed (Jacobsen, 2002). It is, therefore, of great importance to seek discretion by offering the participants guaranteed anonymity and confidentiality, in order to avoid that the information can be linked to a private person (Jacobsen, 2002). We, therefore (as stated previously in 3.2) offer interview participants the right to remain anonymous early in our correspondence.

The last requirement is correct reproduction (Jacobsen, 2002). This means that the interview participant should be correctly represented in any transcript or recreation of the interview and that they need to be presented in the right context (Jacobsen, 2002). To ensure that this was done correctly, the respondent can always demand complete reproduction of the interview (Jacobsen, 2002). If the correspondent had any concerns regarding the reproduction, the issue would be reviewed and corrected if it was indeed deemed to be incorrect.

4 Results

This chapter presents the results of our empirical study. The results include a series of seven interviews with cyber-security practitioners and IT managers with experience of SAM and the IT risks outlined in the research question. The full interview transcripts can be found in appendices Part C to Part I and are referenced by e.g. (Appendix D, answer 1), denoting Appendix: Part D, answer 1.

All of the participants agreed that there are numerous IT risks pertaining to software assets in organizations. Some argued that SAM could be utilized to control and manage these IT risks, especially in large-scale organizations (see e.g. Appendix D, answer 4). In particular, it was indicated that SAM could potentially complement more rigorous information security efforts:

> I think when it comes to information security it's pretty effective. This is the first step that corporations should think of when introducing their information security policy. (Appendix D, answer 4)

> I would say that Software Asset Management should be part of a security strategy. Not necessarily a SAM strategy. I'd place it as a security strategy and SAM as adding to that and show with data that these are the reasons why we need this kind of strategy: to protect ourselves from malware or whatever. (Appendix H, answer 4)

The problem with existing information security policies, as one participant informed us, is that it is difficult to find a system that covers all of the various IT risks (Appendix D, answer 2). Another participant stated that it is, therefore, vital to control and manage software assets, on both clients and servers, and that organizations' information security efforts must be backed up by good policies, such as SAM (Appendix F, answer 2). He added that whilst SAM is not typically associated with information security, organizations should take note of its ability to mitigate IT risks that stem from the poor management of software assets (Appendix F, answer 3).

Equally, one participant argued that SAM allows organizations to distribute software assets and manage them effectively through licenses that are specific to employees (Appendix C, answer 4). It is further claimed that in the absence of this effective license management, there is a risk that employees use unapproved software, or the wrong version, which consequently could open the door to a number of vulnerabilities (Appendix C, answer 4). Another participant expanded on this idea and said:

> If you're unable to control what's installed on the estate there might be risky software... Obviously, you need to control that and minimize that and enable discovery and a quick resolution to remove those types of risks on the estate where they arise. If you don't have proper controls on, you know, admin rights on machines not setup properly so that end users can just install what they like on their computer. (Appendix H, answer 2).

However, one of the participants acknowledged that risk management efforts led entirely by SAM is not possible. The participant suggested a number of factors that inhibit the effect of

SAM against IT risks, including the need to consider other IT risks like Advanced Persistent Threats (APT):

> [I]f you really want to defend your system against serious cyber-security risks like APT threats... I mean, no cyber-security professional would recommend SAM. I mean, [no one] would say that introducing SAM will solve all cyber-security risks, especially defending themselves against APT. (Appendix D, answer 4)

In this endeavor, organizations are more apt to use systems like Endpoint Detection and Response (EDR) or Web Application Firewalls (WAF), as opposed to SAM (Appendix D, answer 4). This notion was also shared by two other participants, who both stated that it would be impossible to rely exclusively on SAM to mitigate IT risks (Appendix F, answer 4; Appendix I, answer 4).

Yet, in terms of affecting the IT risks outlined in our research question, there was no general consensus from the participants. Some of them were aware of utilizing SAM to improve e.g. information security in organizations (Appendix D, answer 3; Appendix E, answer 3). One participant claimed to have experience with SAM in organizations and stated that it is quite limited from an information security standpoint (see Appendix H, answer 3). Conversely, one participant replied that there are much larger IT risks at hand, not outlined in the research question, which SAM does not affect, such as:

> Managing codebases, deployments, costs and user access. (Appendix E, answer 2)

He further noted that SAM is not really that effective against these IT risks (Appendix E, answer 4). Likewise, another participant added that there is an additional risk component related to software assets that SAM does not address:

> Also, I'd suggest with software-as-a-service models these days are problematic because there are data security issues there... You don't know where your data is being stored because generally with software-as-a-service it's stored somewhere else not within your network, so you haven't got control... And interestingly enough, you don't need admin rights to use software-as-a-service products you can generally buy it and you can create an account and you can just use that software over the internet and usually store the data in whatever you're working on with the external parties which means you have no control over that. (Appendix H, answer 2)

4.1 Unauthorized or rogue software (Shadow IT)

Nearly all participants agreed that the IT risks associated with unauthorized or rogue software (Shadow IT) can be mitigated by SAM to a certain extent. This rings especially true for large-scale organizations (Appendix D, answer 5). A major problem, as contended by one participant, is employees' use of open-source software to perform their daily work tasks: this results in additional vulnerabilities of which IT management does not have the capacity to approve nor analyze beforehand (Appendix C, answer 4). By comparison, one participant is more adamant about how SAM affects unauthorized software:

> It should be the main way really. I would say you have a whitelist which is supported software that's allowed on the estate and everything else is blacklisted... [T]here is no reason why something blacklisted can't find its way into the whitelist if there is a business requirement and if it's gone through the appropriate approval process to become a supported software item in the catalogue. (Appendix H, answer 5)

However, at the same time, the participants also answered that SAM utilization is only a part of organizations' overall efforts to mitigate unauthorized and rogue software. As an example, some of the participants posited that in order to combat the numerous IT risks, organizations require many different tools in their arsenal (Appendix D, answer 5; Appendix G, answer 5). It is further conveyed that SAM as a principle might work, but that it is a lot more difficult to combine it with e.g. existing information security capabilities:

> I think usually before introducing any IT security products they usually need to formulate their internal security policy first. And then get management to introduce relevant information security system [sic] inside the departments. (Appendix D, answer 5)

In a similar vein, one participant argued that SAM can only really affect unauthorized and rogue software if it is implemented correctly, and a lot depends on e.g. whether SAM operations are supported by the organization's various operating systems, architecture, and processors (Appendix F, answer 5).

By contrast, another participant argued that whilst SAM might not be utilized to discover unauthorized or rogue software, it could be utilized to avoid their presence on networks (Appendix C, answer 5). The participant added that there is also a risk of employees using their own software assets in organizations, and not in an organizational context, which is not licensed by the IT management (Appendix C, answer 5).

It is worth mentioning that one participant argued that SAM does not affect e.g. unauthorized attacks, which is a common security problem related to software assets amongst organizations (Appendix D, answer 5). Equally important to note is that whilst SAM is limited in this regard, there are no alternative products or methods against these types of IT risks; it is, therefore, up to third-party suppliers of SAM to prove that it can mitigate unauthorized elements on organizations' networks (Appendix D, answer 5).

Likewise, centralized management of software assets through SAM could be tricky to enforce depending on the organization. One participant stated that in the consulting industry SAM might even have adverse effects for business operations: a lot of the software needs to be on par with the latest technological capabilities, and IT management might not be as quick to respond to the changing requirements of software assets (Appendix C, answer 5). Another participant stated that it depends entirely on the organization in question:

> Only if an organization has a fixed number of software applications. In a fast-paced development environment, it should be more of a guideline than a strict process. (Appendix E, answer 5)

Interestingly, we discovered that one participant believed that SAM is particularly effective in certain large-scale organizations, such as hospitals, when it comes to unauthorized or rogue

software assets (Appendix C, answer 5). This sentiment was shared by another participant, but he added that it is not effective against e.g. supply-chain attack risks (Appendix G, answer 5).

4.2 Outdated software assets (Patch management)

The participants were more divided on how SAM affects IT risks associated with outdated software assets (Patch management). One participant, incidentally, answered that he did not have sufficient knowledge of SAM to be able to answer what this might look like in practice (Appendix D, answer 6). Another participant answered that SAM makes a vast difference (Appendix C, answer 6). It could also potentially be combined with other technologies, which in turn could force an uninstallation or prevention of programs (Appendix F, answer 6; Appendix I, answer 6). Similarly, one of the participants answered that SAM could encompass the following functionalities:

> It could track which software has been updated and provide reporting capabilities for business owners. (Appendix E, answer 6)

But, on the flip side, he also added that SAM might have nothing to do with patch management itself:

> Currently, this kind of functionality is provided by third-party anyway, e.g. cloud providers take care of patches, hardware/OS vulnerabilities and asset life-cycles. (Appendix E, answer 9)

Another participant further stated that SAM is limited for patch management purposes:

> SAM can only track limited assets... [such as] OS base, drivers, installed software, [but not e.g.] portable software, downloadable executable, managed codes/dot net, browser extensions, activex, java app, flash/AIR app, HTML5 app, powershell, wsh, batch scripts. (Appendix G, answer 6)

Equally important:

> I personally do not think that Software Asset Management should be part of that function to be honest with you. That is a purely security one and going back to the security policy that's created where that has to be done... [SAM] can supplement [patch management] and assist it. (Appendix H, answer 6)

4.3 Data and software integrity

The majority of participants agreed that SAM could potentially mitigate IT risks pertaining to data and software integrity (Appendix C, answer 8; Appendix D, answer 8; Appendix E, answer 8; Appendix H, answer 8). One participant described SAM as a “source of truth” with regard to software assets:

> That's how you could maintain software integrity on what's used on your estate because it's from one source and it's a safe secure source. (Appendix H, answer 8)

By contrast, one of the participants responded that it is “quite limited” for this purpose:

> For example, SAM won’t prevent malicious DLL-sideloading. (Appendix G, answer 8)

Likewise, another participant contended that whilst SAM could play a part in ensuring data and software integrity, these security requirements are often trickier to oversee (Appendix F, answer 8). It could be utilized in conjunction with other control mechanisms, but much of the IT risk depends on the software itself and the way that information is stored on the program (Appendix F, answer 8). For this purpose, the software assets require additional protection, not covered by SAM (Appendix F, answer 8; Appendix I, answer 8).

4.4 Access control

Two of the participants largely agreed that SAM is useful for access control purposes, especially in large-scale organizations (see e.g. Appendix D, answer 7, or Appendix F, answer 7). Furthermore, one participant posited that SAM could help standardize and control the number of software assets that employees have access to in organizations, which in turn reduces some of the related IT risks (Appendix C, answer 7). He noted, though, that the access control aspect of SAM does not carry over well to e.g. consulting firms, where software usage is more dynamic (Appendix C, answer 7).

By contrast, one participant stated that SAM is limited for access control purposes:

> ACL or Whitelisting require kernel module, which is quite different from the collection function of SAM. Granularity (folder path, single executable file or MSI package) is also not easy to define. (Appendix G, answer 7)

Likewise, one participant answered that when it comes to dealing with access control and authorization management SAM is not really applicable:

> I don't think SAM has a role really, apart from if that access is user license access then they need a license and for that there should be separate policy and security for that person...This might be a job id control to determine that a person with a particular job needs access due to a certain system and makes it easy for them to get it. I don't think that's really a SAM perspective. (Appendix H, answer 7)

This echoes what yet another participant wrote about SAM, stating that it is not a means to ensure proper access control, but that additional measures must be enforced in organizations (Appendix I, answer 7). Interestingly, one of the participants claimed that SAM might lead to some adverse side-effects:

> I’ve wasted hours on the phone with help-desk, trying to get my access back for basic stuff so in my opinion it’s annoying as hell. I would limit it to user creation and leave the rest to product owners or developers. (Appendix E, answer 7)

5 Analysis and discussion

5.1 Software Asset Management versus IT risks

The empirical results highlight a number of IT risks associated with the poor management of software assets. Specifically, they explore the relationship between SAM and the IT risks outlined in our research to a greater degree. Whilst the theoretical results were more positive in this regard, the empirical results are not as clear-cut about how SAM affects respective IT risk.

5.1.1 Unauthorized or rogue software (Shadow IT)

Both the literature and interview participants are in agreement that SAM could be useful against unauthorized or rogue software. The exact role of SAM in this respect is a bone of contention, though. One of the participants was more adamant about how SAM should manage software assets, citing its ability to whitelist or blacklist all software assets on organizations' networks (Appendix H, answer 5). This reflects a prevailing theme in the literature. It is argued that SAM can mitigate IT risks pertaining to unauthorized software by managing all software assets, which, if left unchecked, could be exploited by attackers (Dempsey et al., 2013).

However, at the same time, the interview participants do not believe that SAM is capable of identifying all software assets. The participant in e.g. Appendix G (answer 6) contends that it can only track a limited amount of software, specifically OS base, drivers, and installed software. This is in contrast with what Vion et al. (2017) write about in their article, which states that SAM can also detect software pertaining to the cloud. Yet our interview participants are hesitant to proclaim that SAM can prevent all software deployments. One of the participants contends e.g. that SAM utilization is only a part of organizations' overall efforts to mitigate unauthorized software (Appendix D, answer 5). This sentiment is shared by the participant in Appendix G (answer 5). Furthermore, the participants present a number of issues with SAM against this IT risk.

Firstly, it depends on whether SAM operations are supported by the organizations' systems and architecture (Appendix F, answer 5). Boyes et al. (2014) argue, by comparison, that SAM provides an accurate overview of all installations and deployments of software assets, including limiting usage to those authorized by IT management. We believe that the discrepancy between attitude here is that software assets and systems have developed in recent years, long after the literature was written. Specifically, many organizations today work in a more fast-paced development environment (as presented in Appendix E, answer 5). They also have different software requirements. The issue of e.g. software-as-a-service is not discussed in the literature but has been brought up as an obstacle to traditional SAM by one of the participants (Appendix H, answer 2).

Secondly, the effect of SAM against unauthorized software also largely depends on whether or not it is used in conjunction with more rigorous information security policies. As we discovered in the empirical results, many of the participants argue that SAM efforts to steer

unauthorized software assets, including those not approved by management, are contingent on additional methods. For this purpose, organizations are more apt to use systems like Endpoint Detection and Response (EDR) or Web Application Firewalls (WAF), as opposed to SAM (Appendix D, answer 4). Yet this also reflects what Tvrdikova (2008) writes about in terms of only using technical tools to manage software assets. We contend that the problem with the existing literature on SAM is that it focuses too much on the technical aspects, and it fails to grasp the importance of organizational components.

To drive the point home, SAM affects unauthorized software assets and rogue software in a limited fashion. It sometimes manages to mitigate the IT risk, but it is limited in this regard unless it is complemented by other methods.

5.1.2 Outdated software assets (Patch management)

The literature and the interview participants are more divided on how SAM affects outdated software assets. In fact, there are even some points of disagreements between the participants themselves.

One of the participants answers that SAM makes a vast difference against outdated software assets (Appendix C, answer 6). Another participant states that SAM could track which software assets have been updated and provide reporting capabilities for IT management (Appendix E, answer 6). This is largely in line with what Ben-Menachem (2005) writes, contending that SAM is an important step to ensure that software has received the latest security updates. It is additionally argued in the literature that organizations lack cost-effective tools in their efforts to manage outdated software assets (Dacey, 2003, as cited in Cavusoglu et al., 2008). The argument, then, is that SAM could provide organizations with this functionality (see e.g. Appendix C, answer 6).

Nevertheless, we discovered that SAM might be quite limited in terms of managing outdated software assets. One participant goes so far as to dismiss it altogether, stating that it should first and foremost be a security concern (Appendix H, answer 6). Similarly, another interview participant contends that SAM can only track limited assets; it does not cover all the various software in organizations (Appendix G, answer 6). This argument against SAM is in line with what is written about risk management methods for detecting outdated software assets. Tvrdikova (2008) writes e.g. that organizations require specialized automatic scanning tools for this purpose. In contrast to what is argued by e.g. Ben-Menachem (2005), SAM is simply unable to gauge software assets' security updates. This is corroborated by one of our participants, who maintains that it is a responsibility of the software vendors instead (see e.g. Appendix E, answer 9).

We contend that the difference in results above is due to the limitations of our study. Interesting to note here is that some of the interview participants are more apt to acknowledge that SAM does not affect outdated software assets. Other participants, however, are more optimistic about it yet speak of it in general terms or as a principle (Appendix C, answer 6). This disagreement is most likely due to the participants' outlook as well as experience of SAM. In other words, it highly depends on whether or not they have previously utilized SAM against outdated software assets. One of the participants, illustratively, stated that he lacked sufficient knowledge of SAM to posit how it might affect the IT risk (Appendix D, answer 6). Indeed, this is a shortcoming of the existing understanding of SAM amongst cyber-security

practitioners and IT managers, as well as the literature, and in turn our study's generalizability.

5.1.3 Data and software integrity

An integral aspect of mitigating IT risks pertaining to data and software integrity is the effective management and control of all software assets. We have already seen that SAM is quite limited in this regard (see chapter 5.1.1 above). Yet one of the participants maintains that SAM could act as a "source of truth" with regard to software, explaining that it is a method to maintain software integrity (Appendix H, answer 8). This notion is discussed by Boyes et al. (2014) when they conclude that SAM detects threats and vulnerabilities of organizations' software assets. It further reflects what Ben-Menachem and Marliss (2004) write about in terms of how SAM gauges software dependencies and the state of organizations' networks.

Conversely, some of the participants disagree with this line of argument. They contend that SAM is limited for this purpose, stating that it is unable to prevent e.g. malicious DLL-sideloading (Appendix G, answer 8). Data and software integrity is a whole new ball game today, one which SAM is not able to manage (see e.g. Appendix F, answer 8; Appendix I, answer 8). This is in line with what Stoneburner et al. (2002) discuss in their article. They write that organizations require rigorous vulnerability analyses of their software assets (Stoneburner et al., 2002). It is also an important part of risk management efforts against IT risks related to data and software integrity, as argued by Oleg and Ekaterina (2017). Notably, SAM does not currently provide this functionality, and it is not something that we have encountered in our theoretical results or during our interviews. We believe that the literature, yet again, is out of date with recent developments of software assets. Most of the literature was e.g. written long before the introduction of recent software vulnerabilities, which were brought to our attention by the participants in Appendix G (answer 8) and Appendix H (answer 8).

5.1.4 Access control

The majority of interview participants disagree with the literature on how SAM affects access control, and subsequently if it can mitigate IT risks associated with this. Notably, two of the participants contend that SAM is useful for access control purposes, especially in large-scale organizations (Appendix D, answer 7; Appendix F, answer 7). This is a similar contention to what Ben-Menachem (2005) writes about SAM and its carry-over to the deployment of software assets. Equally, it is argued in the literature that this aspect of SAM allows organizations to control all software assets, through e.g. whitelisting or blacklisting (Albert et al., 2013; Dempsey et al., 2018; Vion et al., 2017).

Nevertheless, we discovered that the rest of the participants did not believe that SAM has any bearing on IT risks related to access control. As an example, one of the participants above adds that it does not extend to e.g. consulting firms, where software usage is more dynamic (Appendix C, answer 7). Another participant claims that SAM is limited for access control purposes, stating that the access control functionality of SAM does not include granularity, i.e. folder paths, single executable files, or MSI packages (Appendix G, answer 7). Yet another participant states that SAM does not play a role at all and that unless "access is user

license access then... there should be separate policy and security [for this]" (Appendix H, answer 7). This corroborates what Benantar (2006) writes in terms of access control and that various conditions must be fulfilled to install new software assets; or what Jacomme and Kremer (2018) write. Additionally, it echoes the arguments put forward by the participant in Appendix I (answer 7).

However, there is a hitherto not discussed deficiency of our study which may ultimately affect the results. We attribute this to our failure to explain more in-depth the precise mechanism of SAM for access control. As an example, Swartz and Vysniauskas (2013) discuss how the verification and compliance steps of SAM can detect all exceptions to policies, processes, and procedures. Yet this is not something which we outline in the interview definition of terms (Appendix 2). Certainly, there are software components which are not included in this scope, as illustrated in Appendix G (answer 7). But since we do not discuss the access control aspect of SAM in more detail, it is impossible to determine how SAM affects these IT risks. For example, the interview definition of terms does not specify whether the access controls encompass software assets running on the cloud or those provided by third-parties (Software-As-A-Service). In a similar vein, there are certain organizational factors which determine how SAM affects IT risks related to access control, e.g. employees' awareness of IT risks when they install various software assets. This is notably discussed by Giorgini and Paja (2017). Chen (2009) further writes that organizations would err to rely solely on authorization and access controls in order to mitigate IT risks. We, thus, contend that our understanding of SAM for access control purposes presents a limitation to the study; and for this reason, it is difficult to draw a conclusion on how it affects the IT risk.

5.2 Obstacles from both literature study and interviews

Based on the literature study and interviews it is apparent that many factors affect the IT risks outlined in the research question. This presents an obstacle to the way we understand how these IT risks are affected by SAM. SAM might have some bearing on the secure management of software assets, but as outlined in e.g. Appendix D (answer 5) other factors may play a part in mitigating the various IT risks. This is a sentiment shared by Chen (2009) and Giorgini and Paja (2017) when they describe that it is impossible for organizations to rely entirely on one method to mitigate risks. Indeed, a prevailing theme is that organizations may use many methods apart from SAM in their attempts to mitigate the IT risks. As an example, the interviews in Appendix: Part D, E, and G had a very different outlook on how to manage the IT risks compared to e.g. the participants in Appendix: Part C and H. In Appendix: Part D it is stated that whilst SAM might be a good initial step towards ensuring information security, more practical solutions such as Endpoint Detection and Response (EDF) or Web Application Firewalls (WAF) are sometimes required (Appendix D, answer 4). In other words, what could work for some organizations might not hold water in other organizations and how they may utilize SAM against the IT risks.

Furthermore, our participants never fully disclosed the precise correlation between SAM and the IT risks. In a lot of instances, they were talking about SAM as a theoretical concept or how it might behave in principle (see e.g. Appendix D, answer 4). The chief culprit for this is most likely that information security is a confidential matter. In turn, our results were by and large based on generalized perceptions of how SAM affects respective IT risk, and not specific to the participants' organizations, an obstacle similar to that faced by Mbowe et al.

(2014). This could to a certain degree have been avoided if we had conducted a study of SAM more similar to Swartz and Vysniauskas's (2013). Whilst the focus of their study did not encompass IT risks, it would have given us greater insight into how these IT risks are affected in practice without compromising the participants' integrity.

We were, illustratively, forced to redact the majority of the full interview transcript in Appendix E. The reason for this being that we had to meet the approval of the firm's requirements for redistribution. Even though the firm's name was made anonymous (according to the principles outlined in our method chapter) the interview participant expressed concerns that he might risk his security clearance, since the firm deals with classified material. Thus, any linkage to the firm's reputation had to be obfuscated. Clearly, this presented an obstacle to our study and similar reservations were shared by other participants (e.g. Appendix D, Appendix G, Appendix H, and Appendix I). All these participants expressed a desire to be made anonymous. They also stated that they could not comment on any intrinsic details of their risk management efforts and that they were unable to reveal data other than those canvassed in the interview transcripts. The study of IT risks and information security ultimately represents a delicate matter and one that organizations are not keen to disclose openly.

However, whilst this was an obstacle to our study, we did discover that our results corroborated a lot of what is discussed by Klint and Verhoef (2002) in terms of the benefits of managing software assets. This is in line with what e.g. Tvrdikova (2008) and Stoneburner et al. (2002) write about risk management in an organizational context. The majority of participants acknowledged that SAM might have a partial effect in affecting the IT risks (see e.g. Appendix C, answer 6-7). Equally, this is something that the existing theoretical results about SAM as part of risk management efforts have failed to establish.

5.3 Whither Software Asset Management?

As the discussions above (chapter 5.1 and 5.2) demonstrate, SAM is limited in how it affects the IT risks outlined in our research question. Yet this does not mean that SAM is altogether useless. As some of the participants acknowledge, it could potentially complement more rigorous information security efforts (Appendix D, answer 4; Appendix H, answer 4). This is in line with what is discussed by Alnatheer (2015), who states that organizations should manage their software assets so as to enforce compliance with their overall information security policies. Whilst we have seen that SAM is inherently limited in this respect, it could be useful to at least support better management of software assets. This is a contention that is also shared by the participants, see e.g. Appendix F (answer 3) and Appendix C (answer 4).

Furthermore, SAM can be utilized in order to help organizations approve various software assets. This might in turn affect employees' behavior in organizations. One of the participants expands on this idea, and states that it could provide an overarching framework that determines which software assets are permitted on employees' machines (Appendix H, answer 2), much like what is discussed by Albert et al. (2013). Equally, Fenz et al. (2011) contend that risk management efforts should be discussed in terms of how it can improve managerial insight and control of software assets. The interview participants offered no consensus over SAM in this respect, but it is nonetheless a prevailing theme in the literature. As an example, Datta (2010) writes that any technical controls that can help reduce IT risks

should be part of risk management efforts. In this sense, SAM is similar to the automated scanning tools described by Kondakci (2006) or the inventory for software assets discussed by Klint and Verhoef (2002). Consequently, we contend that SAM can be utilized as part of the methods used by organizations to mitigate IT risks, albeit in a supplementary fashion.

6 Conclusion

6.1 How does SAM affect IT risks in organizations?

This paper presented how SAM affects the following IT risks in organizations: unauthorized or rogue software (Shadow IT); outdated software assets (Patch management); data and software integrity; and access control. In particular, it strived to bring the critical discussion of how SAM can be utilized against these risks to a wider resonance. In this endeavor, we attempted to assess the credibility of the claims made about SAM by firms around the world. We subsequently described cyber-security practitioners and IT managers' perceptions on the matter. Based on our study we discussed a lot of factors which may impact the IT risks outlined in the research question, as well as provide a preliminary assessment of how SAM affects respective IT risk.

It is clear that there are many factors which underpin the effective management of software assets and IT risks in organizations. Indeed, even with SAM as the basis for managing these assets, our results show that SAM does not directly mitigate the IT risks, at least not by itself. We thus conclude that SAM utilization only partially affects some of the IT risks, such as detecting unauthorized or rogue software. Tentatively, it could be argued that SAM does have some bearing on outdated software assets, but it is quite limited in how it affects IT risks pertaining to these. The data gathered in this paper contradicts the claims that SAM may be utilized to ensure data and software integrity. This relationship lends itself to other factors, which are currently not canvassed in the existing literature nor by the firms. Moreover, it is not clear how SAM affects IT risks related to access control, but it is certain that there may be more factors at work which in turn affect this IT risk.

To drive the point home, we maintain that SAM has a limited effect in mitigating the IT risks. Yet it is important to bear in mind that the precise correlation between SAM and the risks has not been established. A major difficulty of this study is derived from the fact that organizations face many IT risks, beyond those outlined in the research question. Some of these are mitigated by organizations' internal information security policies as well as various technological tools. It should be noted, though, that some of the IT risks may also be mitigated by improved management and control of software assets. To this end, our paper demonstrates that SAM could be somewhat useful, but that it does not reduce IT risks by itself. We contend that this ambiguity is the result of the obstacles inherent with risk management efforts, and the paper asserts that organizations need a plethora of tools in their arsenal against IT risks, including SAM.

6.2 Future research and limitations

Our study was carried out in an interview context and involved seven different interview participants. However, we did not assess how SAM affects the IT risks outlined in the research question in practice, and in this regard, we have only really scratched the surface. We contend that more research could be done to investigate how these IT risks are affected in organizations.

The next course of action for future research would be to assess the role of SAM from an information security standpoint. More evidence from practical studies would be a great leap forward. In an industry where software assets highly correlate to an organization's operational success, the critical understanding of how SAM can protect software assets would do away with a lot of speculation. Whilst SAM might be utilized as a method to mitigate various IT risks, future research could help probe more precisely how SAM affects these as part of risk management efforts.

Appendix

Part A: Interview questions

1. What are your current responsibilities?
   Vad har du för ansvarsområden i dagsläget?

2. Are there any challenges in terms of IT risks when it comes to managing and controlling software assets?
   Finns det utmaningar avseende IT risker när det gäller att hantera och styra mjukvara?

3. Are you aware of, or have heard about, Software Asset Management (SAM) as a utility for improving information security?
   Är du medveten kring, eller har hört talas om, att använda Software Asset Management (SAM) som ett styrmedel för informationssäkerhet?

4. Do you think you/organizations would benefit by implementing a SAM strategy to mitigate the IT risks you identified above (Q2)?
   Tror du att SAM som styrmedel skulle gynna er/organisationer genom att minimera IT risker som du identifierat ovanför (Q2)?

5. Do you perceive SAM to be effective in detecting unauthorized/rogue software on networks?
   Upplever du att SAM är effektiv för att upptäcka oauktoriserad mjukvara på nätverk?

6. In terms of patch management and ensuring that software assets receive the latest security updates, what role does SAM play when it comes to tracking software assets?
   När det gäller patch management och att försäkra att mjukvara erhåller senaste säkerhetsuppdateringar, vilken roll spelar SAM när det gäller att spåra mjukvara?

7. What is your opinion about utilizing SAM as a measure for access control/authorization management?
   Vad är din åsikt kring SAM som styrmedel för åtkomstkontroll och behörighet av mjukvara?

8. Do you think SAM can ensure data and software integrity through its control of software assets on networks?
   Tror du att SAM kan försäkra integritet av data och mjukvara genom att styra samtlig mjukvara på nätverk?

9. Anything else you would like to add?
   Finns det något annat som du vill tillägga?

Part B: Interview definition of terms

[Authors' notes: The following definition of terms was used to set up the interview and establish a common point of reference for some of the concepts used in this paper.]

Information Security:

- Ensures that the software and networks pertaining to an organization remain secure against unauthorized actors/elements (e.g. hackers, malware).
- Also refers to data integrity whereby data and information are not unduly modified, destroyed, or disseminated (e.g. data breach, leaks).

IT Risk:

- A function of the likelihood of a given threat-source exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization (e.g. an outdated software asset that contains vulnerabilities that might be exploited by external threats, such as hackers).

Risk Management:

- The methods employed to reduce IT risks, such as identifying and evaluating risks on networks and/or systems.
- Can be either technological or organizational (e.g. information security policies).

Software Asset:

- Software which supports the business operations of an organization.
- May refer to either program specific software (e.g. Microsoft Office, Microsoft Teams) or operating systems (e.g. the OS that runs on a computer, the OS that runs on a virtual machine).

Software Asset Management (SAM):

- Refers to the processes and tools which enable organizations to track software usage on their networks.
- Primarily used for licensing purposes whereby organizations ensure that they are licensing compliant.
- SAM may be utilized through product specific families (e.g. IBM or Microsoft products) or it may be utilized as a policy to manage and control all software usage within an organization. We are interested in the latter.
- In particular, some of the components employed by SAM include access control, authorization management, patch management, and an inventory of software assets.

Part C: Interview response Omegapoint Malmö AB

Interview participant: Fredrik Lundbeck, CEO
Date: May 9th, 2019
Location: Malmö, Sweden
Method: In person interview

[Authors' notes: The following interview was conducted in Swedish and italicized text denote the interviewers' words.]

1. Jag är VD för bolaget här i Malmö.

Följdfråga: Och Omegapoint jobbar då med IT-Säkerhet?

Ja, delar av det vi jobbar med är IT säkerhet. 20% av vår leverans är kanske IT säkerhet.

2. Ja, det skulle jag säga.

3. Absolut! Det är ju ett av de bärande elementen i hela idén skulle jag säga, att kunna centralstyra detta.

4. Ja, det tror jag definitivt.

Följdfråga: På vilket sätt?

Annars är det lätt att det blir isolerade data med egna versioner och egna licenser. Så det är jättebra om vi kunde styra när folk slutar och när folk börjar så kan vi skicka ut; Ja men, du ska ha den här eftersom du börjar som programmerare så du ska ha dom här licenserna. Och sen när du slutar kan det ju va så att vissa licenser försvinner med folk då dom slutar, att dom är personliga på något sätt. Och sen finns det ju även andra risker med att folk använder fel versioner, fel mjukvaror och där öppnar man ju upp för sårbarheter på olika sätt genom att man till exempel kör en annan version eller en gammal version, men det kan ju även vara att man kör helt fel [mjukvara], ex. om det är någon programmerings editor så kanske det är någon som hittar någon häftig open-source variant men vi vill absolut inte att man använda det för det kan finnas tydligare sårbarheter i den typen av mjukvaror. Så sitter man och programmerar i den där och så skickar man in den där och så får man med något som inte skulle varit med. Nu spånar jag bara, men det är inte bara patch management, versioner eller snabbt kunna täppa igen eventuella håll utan det kan vara att man kör verktyg med samma funktion men som varken är verifierade eller godkända centralt.

Följdfråga: Så rouge software?

Ja och det kommer kanske längre fram, men det är ju också kanske ett tydligare problem om det är Bring Your Own Device för då kommer kanske någon helt plötsligt med någon Ubuntu, som kanske är jättebra och effektiv att använda men det blir ju problem med att man ska godkänna den då och analysera riskerna i den mjukvaran då.

5. Ja, men kanske inte upptäcka utan mer slippa helt och hållet.

Följdfråga: Så kanske att likna med att whitelista istället för att blacklista?

Ja, men lite mer åt det hållet. Har vi inte checkat i att du ska ha en licens är det ju tveksamt… Alltså var kommer den licensen ifrån? Det är ju klart att folk kan sitta med sina laptops helt utan att vara uppkopplade till vår miljö, med internet via någon telefon. Då har vi ju inte någon kontroll.

Följdfråga: Använder ni någon whitelisting eller blacklisting inom er organisation?

Nej, det gör vi inte, men det tror jag att det skulle vara bra om vi gjorde det. Men det är ju så svårt också, det är ju en konsultverksamhet. Det kommer nya versioner, det kommer nya grejer hela tiden och om man centralt ska godkänna allt så ligger du alltid ett steg efter och du hade upplevt det som väldigt trögt. Jag tror definitivt att det ur ett riskperspektiv att det hade varit bra. Sen så är det snabbhet och flexibilitet kontra risk och så. Stryper man ner allting och inte låter någon göra något är det ju väldigt låg risk. Sen har vi väldigt mycket flexibla arbetare och vi hoppar mellan kundnätverk och vi sitter här och ute på stan och över allt så det är en utmaning i sig att få rätt på det där. Men vissa grejer tror jag att vi skulle kunna köra.

Följdfråga: Er verksamhet kanske inte är helt passande för en så strikt white/blacklisting miljö, men från ditt perspektiv, tror du att organisationer i allmänhet skulle kunna gynnas mycket av sådan här white/blacklisting?

Verkligen! En av våra kunder som innefattar alla sjukhus, där…. Förutom att det kan läcka massa känsliga data finns det ju vissa verksamheter som håller på med saker som skulle vara katastrof om det kom till allmän kännedom. Vi har ju varit inne och gjort ett par studier där och det finns saker och göra. Jag kan inte ge några exempel, och det var länge sedan så det har säkert blivit bättre nu.

Följdfråga: Kanske speciellt för att en av våra kunder kanske inte alla är så ”computer savvy”?

Nä precis. Men det fungerar ju bra ändå, det har ju inte hänt några stora läckor eller så som jag känner till men det skulle ju definitivt, särskilt i stora verksamheter där det finns ett hyfsat standardiserat arbetssätt där du vet vilka verktyg och vilka program som ska vara tillåtna och som behövs för att utöva din profession. Då är det nog ganska bra om det är ganska ner strypt.

6. Den har en jättestor roll skulle jag säga. Mycket stor roll. Jag tror det är ett väldigt effektivt sätt att göra just dom sakerna du nämner.

7. Som jag varit inne på lite tidigare i diskussionen tror jag det är ett bra sätt att styra åtkomst och behörighet för diverse olika funktioner samt åtkomst till olika mjukvara och verktyg. Jag tror också att det beror en del på vad det är för verksamhet. Är dina arbetsuppgifter hyfsat standardiserade såvida att du på ett enkelt sätt kan definiera vilken mjukvara du behöver tillgång till och vilken behörighetsnivå du ska ha och vilka nätverk… alla såna där saker. Om du kan göra det ganska enkelt så tror jag det är jättestor…. Alltså. Men sen på andra sidan har du en konsultverksamhet där du ska jobba med massa olika verktyg. Då kanske det inte passar

så bra. Men just i en sån här verksamhet där vi då förutsätter att det passar väldigt bra så skulle jag säga att man kan göra riskminimeringar och det kanske till och med då finns andra effektivitetsvinster i hanteringen av licenser och patcher. Man kan göra det centralt helt enkelt istället för att man ska sitta och göra det lokalt. Så jag tror det finns riskminimering och effektivitetsvinster och sen vad den vinsten är, om det kan översättas till pengar, det vet jag inte.

8. Ja, det tror jag absolut.

Följdfråga: Hur känner du att utmaningar med Bring Your Own Device påverkar det här?

Det finns många aspekter på det där och jag tror att en svårighet är att ha en central, gemensam eller universal autentiseringsmekanism. För arbetsstationer så kan man ju ha ett Active Directory eller någonting så att du verkligen [kan se att]: ”Ja du är en autentiserad användare, vi vet att du har loggat in här och vi vet att det är du”. Sen om du kommer med en två år gammal Android surfplatta, hur autentiserar du det? Det finns ju lösningar så klart, men jag har ingen enkel lösning på hur man enkelt autentiserar det. Det går säkert att göra genom någon webbinloggning eller kanske kan använda Facebook eller multi-factor authentication.

9. Nej, jag kommer inte på någonting nu direkt.

Part D: Interview response cyber-security venture firm

Interview participant: Anonymous, COO
Date: May 11th, 2019
Location: Tokyo, Japan
Method: Skype interview

1. Chief Operations Officer (COO) in a small cybersecurity venture firm based in Tokyo.

2. Yes, a lot. Difficult to prioritize IT risks. Especially me and my clients we have a lot of difficulties in terms of how to prioritize which is more seriously important: IT risks, cyber-crimes, APT ransomware, or internal fraud. So, they're trying to find out some system that covers all these risks. But in reality, you don't find e.g. SAM effective for all IT risks.

3. In my country, I think there's a really popular product called LanScope Cat [a monitoring and asset management software].

In recent years, it has advertised itself as an endpoint system. This thing wasn't originally an endpoint system, though. But because of the popularity of EDR they're trying to shift from marketing itself as a SAM oriented product to an EDR oriented product. But the initial purpose of using this popular product was part of SAM.

4. I think it's very effective for big firms or corporations to manage and order information security risks. But when it comes to cyber security it's limited. I think when it comes to information security it's pretty effective. This is the first step that corporations should think of when introducing their information security policy. But, if you really want to defend your system against serious cyber-security risks like APT threats (sorry if I keep saying this because I am a cyber-security professional) ... I mean, no cyber-security professional would recommend SAM. I mean, [no one] would say that introducing SAM will solve all cyber-security risks, especially defending themselves against APT. That's why our clients try to introduce systems like EDR or WAF or something like that, depending on the IT risk they wish to protect themselves against. So, I think it's a first step but not the last. SAM doesn't cover all the IT risks. It's a good starting point.

The problem for small or medium companies is that I don't think SAM is effective at all. In terms of their operational perspective I don't think they have enough assets to implement information security policies through SAM. They tend to ignore as long as they can. Based on my experience, that's why it's very difficult to convince executives of small and medium companies of the importance of information security risks. They don't give a [expletive removed], to be honest.

5. I think it depends. I think usually before introducing any IT security products they usually need to formulate their internal security policy first. And then get management to introduce relevant information security system inside the departments. So, it depends how they want to defend their internal security system.

I suppose talking about big organizations that SAM related products are pretty effective. Even in my previous company that I worked for [company name redacted] they would recommend SAM to their clients. But again, there's no perfect method to defend the internal system

against cyber-security risks. In reality, if you want to defend the system you usually have to combine different kinds of security systems. And then operate [them together]. Because, I mean, introducing these systems all at once is much easier than actually making them work together. Combining these different systems into one unified system like SAM is pretty hard.

As a principle, it works. And it's effective in theory. But in reality, you have to think about a lot of things in terms of the effectiveness of the product. So, it's difficult to answer the question. When we think of unauthorized attacks we tend to think about APT (because of the nature of my work). To be honest, I don't think that SAM is effective to take on unauthorized attacks. It's sort of a waste of money here. But there's no alternative product, so it's not like I want this and then this product will protect our system perfectly. In reality, clients usually say that every product has pros and cons. And there's no ideal. Maybe product A will be better than product B, and also cheaper, so for the next two years let's use this. But they don't really have high hopes for product A due to limitations. Usually before implementation they verify whether the product can actually detect, like visualize and simulate daily operations. Not by asking the company, but the third party [supplier of SAM].

6. I don't think I have enough knowledge of SAM to comment on that question.

7. I think it's a good idea. To some extent, though. Nothing is perfect. If you work for big organizations, you have to obey to their policies. Especially in my country [Japan] you're not allowed to install different software by bypassing e.g. SAM. Everything is actually controlled according to SAM. Maybe the information security department can bypass [the policies] since they're in charge of managing IT risks.

But concepts like BYOD is not allowed in my country, I believe, to avoid the associated IT risks. This is also part of a serious problem when you talk to information security departments of big organizations in my country. They were having difficulties managing private devices of employees being used during work time. Security professionals want to avoid stupid mistakes, like malware. But employees don't really understand. So, they try to consolidate software assets or educate the employees by telling them about SAM or more effective SAM policies. It's becoming successful, but still there are some employees trying to bypass. And then this is considered part of internal IT risks. Security professionals always think in terms of how to educate employees how BYOD poses risks from outside. So, they only support use of company computers and devices. If employees only use company authorized devices, then the security department has confidence in their efforts. In this regard, SAM is effective. SAM verifies proper use. Security guys can also control and verify if software is legitimately downloaded. Checking private phones and PC is voluntary, though. You can't really check their phones and PC unless they got infected. In this case it's also effective. SAM is a good starting point.

8. Yes.

9. No

Part E: Interview response defense and cyber-security firm

Interview participant: Artur R., cyber-security consultant
Date: May 16th, 2019
Location: Sydney, Australia
Method: Skype interview

[Authors' notes: The firm is one of the largest defense and cyber-security firms in the world. It specializes in "security and resilience" as well as provides analytics for intelligence-grade security and financial crime work, amongst other things. Due to legal reasons, we were not permitted to share the full interview transcript. The text below represents only a summary of the interview and has been green-lit for redistribution by the firm.]

1. Taking care of the cloud infrastructure, product development and client consulting.

2. Managing codebases, deployments, costs and user access.

3. Yes.

4. Not really.

5. Only if an organization has a fixed number of software applications. In a fast-paced development environment, it should be more of a guideline than a strict process.

6. It could track which software has been updated and provide reporting capabilities for business owners.

7. I've wasted hours on the phone with help-desk, trying to get my access back for basic stuff so in my opinion it's annoying as hell. I would limit it to user creation and leave the rest to product owners or developers.

8. Yes.

9. Currently, this kind of functionality is provided by third-party anyway, e.g. cloud providers take care of patches, hardware/OS vulnerabilities and asset life-cycles.

Part F: Interview response ATEA AB

Interview participant: Mika Koivisto, senior IT-security consultant
Date: May 16th, 2019
Location: Malmö, Sweden
Method: Skype interview

[Authors' notes: The following interview was conducted in Swedish and italicized text denote the interviewers' words.]

1. Jag arbetar som seniorkonsult inom Säkerhet med inriktning Säkerhetsanalyser, dvs. Penetrationstestning, Sårbarhetsskanning och Säkerhetsrådgivning. Mycket teknisk säkerhet som inriktning.

2. Ja absolut! Det är otroligt viktigt att ha kontroll och inventering av mjukvara, både på servrar och klienter då det är väldigt vanligt att dessa innehåller sårbarheter som angripare ofta använder för att utöka sina rättigheter eller ta kontroll över underliggande serverplattform eller klient. Utan en kontinuerlig kontroll och hantering av mjukvara utsätter företag sin IT-miljö för stora säkerhetsrisker.

Man har mjukvaran på sin IT-miljö som inte är säkrade. Man glömmer att patcha dem och så vidare. Och det är därför jag tycker det är otroligt viktigt att ha kontroll vem och vad som får installeras på en maskin. Och de är ju återigen då policys, som går hand i hand med informationssäkerheten. De måste finnas policys för allt det här, vilken mjukvara ska vara tillgänglig i våra IT-system och de är framförallt den punkten där SAM kommer att bidra med att identifiera och hålla kolla på mjukvara.

3. Ja, många aktuella säkerhetsramverk inom informationssäkerhet och IT-Säkerhet har SAM som ett krav för att upprätthålla en säker IT-miljö.

Som begrepp är jag kanske inte så bekant med SAM men just Asset managment har jag ju hört. Det är otroligt viktigt, det är ju de vi gör när vi utför säkerhetsgranskningar enligt regelverket. Så är de ju en stående punkt att kontrollera hur hanteras mjukvara i nätverket och den är väldigt högt rankad också i säkerhets ramverken. Jag tycker de är väldigt viktigt att man pratar om SAM för de är en ganska förbisedd aktivitet hos kunderna. Man tänker inte så mycket på att de inte är så förenat med säkerhetsrisker att inte ha kontroll på sin mjukvara.

4. Ja absolut! Det gynnar givetvis organisationer. Det bidrar med mycket, samtidigt måste man knyta ihop de med den övriga miljön. Jag vet inte om man får så mycket värde om man ska satsa på informationssäkerhet eller säkra IT miljön, och så gör man SAM men inget mer. De är ju en viktig punkt men man kan inte bara använda SAM som begrepp för att styra alltihop. Man kan inte bara förlita sig på SAM utan de finns så mycket mer som kan vara brister. Ett vanligt ramverk vi använder är att vi identifierar enheten, prioritet 1. Hur ansluts enheten? Och sen har vi mjukvara som nummer 2.

5. Ja, det är den ju om den är korrekt implementerad. Det krävs ju styrning så att man tvingar detta på sin IT-miljö, genom styrning av group policy och annat. Det beror ju på implementeringen också. Det kan ju också vara så att det inte är möjligt om de inte har den funktionen, jag tänker på om det är olika operativsystem, med olika arkitekturer och

processors, så kanske de inte är möjligt. Hur ska man upptäcka oauktoriserad mjukvara på en gammal stordator? SAM måste ju spana över alla operativsystem och man ska ha krav på den produkten man satsar på.

Följdfråga: Använder ni er av whitelisting eller blacklisting inom er organisation?

Absolut det är jätteviktigt i en IT-miljö med Windows maskiner. Att man har skrivit in de in sin policy att de här programmen får köras på våra maskiner och att man förhindrar allt annat. Jag kontrollerar ju detta eftersom jag jobbar som säkerhetstestare. De är ju dessa åtkomstkontrollerna man vill kringgå. Kanske till och med både och, whitelisting på sina servrar och blacklisting på sina brandväggar.

6. En korrekt implementerad SAM kan dagligen hämta in information om installerad mjukvara på alla enheter samt kontrollera dessa mot sårbara versioner. Det är också viktigt att SAM knyts ihop med övriga IT-processer så att åtgärder tas. Till exempel genom att larm sätta vid upptäckt samt påtvinga en avinstallation eller genom förhindra mjukvaran från att startas.

7. Det här kan ju användas när tex en anställd börjar på ekonomigruppen och man kan ge personen rätt privilegier. Det är ju klart att man behöver ett sådant mjukvarusystem, jag tyckte att de nästan var underförstått. Det är ju något man behöver för att hantera mjukvara.

8.SAM kan vara en del i denna hantering, men integritet kan troligen inte försäkras enbart genom SAM. Det ställer även krav på mjukvaran och sättet informationen lagras.

Ja, om då den här mjukvaran har gått igenom en kontroll av en leverantör, den leverantören garanterar att produkten är säker. Då tror jag den kan hantera en del i den hanteringen. Det är ju så med integritet, att de finns ingen som kan missbruka den eller förändra information. Men kanske inte endast med SAM utan det måste ju finnas andra kontrollfunktioner. Så de jackar ju in i det här SAM så att de ska kunna fungera effektivt.

9. Nej, det tror jag inte.

Part G: Interview response cyber-security practitioner

Interview participant: Anonymous
Date: May 19th, 2019
Location: London, England
Method: Email correspondence

[Authors' notes: The interview participant works at a leading cyber-security firm. He has previous experience working as a malware reverse engineer/analyst.]

1. CSIRT handler.

2. Organisations often do not know which PC has outdated software installed. Sometimes outdated means vulnerabilities, such as Java, Flash Player, etc.

3. Yes, I used OCS Inventory NG and it has a difficult interface. I tried Facebook osquery, but it is not sufficient.

4. Yes, definitely.

5. For Shadow IT risks, yes. For supply chain attack risks, no.

6. Limited role. In my honest opinion, SAM can only track limited assets, only first 3 kinds below [sic] but not the rest:
- OS base
- drivers
- installed software
- portable software
- downloaded executable
- managed codes / dot net
- browser extensions, activex
- java app, flash / AIR app, HTML5 app
- powershell, WSH, batch scripts

7. ACL or Whitelisting require kernel module, which is quite different from the collection function of SAM. Granularity (folder path, single executable file or MSI package) is also not easy to define.

8. Yes, but quite limited. For example, SAM won't prevent malicious DLL-sideloading.

9. Forget SAM, I focus on software threat hunting tools.

Part H: Interview response transport and logistics firm

Interview participant: Stuart T., IT asset manager
Date: May 17th, 2019
Location: London, England
Method: Skype interview

[Authors' notes: Italicized text denote the interviewers' words.]

1. So, current responsibilities are to run the Software Asset Management business unit within [company name redacted] where I'm currently working. At the moment we are currently building up the process to get a global view, because historically it was maybe a bit disparate and sort of everyone around the place doing their own little bit and it's to bring it into a global view. And that's kind of to manage, control and protect the software assets across the estate which would include the management of the risks rising from use of software; that's a financial risk or commercial or reputational obviously if you're in a non-compliance position.

2. Well, generally speaking, yes there are always challenges and risks. In what respect are you kind of asking this question? Just to get some clarification. Like is there unauthorized software installation, unauthorized purchase, managing things like that?

Follow-up question: Just in general, what's the IT risks that are associated with the controlling of software assets.

Obviously, there are compliance risks when you're over-deploying a software when you haven't got enough licenses, making sure you don't do that I'm putting controls in place to stop that from happening. There are risks around using unsupported software or software that is not approved to be on the estate you are working on for whatever reason.

Follow-up question: Are there more cyber-security risks that are associated?

Yes, there is. If you're unable to control what's installed on the estate there might be risky software. You can get admin rights software that overrides admin rights. You can get IP blockers that mask IP addresses and things like that. So, you don't know what people are looking at on the estate. You can also get to torrent software, you know, peer-to-peer. Obviously, you need to control that and minimize that and enable discovery and a quick resolution to remove those types of risks on the estate where they arise. If you don't have proper controls on, you know, admin rights on machines not setup properly so that end users can just install what they like on their computer. And it doesn't go through any security checks with the security department to verify that that software is actually safe. Also, I'd suggest with software-as-a-service models these days are problematic because there are data security issues there, I think. You don't know where your data is being stored because generally with software-as-a-service it's stored somewhere else not within your network, so you haven't got control. You don't know where it is sometimes; you don't know how secure that is. That tends to be a quite big one as well these days. And interestingly enough, you don't need admin rights to use software-as-a-service products you can generally buy it and you can create an account and you can just use that software over the internet and usually store the data in whatever you're working on with the external parties which means you have no control over that.

3. I have, yes and in a previous world we basically started… So, I don't know if you're aware of the National Institute of Standards and Technology, which is a US based organisation (NIST). They've got quite a good vulnerability database so for every bit of software listed, if they discover any vulnerability, they list them for all versions. So, when you go on NIST, someone's estate can have a look and see what's installed and then highlight specific vulnerabilities to that organisation. It might be that the software is old and no longer supported by that vendor. That's one of the positive ways of doing that, I think. Specifically, on what's installed on the estate generally, rather than behavioural issues of people not securing stuff, not locking their machines down and things like that. People tend to think that they install software and it's safe, but that's not necessarily the case depending on the support level that the vendor supplies. Obviously, that kind of information is quite good to get when making a Software Asset Management policy because you can argue that you shouldn't be using software that's older than two versions or things like that because of if you use "n-3" version these are the vulnerabilities and the software is not supported any more so they are not necessarily getting fixed; there is no hot fixes or security patches being produced so you should use the most recent version because the patches are in there as part of that software installation.

4. Well, yes. I would say that Software Asset Management should be part of a security strategy. Not necessarily a SAM strategy. I'd place it as a security strategy and SAM as adding to that and show with data that these are the reasons why we need this kind of strategy: to protect ourselves from malware or whatever.

Follow-up question: I see, we worded it as a SAM strategy, but do you think that it's more like SAM should be a part of the IT security strategy?

Absolutely.

5. Yes, absolutely. It should be the main way really. I would say you have a whitelist which is supported software that's allowed on the estate and everything else is blacklisted. That's how I would see it. You know, there is no reason why something blacklisted can't find its way into the whitelist if there is a business requirement and if it's gone through the appropriate approval process to become a supported software item in the catalogue. But that should be a Bible really to make sure it's properly managed, and again I think there's deeper dives you can do. We used to do file path and executable name searches across whole SCCM databases on keywords that will pull out information that may or may not be security issues with something that says cracked, something that says torrent something that says serial or that says portable or password crack/password remover. There are always ways to get things installed on computers and this is a good way of trolling that information, to pull out that and investigate or supply to security guys so that they can investigate that kind of information, which is kind of an operational SAM and it is necessarily not the bread-and-butter. SAM is more about the compliance, control and governance of software assets. This is an offshoot of what's installed, and people can install and use.

6. I personally do not think that Software Asset Management should be part of that function to be honest with you. That is a purely security one and going back to the security policy that's created where that has to be done. Maybe Software Asset Management, it's tool set and its data can be used to say "Okay, well we've got all this this software with that version installed there. Have all of those been patched?" It's kind of supplying data rather than having any

ownership or responsibilities to do patch management to be honest with you. And again going back to making sure you use software that is supported by the vendor again, there's stiff policy that says we can't have software installed on the estate from a particular vendor if it's older than three versions because we know that it will not be supported. That would be more of a Software Asset Management function, to say that you have to upgrade to the newest version well. Patch management, to be honest, is something that I've not been involved with and I've worked in SAM for over 15 years and it's not part of the work that I've ever had to deal with in my many rolls. Except that there is a data source about the Software Asset Management team managers which is the inventory of all the assets on the estate and as long as that is a trusted source of data and you got up to date inventory coming in for all the assets then that could be used. But then again, you would give access to the patch management team for that information. I'd imagine now that there is technology that they'd be using to be able to ensure that they are doing what they need to do.

Follow-up question: So, SAM should not be seen as the main tool to perform patch management?

No, I don't think so. It can supplement it and assist it.

7. So again, I would say that access control and authorization management, if you're just talking about how people access systems and things like that... I don't think SAM has a role really, apart from if that access is user license access then they need a license and for that there should be separate policy and security for that person, like "Are you the right person?". This might be a job id control to determine that a person with a particular job needs access due to a certain system and makes it easy for them to get it. I don't think that's really a SAM perspective and I don't even think we would touch any of that at all to be honest with you. Now if you're talking about access to the end user device and should somebody be able to install software onto their computer absolutely not, but for the bigger systems and getting admin rights for servers, that's a job role rather than something SAM should interact with.

8. Yes, I do. I use the term "Single source of Truth", a trusted source of truth and it should be at tool that SAM manages that inventory's assets and knows the coverage of the estate, or how many assets an agent has installed and manage that. That would be specific tools like Flexera and Snow Software or SPEAR and things like that. I would say that there are complimentary tools that need to be used to verify that data like SCCM, Qualice, McAfee things like Eternity. They supplement and you can kind of self-audit, say that all those machines say they've got ten products installed on our tool and then you can verify that by an audit process: "Yes, actually, our SCCM tool reported exactly the same so it's trusted." Also with the integrity is to have a DML, a Definitive Media Library, which is where you store and manage things like media, so that particular application has been packaged in SCCM but you got that particular exact media stored in a safe, secured location and the DML:s can help with commercial licenses, media, business applications, release packages, patches and a lot of stuff. It could be a federated model where patch management aim to patch build, or a team aims to build and then the Software Asset Management team aim the media and licensing for commercial off-the-shelf software. That's how you could maintain software integrity on what's used on your estate because it's from one source and it's a safe secure source.

9. Not really.

Part I: Interview response energy production firm

Interview participant: Anonymous, Head of Commercial IT and IT-security
Date: May 21st, 2019
Location: Stockholm, Sweden
Method: Email correspondence

1. IT-chef för drift av kommersiell IT samt IT-säkerhet

2. Med den omvärld vi har där 0-days och även äldre sårbarheter ständigt uppstår i olika typer av mjukvara finns det stora risker med att hantera mjukvara. Utmaningarna består i att se till att systemen är uppdaterade men även att de är kompatibla med den mjukvara som körs på systemen.

3. Ja det är ett begrepp jag är medveten om

4. Jag ser att SAM är en del i den övergripande bilden gällande IT-risker, enbart SAM som styrmedel ser jag inte som tillräckligt utan det behövs kompletteras med ytterligare system som till exempel IDS,IPS och SIEM lösningar. En mycket viktigt faktor är även att ha kompetent personal som hanterar systemen.

5. Kan tyvärr inte gå in på denna fråga

6. Här kan SAM absolut vara till nytta och hjälp för att få en bild över vilka versioner som körs och även hur status är i förhållande till senaste patchar som är släppta. Jag tror dock det är mer tillämpbart för ”mindre” program. De större aktörerna såsom Microsoft, Oracle och Linux har redan processer för att hantera patchning och därmed säkerställa systemen är uppdaterade.

7. Jag ser som jag nämnt tidigare SAM som en del i den övergripande säkerheten. Jag ser inte det som ett primärt sätt att hantera behörigheter. Även om en person skaffar sig felaktiga behörigheter för ett system ska det finnas ytterligare barriärer som förhindrar.

8. Gällande data är jag tveksam, kan säker vara tillämpbart inom vissa områden. Men igen anser jag att det behövs ytterligare skydd för att säkerställa data och mjukvara inte lämnar företaget eller kommer in till företaget

9. Nej

References

Albert, B. E., Santos, R. P., & Werner, C. M. (2013). Software ecosystems governance to enable IT architecture based on software asset management. *2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*. doi:10.1109/dest.2013.6611329

Alnatheer, M. A. (2015). Information security culture critical success factors. *2015 12th International Conference on Information Technology – New Generations*. doi: 10.1109/ITNG.2015.124

Alpcan, T., & Bambos, N. (2009). Modeling dependencies in security risk management. *2009 Fourth International Conference on Risks and Security of Internet and Systems*. doi: 10.1109/CRISIS.2009.5411969

Backman, J. (2016). *Rapporter och Uppsatser*. Lund: Studentlitteratur.

Benantar, M. (2006). Access Control Systems. [electronic resource] Security, Identity Management and Trust Models. *Springer Science & Business Media, Inc*. Available at: http://search.ebscohost.com/login.aspx?direct=true&db=cat07147a&AN=lub.4038611&site=eds-live&scope=site [Retrieved May 7]

Ben-Menachem, M., & Marliss, G. S. (2004). Inventorying information technology systems: supporting the "paradigm of change". *IEEE Computer Society*, 21(5), 34-43.

Ben-Menachem, M. (2005). IT assets – control by importance and exception: supporting the "paradigm of change". *IEEE Software*, 22(4), 94-102. doi:10.1109/MS.2005.99

Boyes, H., Norris, P., & Watson. T. (2014). Application of asset management in managing cyber security of complex systems. *Asset Management Conference 2014*. doi: 10.1049/cp.2014.1028

Cavusoglu, H., Cavusoglu, H. & Zhang, J. (2008). Security Patch Management: Share the Burden or Share the Damage? *Management Science*, 54(4), 657–670. doi: 10.1287/mnsc.1070.0794.

Chen, T. (2009). Information security and risk management. *Encyclopedia of Multimedia Technology and Networking, 2nd edition*. Available at: http://engweb.swan.ac.uk/~tmchen/papers/info-sec-risks.pdf [Retrieved May 5]

Datta, S. P. (2010). Risk management process for information security system. *International Journal of Computer Science & Communication,* 1(1), 33-38.

Deloitte (2015). Minimizing the threat landscape through integration of Software Asset Management and Security. Available at: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/risk/us-aers-sam-security-whitepaper-042815.pdf [Retrieved April 2]

Dempsey, K., Eavy, P., Goren, N., & Moore, G. (2018). *Automation Support for Security Control Assessments: Software Asset Management* (Vol. 3, NISTIR 8011, Rep.). National Institute of Standards and Technology. doi: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8011-3.pdf

Dempsey, K., Eavy, P., Goren, N., & Moore, G. (2017). *Automation Support for Security Control Assessments: Volume 1: Overview* (Vol. 1, NISTIR 8011, Rep.). National Institute of Standards and Technology. doi: https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8011-1.pdf

Fenz, S., Ekelhart, A., & Neubauer, T. (2011). Information security risk management: in which security solutions is it worth investing? *Communications of the Association for Information Systems*, 28(1), 329-356.

Flexera (2016). How Security Risks & the Shift to the Cloud Are Transforming SAM. Available at: https://resources.flexera.com/web/pdf/WhitePaper-SLO-Security-Risks-Cloud-Transforming-SAM.pdf [Retrieved April 2]

Gartner (2011). IT Asset Management: It's all about process. Available at: https://www.gartner.com/imagesrv/media-products/pdf/provance/provance_issue1.pdf [Retrieved April 2]

Giorgini, P., & Paja, E. (2017). Information security risk management. *The practice of enterprise modeling: 10th IFIP WG 8.1 Working Conference*, 18-33. doi: 10.1007/978-3-319-70241-4_2

Henttinen, H. (2018). Improvement of information security management system in media x corporation (Master's thesis). Available at: https://www.theseus.fi/bitstream/handle/10024/151489/Henttinen%20Harri.pdf?sequence=1&isAllowed=y [Retrieved April 4]

Holsing, N. F., & Yen, D. (1999). Software asset management: analysis, development and implementation. *Information Resources Management Journal (IRMJ),* 12(3), 14-26.

Ioannidis, C., Pym, D., & Williams, J. (2012). Information security trade-offs and optimal patching policies. *European Journal of Operational Research*, 216(2), 434–444. doi: 10.1016/j.ejor.2011.05.050.

Islam, S., & Falcarin, P. (2011). Measuring security requirements for software security. 2011 IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS). doi: 10.1109/CIS.2011.6169137

ISO (International Organization for Standardization). (2015). Software Asset Management (ISO/IEC 19770-5:2015). Available at: https://www.iso.org/standard/68291.html [Retrieved April 2]

Jacobsen, D. (2002). *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Lund: Studentlitteratur.

Jacomme, C., & Kremer, S. (2018). An extensive formal analysis of multi-factor authentication protocols. *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. doi: 10.1109/CSF.2018.00008

Jakubicka, M. (2010). Software asset management. *2010 26th IEEE International Conference on Software Maintenance*. doi: 10.1109/ICSM.2010.5609662

Kelleher, E. & Greene, A. (2018). Assuring Data Integrity and Data Privacy Compliance when using Software-as-a-Service (SaaS) in the Life Science Sector. *Journal of Validation Technology*, 24(6), 1–2. Available at: http://search.ebscohost.com/login.aspx?direct=true&db=ccm&AN=133917760&site=eds-live&scope=site [Retrieved April 29]

Kirk, J., Miller, M. L., & Miller, M. L. (1986). *Reliability and validity in qualitative research*. Sage Publications.

Klint, P., & Verhoef, C. (2002). Enabling the creation of knowledge about software assets. *Data & Knowledge Engineering*, 41, 141-158.

Kondakci, S. (2006). A remote IT security evaluation scheme: a proactive approach to risk management. *Fourth IEEE International Workshop on Information Assurance*. doi: 10.1109/IWIA.2006.1

KPMG (2009). Software Asset Management: Mitigating Risk and Realizing Opportunities. Available at: https://i.forbesimg.com/forbesinsights/StudyPDFs/KPMG_SAM.pdf [Retrieved April 2]

KPMG (2013). Is unlicensed software hurting your bottom line? Compliance trends and practices to increase revenue. Available at: https://info.kpmg.us/content/dam/advisory/en/pdfs/risk-assurance/unlicensed-software-increase-revenue.pdf [Retrieved May 6]

KPMG (2018). Software Asset Management (SAM) Rediscovered. Available at: https://assets.kpmg/content/dam/kpmg/au/pdf/2018/software-asset-management-as-a-service-factsheet.pdf [Retrieved April 2]

Mbowe, J. E., Zlotnikova, I., Msanjila, S., & Oreku, G. (2014). A conceptual framework for threat assessment based on organization's information security policy. *Journal of Information Security, 5*, 166-177.

McAfee (2013). McAfee Asset Manager Continuous network monitoring for real-time visibility. Available at: https://www.websecurityworks.com/datasheets/ds-asset-manager_new.pdf [Retrieved April 2]

MHRA. (2018). *"GxP" Data Integrity Guidance and Definitions*. Revision 1, MHRA.

Microsoft (2017). A Major Sugar Producer Optimizes Software Asset Management to Help Mitigate Cybersecurity Risks. Available at: https://customers.microsoft.com/en-us/story/mitr-phol-sugar-manufacturing-office-365-en [Retrieved April 2]

Microsoft (2018). Software Asset Management: A new defense against cybersecurity threats. Available at: https://hbr.org/sponsored/2018/03/software-asset-management-a-new-defense-against-cybersecurity-threats [Retrieved April 2]

Mitchell, V. (1996). Assessing the reliability and validity of questionnaires: an empirical example. *Journal of Applied Management Studies*, 5(2), 199-207.

Oleg, M., & Ekaterina, P. (2017). Security and privacy risk estimation for personal data stored on mobile devices aposteriori statistical approach to risk estimation. *2017 8th International Conference on Information Technology (ICIT)*. doi: 10.1109/ICITECH.2017.8079935

Pickard, C., Miladinov, S. (2012). Rogue software: protection against potentially unwanted applications. *Proceedings of the 2012 7th International Conference on Malicious nad Unwanted Software (MALWARE)*, 1-8.

Red Hat (2016). How security and innovation meet at Red Hat. Available at: https://www.redhat.com/cms/managed-files/rh-security-innovation-whitepaper-inc0374232mm-201604-en.pdf [Retrieved April 2]

Rotella, P. (2018). Software security vulnerabilities: baselining and benchmarking. *2018 IEEE/ACM 1st International Workshop on Security Awareness from Design to Deployment (SEAD)*. doi: 10.23919/SEAD.2018.8472847

Saunders, M., Lewis, P., & Thornhill, A. (2012). *Research methods for business students*. Harlow [etc.]: Pearson Education Limited.

Stoneburner, G., Goguen, A., & Feringa, A. (2002). *Risk Management Guide for Information Technology Systems: Recommendations of the National Institute of Standards and Technology*. National Institute of Standards and Technology.

Swartz, J., & Vysniauskas, P. (2013). Software asset management in large scale organizations: exploring the challenges and benefits (Bachelor's thesis). Available at: https://gupea.ub.gu.se/handle/2077/38602 [Retrieved April 4]

Symantec (2019). Solution Brief: Symantec Endpoint Management. Available at: https://www.symantec.com/content/dam/symantec/docs/solution-briefs/endpoint-management-security-productivity-versatility-en.pdf [Retrieved April 2]

Tvrdikova, M. (2008). Information system integrated security. *2008 7th Computer Information Systems and Industrial Management Applications*. doi: 10.1109/CISIM.2008.41

Varela, A. M., Méxas, M. P., & Drumond, G. M. (2018). The scenario of software asset management (SAM) in large and midsize companies. *Independent Journal of Management & Production, 9*(2), 301-320. doi:10.14807/ijmp.v9i2.730

Vion, A., Baillon, N., Boyer, F., & De Palma, N. (2017). Software license optimization and cloud computing. *The Eighth International Conference on Cloud Computing, GRIDs, and Virtualization,* 115-122. IARIA.

Voas, J., & Hurlburt, G. (2015). Third-party software's trust quagmire. *Computer*, 48(12), 80-87. doi: 10.1109/MC.2015.372

Wanderley, M., Menezes, J., Gusmao, C., & Lima, F. (2015). Proposal of risk management metrics for multiple project software development. *Procedia Computer Science*, 64, 1001-1009. doi: https://doi.org/10.1016/j.procs.2015.08.619

Xu, B., Lu, M., & Zhang, D. (2017). A software security case developing method based on hierarchical argument strategy. *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. doi: 10.1109/QRS-C.2017.124

Yeboah-Boateng, E. O., & Boaten, F. E. (2016). Bring-Your-Own-Device (BYOD): An Evaluation of Associated Risks to Corporate Information Security. *International Journal of Information Technology (IT) & Engineering*, 4(8), 12-30.