Security Concerns in Proprietary and Open Source Software

Fabiola Welzenbach
h1552712

WU
WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Contents

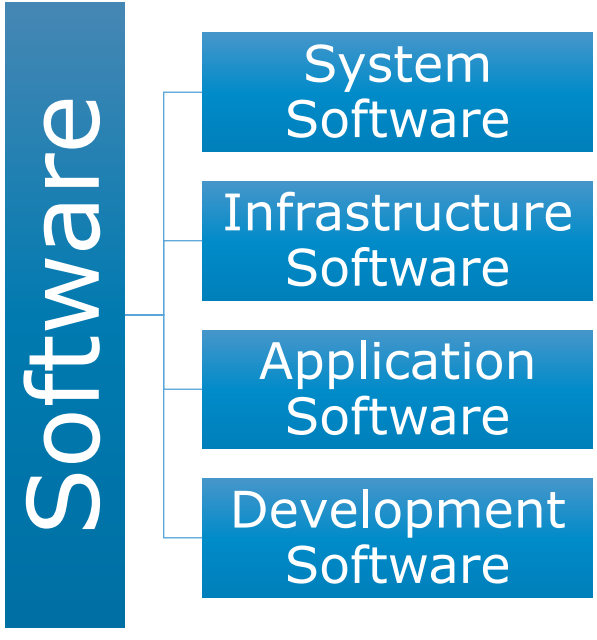
- Introduction and Motivation
- Licensing of Software
- Advantages and Disadvantages of Proprietary and Open Source Software
- Security Concerns of Software
- Software Strategies
- Microsoft and Open Source Software
- Conclusions

Introduction

- The term „digitalization“ is a very popular today
- It offers great advantages like Industry 4.0, new sensors, IoT, etc.
- But security of the systems must be ensured
- Security is an important asset in a company (and for private persons!)
- Attacks are most likely done on software
- Due to the actual crisis this becomes even more important (⟶ home office)
- Software can be proprietary or open source
- Comparison of proprietary and open source software in terms of security

What is Software?

Software is necessary to be able to use the hardware



Controls the existing Hardware
→ operating system

Provides the technical infrastructure for other components → GUIs

Enable the computer to be used for a wide range of applications → Word,…

Used for programming
→ Visual Studio, Netbeans,…

Hansen, H., Mendling, J., & Neumann, G. (2015). Wirtschaftsinformatik. De Gruyter.
Wiedemann, A., Holey, T., & Wiedemann, A. (2007). Wirtschaftsinformatik. Klaus Olfert. Kiehl.

Proprietary and Open Source Software

- Proprietary software:
  - Is software with limitations on using and copying it
  - Limit modifications or copying by
    - Licensing, patent, copyright
    - Release binary-code (only machine readable)
- Open source software:
  - The source code is open to everyone
  - Source code underlies an open source license accepted by the Open Source Initiative (OSI)
  - Goal is to make applications more useful, error-free and more secure

Online Etymology dictionary. (n.d.). Retrieved March 18, 2020, from. https://www.etymonline.com/word/proprietary
Open Source Initiative. (n.d.). Open Source Initiative. Retrieved March 21, 2020, from: https://opensource.org
Sahoo, R., & Sahoo, G. (2016). Computer Science with C++. New Saraswati House (India) Pvt. Ltd.

Licensing of Open Source Software

- Open Source Software:
  - To avoid misuse of open source software, licensing concepts were introduced
  - Open source licensing is a legally valid and binding contract between the developer and the user
  - Over 200 open source licenses exist and each states what users are allowed to do with the software components
  - Two main categories: copyleft and permissive

Honsel, G. (2020). Einmal Utopia und zurück . Technology Review.
Stallman, R. (2002). Selected Essays of Richard M. Stallman, 3rd Edition. Free Software Free Society.
WhiteSource. (2020, February). The Complete Guide for Open Source License. White Source. Retrieved April 6, 2020, from: https://resources.whitesourcesoftware.com/licenses/the-complete-guide-for-open-source-licenses-2020

Licensing of Proprietary Software

- One of the most popular license agreements is the End-User License Agreement (EULA)
- Possible business model:
  - Proprietary software providers sell continually high-margin licenses of the same software
  - Improved versions with new licenses are sold with extra fees
  - Support and maintenance is sold in addition

Landy, G., & Mastrobattista, A. (2008). A Pragmatic Guide to 9 Open Source. In A. J. Gene K. Landy, The IT / Digital Legal Companion: A Comprehensive Business Guide to Software, IT, Internet, Media and IP Law. Syngress; 1 edition.
DiBona, S., & Cooper. (2005). Open Sources 2.0: The Continuing Evolution. O'Reilly Media; 1 edition (31 Oct. 2005)

Advantages and Disadvantages

- Open Source Software:
  - ☺ Cost savings factor as the greatest benefit (no fees for licenses)
  - ☺ High level of security provided by regularly updates
  - ☺ Low error-rates and high stability
  - ☺ Source code is open and “holes” are easier to be found
  - ☺ Being independent from proprietary software providers
  - ☹ Skilled workers are necessary for further development
  - ☹ Unclear warranty situation
  - ☹ The uncertainty of the future of open source software
  - ☹ No or imprecise supplier liability
  - ☹ Fear of security concerns

Bitkom. (2019). Open Source Monitor - Studienbericht 2019 . Retrieved May 15, 2020, from:https://www.bitkom.org/sites/default/files/2020-02/20200218_studienbericht-open-source-monitor-2019_0.pdf

Landy, G., & Mastrobattista, A. (2008). A Pragmatic Guide to 9 Open Source. In A. J. Gene K. Landy, The IT / Digital Legal Companion: A Comprehensive Business Guide to Software, IT, Internet, Media and IP Law. Syngress; 1 edition.

Advantages and Disadvantages

- Proprietary Software:
  - ☻ Usability → not only developers but also application users can use the software
  - ☻ serve directly the end user → smaller scope and fewer functions
  - ☻ Software is checked and maintained regularly
  - ☻ Customized support of proprietary software provider
  - ☹ Dependency on big companies (e.g. Microsoft, Apple, SAP,…)
  - ☹ *Vendor lock-in: s*witching to another product is difficult due to costs and training

Freist, R. (2020). Die Souveränität wiederverlangen. IT & Karriere. Heise Medien.

Security Concerns of Software

- IT-security aims to protect information systems from unauthorised access and unauthorised use
- Guaranteeing accessibility of information, reliability and integrity.
- Most typical software vulnerabilities
  - Cross-site scripting and SQL injections.
  - Security holes at the architecture or conceptual level
  - Security failures at the implementing level
- Testing of implementing security mechanisms is from major importance to prevent security loopholes in the system
- Access control is one example of implementing security mechanisms
- Most applications today lack security and one of the weakest parts is software security

Sametinger, J. (2013). Software Security. Linz : Johannes Kepler University. DOI: 10.1109/ECBS.2013.24
Mouelhi, T., El Kateb, D., & Le Traon, Y. (2015). Inroads in testing access control. from: Advances in Computers, Volume 99, DOI: 10.1016/bs.adcom.2015.04.003

Security Concerns of Open Source Software

- General: Opinion differs drastically whether open source or proprietary software is more secure
- OSS is more secure because many developers have access to the code and fix problems → Peer review
- But source code can also be scanned by hackers for vulnerabilities
- Backdoor: is malicious code, which allows to simply and secretly bypass security mechanisms by an attacker
- After nine years the backdoor in Borland/Inprise's Inter-base database software was found by publishing the source code

Payne, C. (2002). On the security of open source software . Murdoch: School of Information Technology.
Borchers, D. (2020). Sicherheit mit Hintertür. ct Innovative 2020, Heise Medien GmbH & CO. KG.

Security Concerns of Open Source Software

- Employees, who make money with proprietary software are doing their job way much better than the open source community
- Open source developers are mainly interested in the progress of the development and further improvements
- But are not so much interested to invest time and energy on software auditing → security problem!
- Sardonix Project → encourage the open source community to a higher security standard
- Measures the quality by the amount of audited codes and the missed vulnerabilities detected by others
- US government requires for IT products to pass a Federal Information Processing Standard audit

Cohan, C. (2003). Software Security for Open-Source Systems. The IEEE Computer Society. DOI: 10.1109/MSECP.2003.1176994
Lawton, G. (2002). Open Source Security: Opportunity or Oxymoron?. DOI: 10.1109/2.989921
Schneider, F. (2000). Open Source in Security: Visiting the Bizarre. DOI: 10.1109/SECPRI.2000.848477

Security Concerns of Proprietary Software

- Company data can be protected with legal and technical methods
- Legal methods often include intellectual property rights on the program
- Proprietary software vendors are more organized than OSS communities
- Security checks are an essential part of the processes during the software development
- Monopoly problem: Only the company has the information about the source code
- If vulnerabilities will be detected by the proprietary vendor itself the vulnerability can be undisclosed and not available for the public
- Reputation of the company will suffer if vulnerability is detected by others

Clarke, R., & Dorwin , D. (n.d.). Is Open Source Software More Secure?. Retrieved May 19, 2020, from: https://courses.cs.washington.edu/courses/csep590/05au/whitepaper_turnin/oss(10).pdf
Swire, P. (2006). A Theory of Disclosure for Security and Competitive Reasons: Open Source, Proprietary Software, and Government Systems. Houston Law Review, Vol. 42, Issue 5, 2006 January 31, 2006

Security Concerns of Proprietary Software

- Security concern: Most vendors have undisclosed their vulnerabilities
- Laws were written to disclosure vulnerabilities to protect customers
- Proprietary software providers, who disclosure vulnerabilities will gain trust in the long-term relationship with their customers
- Example for undisclosed vulnerability: Volkswagen Fraud 2015
- VW has installed software in their cars that caused in the real-world higher emissions from diesel cars than in the tests they did before
- The software recognized when emission tests will be conducted
- The Environmental Protection Agency (EPA) has found the fraud
- With open source software this fraud would have been detected faster as with proprietary software

Swire, P. (2006). A Theory of Disclosure for Security and Competitive Reasons: Open Source, Proprietary Software, and Government Systems. Houston Law Review, Vol. 42, Issue 5, 2006 January 31, 2006
Thurrott, P. (2015). Volkswagen Used Software to Cheat on Emissions. Retrieved April 28, 2020, from: https://www.petri.com/volkswagen-used-software-to-cheat-on-emissions

Examples of Attacks

- Day Zero Attack:
  - Starts as soon as the vulnerability is first detected
  - Happens even before developers have implemented a patch to close or defense it
  - Are in general not easy to detect
  - When an attack is published, developers start writing a patch to close the vulnerability, but to detect the error it can take years
- Brute Force Attack:
  - Attempts of guessing the required information by the trial-and-error principle
  - Mainly used to hack username and password
  - The better the CPU/GPU of the attacking system the more combinations can be tried out in less time
  - Can easily be repelled → e.g. password can only be typed in for three times or a lockout is activated

FireEye. (n.d.). What is a Zero-Day Exploit?. Retrieved April 16, 2020, from: https://www.fireeye.com/current-threats/what-is-a-zero-day-exploit.html
Kaspersky. (n.d.). What's a Brute Force Attack?. Retrieved May 11, 2020, from: https://www.kaspersky.com/resource-center/definitions/brute-force-attack

Software Strategies

- Many of the vulnerabilities are shared by proprietary and open source software
- The NIST Computer Security Division created the National Vulnerability Database (NVD) in 2000
- The NVD does not testing vulnerabilities by itself
- The database relies on third parties, mostly security specialists and vendors of software
- A software developing company can use this database to close reported vulnerabilities of their software
- But also attackers have insight and can make use of the issue
- Openness is not always the most secure way

National Institute of Standards and Technology. (n.d.).National Vulnerability Database. Retrieved April 25, 2020, from: https://nvd.nist.gov/general

Software Strategies

- Software Composition Analysis (SCA) supports the risk management, security and compliance with licensing requirements
- SCA is the possibility to receive a list of all components included in the applications, the license types and versions of components
- This list is especially from importance for IT specialists
- Helps to get a better understanding of the components used and leads to an increased knowledge about potential security vulnerabilities
- SCA can be a solution to generate higher security standards
- Regularly patches like Patch Tuesday from Microsoft
- Can include security and/or functional patches
- Is a very successful activity → other companies implemented it as well

Bott, E. (n.d.). Insider's guide to managing Microsoft Patch Tuesday. Retrieved March 21, 2020, from: https://www.techrepublic.com/article/insiders-guide-to-managing-microsoft-patch-tuesday/

Flexera. (n.d.). The typical, modern software application is comprised of more than 50 percent open source code. Retrieved March 29, 2020, from: https://www.flexerasoftware.com/blog/what-is-software-composition-analysis/

Microsoft and Open Source Software

- Microsoft is one of the biggest software developing companies
- Changed the strategy from being the main proprietary software producer to becoming a leading edge in open source software
- Even Microsoft Teams is available for Linux since 2019 → Positiv for both sides
- Microsoft published guidelines how customers benefit by using open source software
- Microsoft published guidelines how companies can reduce their risks when using open source software
  - Know the components that are used
  - Check them for vulnerabilities
  - Always update the components
  - Implement a process for risk management

Microsoft. (n.d.). Open Source Security. Retrieved May 14, 2020, from: https://www.microsoft.com/en-us/securityengineering/opensource?activetab=security+analysis%3aprimaryr3
Salazar, M. (2019). Microsoft Teams is now available on Linux. Retrieved May 15, 2020, from: https://techcommunity.microsoft.com/t5/microsoft-teams-blog/microsoft-teams-is-now-available-on-linux/ba-p/1056267#

Conclusions

- Open source software costs less as proprietary software
- But employees with knowledge to implement it are needed
- Proprietary software comes usually with services and updates
- But it has the negative effect of becoming dependend
- Vulnerabilities may be hidden by the proprietary software vendors
- Being dependent from only one company can be very risky as each company has to react as flexible as possible
- Regularly updates are a necessity
- Overall, it can be said that only when a company put much effort on securing their software, both software types can be secure

Thank You!

Fabiola Welzenbach
h1552712