2019 11th International Conference on Cyber Conflict:
Silent Battle
T. Minárik, S. Alatalu, S. Biondi,
M. Signoretti, I. Tolga, G. Visky (Eds.)
2019 © NATO CCD COE Publications, Tallinn

Hidden Risks to Cyberspace Security from Obsolete COTS Software

Barış Egemen Özkan
Plans Branch Head
Cyberspace Operation Center
Mons/Belgium
BarisEgemen.Ozkan@shape.nato.int

Serol Bulkan
Professor
Marmara University
Istanbul/Turkey
sbulkan@marmara.edu.tr

Abstract: Obsolescence of Commercial Off The Shelf (COTS) hardware and software, with their shorter product life cycles, is one of the major concerns for cyberspace system/service providers. While hardware obsolescence has been widely studied, software obsolescence has received less attention. However, the increased number of cyber incidents globally calls for more attention to the use of COTS software in critical infrastructures and military systems: systems comprising 25+ product life cycles and dominated by sustainment concerns. The number of reported vulnerabilities of COTS software systems more than doubled in 2017 and continued to increase in 2018. It is already a challenge for system/service providers to keep up with the pace of vulnerabilities to sustain the resiliency of the systems. Increased use of COTS software in mission-critical systems exacerbates the situation because it forces system/service providers to manage the risk of not being able to receive security updates for obsolete software. In today's cyber conflict, where hybrid threats are enjoying the highly connected nature of cyberspace terrain enabled with globalization and newer technologies, if cyberspace security risks stemming from obsolete COTS software in critical systems are not addressed properly, they may easily become a national security problem. Such risks must be addressed comprehensively at both governance and management levels. This paper presents the sustainability, operational efficiency and cyberspace security risks of obsolete COTS software in critical infrastructures and military systems and proposes mitigations at both governance and management levels. At the management level, a Multi Criteria Decision Making methodology is proposed for system/service providers to balance the conflicting objective functions of

reaching a cost-effective solution while maximizing the system's cyberspace security and efficiency.

Keywords: *software obsolescence risks, COTS, vulnerabilities, cyber security, sustainment-dominated systems, cyber conflict*

1. INTRODUCTION

Following the end of the Cold War, the small-scale consumer products market share drastically increased and came to dominate the market (Singh and Sandborn, 2006). While military market share of semiconductors in the 1970s was 35% of a $4.2 billion market, it dropped to 0.3% of a $316 billion market in the 2000s (Kelly, 2017). The high speed of technological advances and the ease of access to markets, as well as the dynamic nature of consumer needs, has made product life cycles even shorter, down to 2-5 years (Shen and Willems, 2014). These developments created a new phenomenon called Commercial Off The Shelf (COTS) products (Sandborn, 2008). COTS are hardware and software products or services available in the market for public use (Özkan & Bulkan, 2018) and they are cheaper than custom designed products, usable for multiple environments with well-defined interfaces, most likely available from multiple vendors and usually have faster product upgrade cycles. With these attractive attributes, COTS products became the indispensable choice of system designers to achieve cost-effective solutions.

As well as the many advantages of using COTS products in sustainment-dominated systems, there are also some disadvantages. Sustainment-dominated systems, including military, transportation, aviation and nuclear systems, have an average of 25 years of product life cycles (Özkan & Bulkan, 2016). Military assets such as the B-52 bombers of the US Air Force have been in use since the 1950s (US Air Force, 2015). Using COTS hardware and software with shorter life cycles on such systems generates a sustainability risk if and when the original vendor declares obsolescence or end-of-life/support for those COTS parts.

Although there have been numerous studies on hardware COTS obsolescence, software COTS obsolescence has not been equally studied (Sandborn, 2007), despite the ever-increasing security vulnerabilities of COTS software. There are studies seeking cost-efficient methods to sustain the systems (Wnuk, Gorschek, and Zahda, 2013; Rojo et al., 2010; Munoz et al., 2015; S. Rajagopal, J.A. Erkoyuncu, 2015). The number of reported vulnerabilities on COTS software is increasing (CVEDetails,

2019a). With the current pace of the accumulation of vulnerabilities, owners of non-obsolete systems are already having difficulty patching their systems to ensure cyber security. In addition to this issue, not being able to receive security updates from original vendors at all due to obsolescence leads to a serious silent risk if those vulnerabilities are exploited by cyber threat vectors.

In today's cyber conflict, traditional military threats of armed forces have been overshadowed by hybrid threats, in and through which cyberspace is highly utilized. The whole spectrum of cyber threat actors, including state-sponsored ones, are exploiting the intense digitization and globalization enabled by new technologies; and they are not shy about employing their means to create effects to deny, disrupt and even destruct. Such hybrid threats are challenging classical defense strategies, attribution and deterrence concepts. Increased use of COTS software in National Critical Infrastructures (NCI) and military systems is expanding our vulnerability surface for attackers to exploit, which may become a national security issue if not properly addressed.

In this paper the COTS software obsolescence section provides foundations with definitions of obsolescence in general and software obsolescence, and explains why we continue to use COTS software despite the disadvantages. The next section explains the cyber security risks of using obsolete COTS software, including those stemming from supply chain, and their impacts. This section also provides descriptive findings of several COTS software applications, still in use in numerous enterprises, for which the original vendor has already declared end-of-life and no longer provides support. The following section lays down the recommended mitigations for these risks. In the model section, a methodology is proposed to address the COTS software obsolescence with competing objectives including minimum cost, maximum operational availability and maximum cyber security. This section also suggests a practical approach for the proposed model. The last section sums up with recommendations and conclusions for the cyberspace security risks of COTS software obsolescence on NCI and military systems.

2. WHAT: COTS SOFTWARE OBSOLESCENCE

A. Obsolescence Defined

Obsolescence is the condition of no longer being used or useful, of being obsolete. The state of being obsolete may be voluntary or involuntary (Bartels et al., 2012). With voluntary obsolescence, the manufacturer plans the obsolescence and voluntarily stops support to shorten the repetitive purchase cycles. With involuntary obsolescence,

however, neither producer nor consumer has intentions for such obsolescence (Sandborn and Myers, 2008).

Obsolescence may be due to logistical, functional or technological reasons. Logistical obsolescence is due to loss of ability to procure parts, material, manufacturing or software necessary to manufacture or support a product (Bartels et al., 2012), such as termination of access to software due to digital media obsolescence, formatting, or degradation (Sandborn, 2007). In functional obsolescence, the product still meets the functional requirements of the original design; however, the requirements or environmental factors have changed over time and the functionality that the product meets is no longer relevant. In technological obsolescence, a new product is delivered to the customer due to several possible reasons such as increase in capacity or processing power; it supersedes the older one. This type of obsolescence is the most common in information technology, such as CDs superseding floppydisks and DVDs superseding CDs (Özkan & Bulkan, 2016).

B. Software Obsolescence

While hardware obsolescence is better-known and more studied than software obsolescence, they must be considered together since they tightly depend on and affect one another. Software obsolescence occurs when the original vendor stops support, updates, upgrades and fixes for known bugs, which eventually makes the software unusable for consumers (S. Rajagopal, J.A. Erkoyuncu, 2015).

One of the fundamental drivers of COTS software obsolescence in information technology is the fear of losing market share, as was clearly stated by Bill Gates (Merola, 2006): “The only big companies that succeed will be those that obsolete their own products before someone else does”.

Another major reason for software obsolescence is the fast-degrading quality of software. The quality of software depends on its ability to meet consumer expectations. As consumers can rapidly change their requirements, partly due to the swiftly changing nature of the business environment and partly due to consumers’ lack of ability to specify their requirements clearly upfront, some requirements become obsolete even when the product is still in-house for design and development. For those reasons, vendors tend to deliver products within quicker schedules and issue more frequent updates to mitigate the quality defects of software. After a certain point, it becomes much more viable for vendors to cease support, declare a product obsolete and release a completely new product. This is a good business strategy for vendors, but definitely not for sustainment-dominated system owners who have to keep them up and running for many more years.

C. Why Continue to Use Obsolete COTS Software?

Despite these facts, military organizations continue to use COTS software. Many nations' military procurement strategies strongly support the use of COTS hardware and software over custom solutions. Following US Secretary of Defense William Perry's 1994 initiative, many nations started drifting away from the use of military specifications and began preferring COTS-based solutions (Gansler and Lucyshyn, 2008; Ministry of Defence, 2005; Turkey, 2010; US Navy, 2000).

Microsoft officially ended support for Windows XP in April 2014 after releasing its last major update in 2008. According to Netmarketshare.com (2019), 4.1% of all desktop users are still using Windows XP (around 80 million computers), which means that they are susceptible to security vulnerabilities that will never be fixed by the vendor.[1] Windows XP was 13 years old when it was declared obsolete and it still has more market share than Windows 8, Linux and many Mac OS versions.[2] According to CVEDetails, over 740 Windows XP vulnerabilities remain identified but unpatched (CVEDetails, 2019b). However, many systems, including Automated Teller Machines, schools, police stations, electronic voting machines, transportation systems, airport security systems and even casinos are still using Windows XP. In addition, there have been reports of military systems, including warships, still using this operating system (SpiceWorks, 2017). But the question is: why do individuals, companies and even nations continue to use obsolete software with known vulnerabilities that will never be fixed?

The first reason is that it still works. The complacency created by the software-in-use for many years is one of the major factors to continue with it rather than replacing it. The second reason is the need for efficiency. Hardware and software upgrades go hand-in-hand. However, such upgrades are not always feasible for sustainment-dominated systems and service providers due to the high costs of re-design, adaptation, implementation, re-certification and training. The time and cost implications for organizations to upgrade both software and hardware can become very complex to resolve. In addition, obsolete COTS software that has been in use for a long period may have led to numerous deep dependencies in complex systems and support arrangements. The need for compatibility among the systems forces their owners to continue with obsolete COTS software (Lapham and Woody, 2006).

1 One and probably the last exception for this postulation was the WannaCry incident for which Microsoft issued a security update for Windows XP after the company ended its support.

2 This premise is valid for the data retrieved in January 2019.

3. SO WHAT? THE RISKS OF OBSOLETE COTS SOFTWARE

A. Sustainability Risks

Due to their advantages, COTS software products are widely used in almost every part of our lives; including NCIs, government, military, personal systems. However, with their shorter life cycle, the obsolescence of COTS software at least creates a sustainability risk with significant impact on operations and maintenance costs.

Studies on the sustainability risks of COTS software obsolescence have focused on cost impacts and the techniques in-service to mitigate those impacts (Morris, 2000; Abts, Boehm, and Clark, 2000; Mckinney, 2001; Comella-Dorda et al., 2004; Sandborn, 2007; Rojo et al., 2010). These mitigation techniques include data preservation, managed integration, reengineering, reverse engineering, software license downgrade and redevelopment (Sandborn, 2007). Applying these techniques, however, incurs mitigation, redevelopment, requalifying, re-hosting and media management costs.

B. Operational Efficiency Risks

What is not as much studied as the sustainability risks of using COTS software is the impact on operational efficiency. The reasons for continuing to use obsolete COTS software create operational efficiency risks to the reliability, availability and maintainability (RAM) of the systems.[3] RAM is considered to be one of the major quality metrics of COTS products to measure operational efficiency.

The concept of RAM was developed predominantly for hardware systems, and parameters to measure RAM can easily be found in the datasheets of COTS hardware. Unfortunately, software systems do not enjoy the same level of predictable performance in their datasheets. While there are models to measure software quality with evaluation criteria and quality aspects which also include the RAM of the software (Miguel, Mauricio and Rodríguez, 2014), the features to measure the software RAM metrics are usually stated in the non-functional requirements section of the system specifications. However, they are habitually left blank or weakly specified due to the aggressive market conditions for the COTS software; hence they are rarely if ever tested thoroughly. Consequently, the measurement of RAM metrics for software COTS systems is generally left to the service-in-use phase of the software. Vendors look forward to verification and thorough certification tests by consumers during

[3] Reliability refers to the measure of the probability that failures will occur during operation of a system (PioneerEngineering, 2017). In other words, it is the probability of a system's ability to perform its intended function under defined conditions for a specified time interval without failure (ReliabilityWeb, 2018). Availability is the measure of a system's readiness for operation at a given time under given environmental conditions and is usually measured as point availability (Sebok, 2018). Operational availability is a slightly different term used in military literature to define the ratio of uptime to total time. Uptime is measured by adding standby, mission, relocation, pre-operation tests and operating times. Total time is the sum of uptime and maintenance time, which is composed of corrective and preventative maintenance activities (Pryor, 2008). Maintainability is the probability of being able to repair a system, in other words to perform corrective and preventative maintenance measures in a specified environment within a defined period of time (Sebok, 2018).

operational use and expect them to report the identified bugs and vulnerabilities for vendors to provide fixes via after-sale upgrades.

The downtime for obsolete COTS software is likely lengthy and, under severe conditions, mean time to repair (MTTR) becomes notionally infinite if the obsolete software is tightly dependent on another software component which has been upgraded without backward compatibility. The high figures of downtime and MTTR decrease the availability of the system.

When systems with obsolete COTS software are in use for operations, due to unfixed bugs which have been identified after obsolescence they will often stay longer in downstate or degraded and it will decrease the reliability of the system with decreased mean time to failure and increased MTTR figures. Those systems also suffer maintainability risks due to lack of vendor support to fix the problems that have been identified after obsolescence.

C. Cyber Security Risks

In addition to RAM, one other quality metric for software is cyber security (Altexsoft, 2017). Security refers to the protection of a system from inadvertent or malicious activity that could impair the confidentiality, integrity and accessibility of the data, service or function (Miguel, Mauricio and Rodríguez, 2014). As one of the major drawbacks of COTS software, not being able to fully specify cyberspace security requirements keeps increasing its vulnerabilities.

The number of vulnerabilities on information technologies utilizing COTS software continues to follow an ever-increasing trend. We have seen a conspicuous increase in 2017 with 14,714 identified vulnerabilities, and 2018 did not fall short either with 15,703 identified vulnerabilities[4] (See Figure 1). Both 2017 and 2018 vulnerability threat trends indicate that the scale of threat is increasing on internet-connected and mobile devices. Almost all internet and mobile devices software are COTS (Flexera, 2017; 2018) and pose a significant security risk.

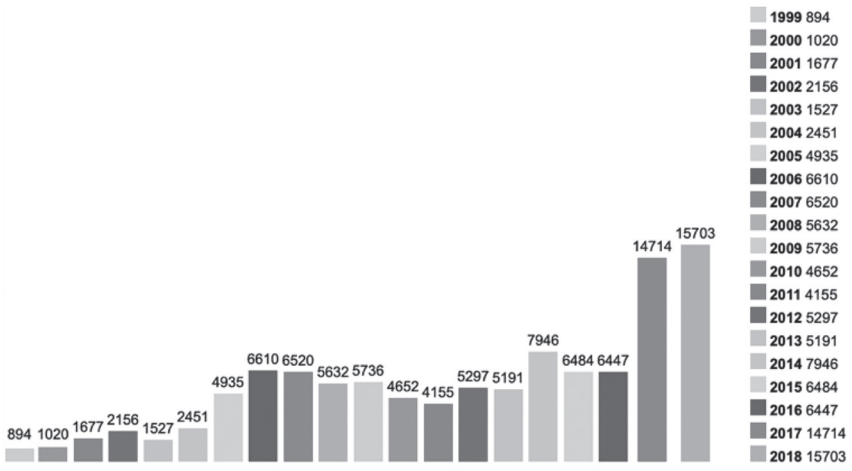
Using obsolete COTS software in systems, and particularly in NCIs, lowers the overall profile of cyberspace security. This is an especially significant concern for vulnerabilities which are discovered after the COTS software is declared obsolete. According to US-CERT, COTS software is risky to use because, compared to custom code, it is a very attractive point of attack: generic, well-known and widely available. Since COTS software comes as a black-box, it is no trivial exercise to mathematically model the security of COTS software and verify it. In addition, COTS software vendors are rarely held liable for direct and consequential damages (US-CERT, 2013).

4 The significant increase is partly due to the actual rise in the vulnerabilities and partly due to enhanced cyber security awareness and maturity in consumers, original vendors and the third-party supply chain. The escalation of such awareness and interest has been an increased incentive to search for vulnerabilities (Flexera, 2018).

Software reuse is an effective strategy using existing working components rather than reinventing the wheel. This strategy brings higher yields, productivity, quality, lower costs and shorter time-to-market (Lee, 2003). It is not rare to see software components that were once used in an older version of software in the newer builds. The component reuse strategy in software design inevitably increases the number of unpatched vulnerabilities on the earlier version of the system at which a vulnerable component from obsolete software is reused in the newer versions. The obsolete software will cease to receive security updates from the vendor and this will increase the probability of exploitation of those later-found vulnerabilities. For example, Microsoft introduced New Technology File System (NTFS) withWindows NT 3.1 and still uses it as its primary file system in Windows 10. Another example is Microsoft's Internet Information Services (IIS) which was first introducedwith Windows NT 3.5.1. and is still used in Windows Server 2019 and Windows 10. Microsoft has declared end-of-life for some of those operating systems and no longer provides updates and patches (Microsoft, 2019).

A thorough survey conducted by these researchers on a particular service provider's approved product list revealed that a number of software applications providing support to critical operational functions are still using COTS software that is already announced as end-of-support by the original vendor. The list includes earlier versions of Adobe Flash Player (v29.0.0.113), Oracle database (11g), Microsoft Office (2007), and Microsoft SharePoint (2007), all of which are beyond their end-of-life and do not receive security patches from their vendors. Readers are advised to refer to many vulnerability exploits against obsolete software in Qualys lists (Qualys, 2019).

FIGURE 1. VULNERABILITIES PER YEAR (CVEDETAILS, 2019A)



With the current pace of the accumulation of vulnerabilities, system owners are already having difficulty patching up their systems. Slow configuration management processes to test patches for safety and interoperability are adding additional difficulty to the timely addressing of vulnerabilities. Finally, except for planned obsolescence, software obsolescence is not very predictable, hence, proactive management strategies developed for hardware obsolescence are not readily adaptable for software obsolescence.

Not being able to receive security updates from original vendors for those known vulnerabilities due to obsolescence leads to a serious silent risk. Especially in today's cyber conflict, cyber threat actors are very talented at disguising themselves in a hybrid threat environment inside highly complex cyberspace terrain. Considering that almost all the published breaches in recent years exploited known vulnerabilities (Gartner, 2017), increased vulnerabilities that will never be patched lead to a high probability of being exploited. The impact of exploitation depends on the system's characteristics. For example, consider a vulnerability stemming from an obsolete COTS software in a maritime harbor, railway or airport system which the military is planning to use during deployment for an operation. Imagine a power plant providing electricity to whole city, a SCADA used in a nuclear power plant or an electronic voting system still using Windows XP. If it is for NCIs, government or military systems, exploitation of those vulnerabilities will easily lead to a national security problem which must not be left alone but instead must be addressed and mitigated thoroughly.

Another risk vector for obsolete COTS software stems from the supply chain. COTS software creates inevitable dependency on certain vendors in the supply chain due to convenience and price advantage. If those vendors are from a foreign country that might be part of possible future conflicts, they can be tempted by foreign intelligence services to create backdoors through silent but deliberate planned obsolescence.

4. NOW WHAT? MITIGATING THE RISKS OF OBSOLETE COTS SOFTWARE

A. Governance-Level Mitigation

Due to the wide spectrum of impacts spanning from personal to national security, the cyber security risks of obsolete COTS software must be addressed with a comprehensive approach at all levels. At the strategic level, a whole-government approach is recommended to provide guidance for obsolescence risks in national cyber security policies. Additionally, while not very trivial, proactive risk mitigation strategies yield more efficiency than reactive models and require a systematic holistic approach as well.

Having reviewed all of the available national defense and cyber security strategies in open sources, it is common for all nations to draw attention to the protection of NCI and the risks of commercialization with intense digitization. However, with a few exceptions, none of those strategies is explicitly referring to the risks of obsolete COTS software on cyber security. Only a white paper on the defense of the Slovak Republic mentions obsolescence for its impacts on the military forces' readiness (Slovak Republic MoD, 2016). While the US Department of Defense's cyber strategy promotes the leverage of COTS capabilities, the US Department of Homeland Security's cyber security strategy explicitly points out the supply chain risks of COTS products.

As nations are becoming more global, connected and digitized, they become more fragile against cyber threats. Considering all the risks of obsolete COTS software mentioned above, it is a much bigger concern for those nations with a larger cyberspace footprint. For that reason, nations are advised to address the risks associated with obsolete COTS software in their cyber security policies, strategies and directives.

In addition, a central cross-domain consultant agency at governance level for public and private institutions may play a significant role to ensure a coherent mitigation approach among government, industry and military organizations. This central agency would also negotiate with original vendors to delay obsolescence, with certain incentives provided by government if needed.

One of the ways to achieve protection of cyberspace as described in national cyber security strategies is through effective relationships and close cooperation with science and technology organizations and academia. It is good practice to exploit the academic relationships and provide them with guidance to improve forecast models for the non-deterministic nature of software obsolescence. Such models will definitely improve proactive obsolescence management strategies.

In the intensely connected nature of cyberspace, where public and private organizations are mutually enabling and supporting each other, governance-level initiatives to enforce cross-domain mapping of all systems in a whole-of-government framework would ease the COTS obsolescence risk management activities at management level through informed assessment, prioritization and resource allocation.

The US Defense Standardization Program Office has been the custodian of a document: "SD-22 – Diminishing Manufacturing Sources and Material Shortages (DMSMS)" (Defense Standardization Program Office, 2016). It is a guidebook of "Best Practices for Implementing a Robust DMSMS Management Program" and primarily intended to provide program managers with increased awareness, robust

obsolescence management processes, metrics for effective measurement and best practices. This document is mainly focused on cost-efficient solutions and slightly touching the cyberspace security risks of obsolete software. Increasing the accounts of the weight of cyber security risks stemming from use of obsolete COTS software and relevant mitigations in this prime reference document for COTS obsolescence will help to increase the awareness among program managers.

Additionally, keeping a definitive list at governance level of all foreign countries in the supply chain that may use software obsolescence as a way to create deliberate cyber security vulnerability in products can be a mitigation for the risks, especially on NCI and military systems.

B. Management Level Mitigation

In order to mitigate security impacts of obsolete software, the UK National Cyber Security Centre (NCSC) recommends to migrate away from them and apply short term mitigations (NCSC, 2017). NCSC recommends using only products still supported by the vendor. When the original vendor declares the obsolescence date, system owners are strongly advised to plan for prioritized migration to newer software. In order to mitigate the security impacts of obsolescence, systems owners are advised to not accept the new developments that will run on obsolete software and to reduce the dependencies on the obsolete software. System owners are also advised to decrease either the probability of the exploitation of the unpatched vulnerabilities or the impact of exploitation if the system is compromised. One method is to prevent malicious code or data access to the obsolete system from outside. This can be done by isolating the system from the enterprise and preventing or at least reducing the system access to untrusted services.

US-CERT recommends practicing a holistic approach to achieve a comprehensive mitigation of the risks stemming from COTS software (US-CERT, 2013). Since there is no such thing as 100% cyber security, it is fair to assume that there will always be more vulnerability than system owners can address. Therefore, the whole cyber security business is based on risk management and it aims to achieve and maintain cyber resilience for better defense, detection, response and recovery with a cognizant prioritization schema.

Mission mapping via thorough asset management is key to a resilience framework. It starts with identifying the COTS components and mapping them to information, functional services, processes and ultimately the critical outputs. The second step is to identify critical points in the mission map for an informed prioritization and cost-effectiveness. Some of those critical points are single points of failures, choke-points, entry-exit points to critical infrastructure, servers interfacing with the outside world,

data centers and core systems for critical mission outputs. Identification of critical points is better achieved through continuing discussions among service providers who own the COTS software and IT infrastructure, the security community and the user community. Once the prioritization is sorted, the next step is to secure both COTS software and hardware to reduce the cyber security risks of using them. Increased redundancy for identified mission-critical points, preferably with different COTS products, is another risk mitigation method. Wrapping the COTS software that is mapped to mission-critical outputs to ensure that it will do only what it is supposed to do is an additional risk mitigation process.

Comparing these two big security organizations, NCSC's recommendations can be considered mostly procedural in order to maximize cyber security. US-CERT's recommendations are much more fit for a mission assurance framework to maximize operational efficiency. In contrast, almost all of the academic studies in the literature, as mentioned above, have focused on minimizing the sustainability cost. All of these approaches are right but not complete without each other. Migration from obsolete COTS software and the selection of mitigation techniques are complex problems for decision-makers. The better solution is a balanced approach to meet all of the objectives with risk informed trade-offs.

C. Balanced Model for Management Level Mitigation

There are at least three objectives for decision: minimize the cost, maximize the cyber security and maximize (or at least sustain) the operational efficiency. The contradictory nature of those objectives makes the decision making more complicated. First of all, high security comes with a significant bill. As we try to minimize the costs by abstaining from certain reactive mitigation measures, we may end up with decreased cyber security. On the other hand, as we try to maximize cyber security we may face with the discontinuity of services and hence decreased efficiency with a considerable impact on mission outputs. Therefore, applying mitigation for COTS software obsolescence is not a single objective decision making process but rather a Multi Criteria Decision Making (MCDM) problem with conflicting objectives.

In MCDM methodology, the decision maker has to use trade-offs and satisfy all objectives with the best effort based on his/her subjective criteria and preferences. The criteria reflect the desires of the decision maker, which points the direction to a better solution (Ehrgott & Xavier Gandibleux, 2002).

The aim of the MCDM is to define a set of candidate solutions in the problem space which will produce representative objective values in the solution space. The latter set in the solution space is called the Pareto Front (Özkan and Bulkan, 2018) and it holds the Pareto optimal solutions.

An improvement in one of the objective functions can only be achieved for Pareto optimal solutions if at least one other objective function's value is worsened. In that case, objective functions can be improved in value at the expense of degrading at least one other objective function's value.[5]

D. Practical Application of MCDM Approach

As listed above, the three objectives in mitigating the risks of obsolete COTS software are minimizing the costs, maximizing the operational efficiency and maximizing the security. Those three objectives are subject to a set of constraints. The first two constraints are resources for budget and time. The third constraint is the Key Performance Indicators (KPI) of the mission outputs derived from the mission mapping.

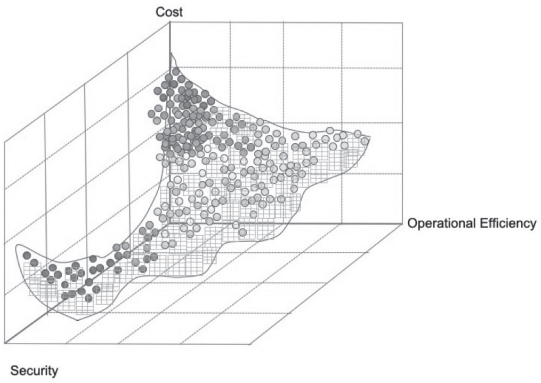
- $min\ \{cost(\bar{x})\}$
- $max\ \{operational\ efficiency\ (\bar{x})\ \}$
- $max\ \{security\ (\bar{x})\}$

subject to

- $budget(\bar{x}) \leq BUDGET$
- $time(\bar{x}) \leq TIME$
- $kpi(\bar{x}) \leq KPI$

BUDGET, TIME and KPI are model parameters and represent the constraints of resource implications and efficiency requirements. Decision variables are mitigation activity on obsolete COTS software and time of implementation. This model will produce a number of Pareto solutions in three-dimensional objective space, as shown in Figure 2.

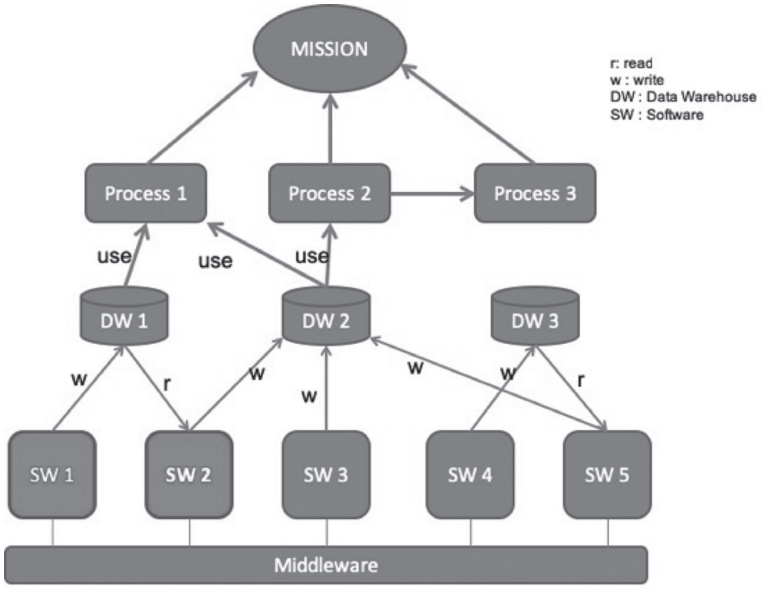
FIGURE 2. PARETO FRONT FOR THREE OBJECTIVES



5 Implementation details and mathematical programming of MCDM problems is beyond the scope of this research. Interested readers are advised to follow Ehrgott and Xavier Gandibleux (2002) and Figueira, Greco, and Ehrgott (2005). A thorough study on the implementation of MCDM by evolutionary algorithms can be found in Özkan and Bulkan (2018).

For the practical implementation of an MCDM problem, consider a system with mission mapping as given in Figure 3. In order to achieve mission objective, threeprocesses are used. Process 1 uses data from the first and second warehouses which all five software applications are either reading or writing. Process 2 uses data fromthe second data warehouse and only second, third and fourth software applications are reading and writing. Process 3 is accepting input directly from Process 2 with no connection to data warehouses. Consider that SW1 and SW2 are called obsolete. In this practical implementation, operational efficiency is measured by weighted product of reliability, availability and maintainability metrics of each SW and data warehouse on mission mapping. The cyber security value is measured by a function of unpatched vulnerabilities. Cost is the parametric value of each mitigation activity adjusted with inflation rate depending on the implementation time.

FIGURE 3. MISSION MAPPING OF A SYSTEM



Each of those solutions within the Pareto set represents an ordered triple of cost, security and operational efficiency objective values (See Figure 4). In this Pareto set, three optimal solutions with varying combinations of mitigation technique and time to implement mitigations are found. Each of those Pareto solutions yields to different objective values in the solution space. It is up to the decision maker to select a solution from the Pareto set based on his or her preferences.

FIGURE 4. OPTIMAL PARETO SET

Decision Variables	Obsolete COTS SW	Mitigation Technique	When	Objective 1 Minimize Cost ($)	Objective 2 Maximize Cybersecurity [0-1]	Objective 3 Maximize Operational Efficiency [0-1]
Solution A	SW1	Migrate to new software	In 3 months	350.000	0.95	0.65
	SW2	Migrate to new software	In 12 months	450.000	0.87	0.72
Solution B	SW1	Reduce users	In 2 months	10.000	0.25	0.15
	SW2	Isolate from system	Immediately	25.000	0.70	0.05
Solution C	SW1	Downgrade SW license	Immediately	45.000	0.10	0.75
	SW2	In house develop	In 6 months	250.000	0.98	0.20

The security community tries to maximize the security and minimize the risks while the user community tries to maximize the operational efficiency and mission outputs, and the service providers try to minimize the costs to increase their profit. This model enables the risk owner to select a Pareto solution for obsolete COTS software that balances the benefits of the three communities of interest.

5. CONCLUSION

COTS software is very appealing to use in complex systems, with numerous advantages such as cost and availability. However, due to market conditions, the product life cycle of such COTS software is very short compared to the longer product life cycles of sustainment-dominated systems, including military ones. Even though there are considerable disadvantages to using COTS software in military systems, the advantages outweigh them and national defense acquisition agencies continue to use COTS products. Increased use of COTS products makes them the capillaries of our NCIs and military systems and is expanding the vulnerability surface for attackers to exploit.

Increased use of COTS software in critical systems with longer product life cycle at minimum leads to sustainability costs, operational efficiency and cyberspace security risks. The sustainability costs are reactive or proactive mitigation costs due to the impacts of obsolescence. The operational efficiency risks are due to reliability,

availability and maintainability risks stemming from obsolescence. The cyber security risks are the unaddressed vulnerabilities of obsolete software.

The security risks of using COTS software have significantly increased within the last two years, as we have seen a considerable increase in the number of reported vulnerabilities in COTS software. Considering the vast amount of existing vulnerabilities in COTS software, the risks associated with obsolete COTS software used in NCI and military systems are highly likely to have a considerable impact if not properly addressed.

In today's cyber conflict, cyber threat vectors have increased their competency and capacity to develop malicious activity for COTS software as well as their intention to use them. This makes the obsolete COTS software a significant element of cyber conflict. Such risks of cyberspace security may easily escalate to a national security issue if not properly addressed.

The cyber security challenges of the obsolete COTS software must be addressed holistically at both government and management levels. At the governance level a comprehensive whole-of-government approach must be pursued. National defense and cyber security strategies and directives are advised to explicitly include hidden risks of COTS obsolescence against NCI and the supply chain. Vendors from foreign countries for systems used in NCI and military systems must be especially closely monitored. A cross-domain central agency between public and private would serve to provide different clusters of organizations with best practices and common approaches. At the management level, each program manager or mission owner must balance the cost, security and operational efficiency objectives within a risk informed trade-off framework. Since those objectives conflict with each other, a MCDM methodology is proposed to find Pareto optimal solutions for all objectives. Decision-makers are compelled to manage the risks within a framework by balancing the needs of the security community, mission owners and the service providers in order to minimize the cost of services, maximize security and maximize operational efficiency.

REFERENCES

Abts, C., Boehm, B., & Clark, E. (2000). COCOTS: A COTS software integration lifecycle cost model-model overview and preliminary data collection findings. *ESCOM-SCOPE Conference*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.8295&rep=rep1&type=pdf

Altexsoft. (2017). What Software Quality (Really) Is and the Metrics You Can Use to Measure It. Retrieved January 5, 2019, from https://www.altexsoft.com/blog/engineering/what-software-quality-really-is-and-the-metrics-you-can-use-to-measure-it/

Bartels, B., Ermel, U., Pecht, M., & Sandborn, P. (2012). *Strategies to the Prediction, Mitigation and Management of Product Obsolescence. Strategies to the Prediction, Mitigation and Management of Product Obsolescence*. https://doi.org/10.1002/9781118275474

Comella-Dorda, S., Dean, J., Lewis, G., Morris, E., Oberndorf, P., & Harper, E. (2004). A process for COTS software product evaluation. *COTS-Based Software Systems*, (July), 86–96. https://doi.org/10.1007/3-540-45588-4_9

CVEDetails. (2019a). Vulnerabilities By Year. Retrieved January 5, 2019, from https://www.cvedetails.com/browse-by-date.php

CVEDetails. (2019b). Windows XP Vulnerability Statistics. Retrieved January 5, 2019, from https://www.cvedetails.com/product/739/Microsoft-Windows-Xp.html?vendor_id=26

Defense Standardization Program Office. (2016). SD-22 – *Diminishing Manufacturing Sources and Material Shortages (DMSMS) A Guidebook of Best Practices for Implementing a Robust DMSMS Management Program Defense Standardization Program Office*.

Ehrgott, M., & Xavier Gandibleux. (2002). *Multiple Criteria Optimization, State of the Art Annotated Bibliographic Surveys*. Kluwer Academic Publisher.

Figueira, J., Greco, S., & Matthias Ehrgott. (2005). *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer.

Flexera. (2017). *Vulnerability review 2017*. Retrieved from https://resources.flexera.com/web/pdf/Research-SVM-Vulnerability-Review-2017.pdf

Flexera. (2018). *Vulnerability Review 2018*. Retrieved from https://resources.flexera.com/web/pdf/Research-SVM-Vulnerability-Review-2018.pdf

Gansler, J. S., & Lucyshyn, W. (2008). *Commercial off the Shelf (COTS)*. https://doi.org/10.1146/annurev.anthro.29.1.107

Gartner. (2017). Focus on the Biggest Security Threats, Not the Most Publicized. Retrieved from https://www.gartner.com/smarterwithgartner/focus-on-the-biggest-security-threats-not-the-most-publicized/

Kelly, S. (2017). Obsolescence Management in Long Term Projects. Retrieved January 5, 2019, from Obsolescence Management in Long Term Projects, 2017, World Codification Forum

Lapham, M. A., & Woody, C. (2006). Sustaining software-intensive systems, (May), 53.

Lee, E. (2003). Software reuse and its impact on Productivity , Quality and Time To Market, 1–6. Retrieved from https://pdfs.semanticscholar.org/08f2/af486985e34d4a46da42b8b4c322c7c22571.pdf

Mckinney, D. (2001). Impact of Commercial Off-The-Shelf (COTS) Software and Technology on Systems Engineering.

Merola, L. (2006). The COTS software obsolescence threat. *Proceedings - Fifth International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, 2006*(Iccbss), 127–133. https://doi.org/10.1109/ICCBSS.2006.29

Microsoft. (2019). Microsoft Product Lifecycle. Retrieved January 5, 2019, from https://support.microsoft.com/en-ie/lifecycle/search?alpha=Microsoft Windows 2000 Server

Miguel, J. P., Mauricio, D., & Rodríguez, G. (2014). A Review of Software Quality Models for the Evaluation of Software Products. *International Journal of Software Engineering & Applications (IJSEA), 5*(6), 31–53. Retrieved from https://arxiv.org/pdf/1412.2977.pdf

Ministry Of Defence. (2005). Defence Industrial Strategy Defence Values for Acquisition. *Defence Studies, 8*(3), 286–310. https://doi.org/10.1080/14702430802252545

Morris, A. T. (2000). COTS score: An acceptance methodology for COTS software. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings, 1*, 4.B.2-1-4.B.2-8. https://doi.org/10.1109/DASC.2000.886948

Munoz, R. G., Shehab, E., Weinitzke, M., Bence, R., Fowler, C., Tothill, S., & Baguley, P. (2015). Key challenges in software application complexity and obsolescence management within aerospace industry. *Procedia CIRP, 37*, 24–29. https://doi.org/10.1016/j.procir.2015.08.013

NCSC. (2017). Obsolete platforms security guidance. Retrieved January 5, 2019, from https://www.ncsc.gov.uk/guidance/obsolete-platforms-security-guidance

Özkan, B., & Bulkan, S. (2016). COTS Parts Obsolescence Management of Sustainment Dominated Military Systems. In *10th NATO Operations Research and Analysis (OR&A)* Conference (pp. 1–18). https://doi.org/10.14339/STO-MP-SAS-OCS-ORA-2016

Özkan, B. E., & Bulkan, S. (2018). Obsolescence Management for Sustainment-Dominated Military Systems: Multiple Criteria Decision-Making Approach Using Evolutionary Algorithms. In *Operation Researches for Military Organizations* (p. 20). https://doi.org/10.4018/978-1-5225-5513-1

PioneerEngineering. (2017). Availability vs. Reliability Part 1. Retrieved January 5, 2019, from https://www.pioneer-engineering.com/resources/availability-vs-reliability-part-1

Pryor, G. A. (2008). Methodology for Estimation of Operational Availability as Applied to Military Systems. *ITEA, 29*(4), 420–428.

Qualys. (2019). Exploits Against Obsolete Software. Retrieved September 20, 2003, from https://www.qualys.com/research/exploits/

ReliabilityWeb. (2018). Understanding the Difference Between Reliability and Availability. Retrieved January 5, 2019, from https://reliabilityweb.com/tips/article/understanding_the_difference_between_reliability_and_availability/

Rojo, F., Roy, R., Shehab, E., & Cheruvu, K. (2010). Key Challenges in Managing Software Obsolescence for Industrial Product-Service Systems (IPS2). *CIRP IPS2 Conference*, 393–398. Retrieved from http://www.ep.liu.se/ecp/077/050/ecp10077050.pdf

S. Rajagopal, J.A. Erkoyuncu, R. R. (2015). Impact of Software Obsolescence in Defence Manufacturing Sectors. *Procedia CIRP, 28*, 197–201. https://doi.org/10.1016/j.procir.2015.04.034

Sandborn, P. (2007). Software obsolescence-Complicating the part and technology obsolescence management problem. *IEEE Trans on Components and Packaging Technologies, 30*(4), 886–888. https://doi.org/10.1109/TCAPT.2007.910918

Sandborn, P. (2008). Trapped on Technology ' s Trailing Edge. *IEEE Spectrum*, (May), 1–6.

Sandborn, P., & Myers, J. (2008). Designing engineering systems for sustainability. *Handbook of Performability Engineering*, 81–104. Retrieved from http://link.springer.com/chapter/10.1007/978-1-84800-131-2_7

Sebok. (2018). Reliability, Availability, and Maintainability. Retrieved January 5, 2019, from https://www.sebokwiki.org/wiki/Reliability,_Availability,_and_Maintainability

Shen, Y., & Willems, S. P. (2014). Modeling sourcing strategies to mitigate part obsolescence. *European Journal of Operational Research, 236*(2), 522–533. https://doi.org/10.1016/j.ejor.2014.01.025

Singh, P., & Sandborn, P. (2006). Obsolescence Driven-Design Refresh Planning For Sustainment-Dominated Systems. *The Engineering Economist, 51*(2), 115–139.

Slovak Republic MoD. (2016). *White Paper on Defence Of the Slovak Republic*.

SpiceWorks. (2017). 10 Computer Systems Still Using Windows XP in 2017. Retrieved January 5, 2019, from https://community.spiceworks.com/topic/2010831-10-computer-systems-still-using-windows-xp-in-2017-three-years-after-eos

Turkey. (2010). 2012-2016 Stratejik Plani, 53, 160. https://doi.org/10.1017/CBO9781107415324.004

U.S. Air Force. (2015). B-52 Stratofortress. Retrieved January 5, 2019, from https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104465/b-52-stratofortress/

US-CERT. (2013). Security Considerations in Managing COTS Software. Retrieved January 5, 2019, from https://www.us-cert.gov/bsi/articles/best-practices/legacy-systems/security-considerations-in-managing-cots-software

U.S. Navy. (2000). Commercial-off-the-Shelf Policy.

Wnuk, K., Gorschek, T., & Zahda, S. (2013). Obsolete software requirements. *Information and Software Technology, 55*(6), 921–940. https://doi.org/10.1016/j.infsof.2012.12.001