Information Security and Open Source Dual Use Security Software: Trust Paradox

Mario Silic and Andrea Back

Institute of Information Management (IWI), University of St. Gallen, Switzerland

Abstract. Nmap, free open source utility for network exploration or security auditing, today counts for thirteen million lines of code representing four thousand years of programming effort[1]. Hackers can use it to conduct illegal activities, and information security professionals can use it to safeguard their network. In this dual-use context, question of trust is raised. Can we trust programmers developing open source dual use security software? Motivated by this research question, we conducted interviews among hackers and information security professionals, and explored ohloh.net database. Our results show that contributors behind open source security software (OSSS) are hackers, OSSS have important dual-use dimension, information security professionals generally trust OSSS, and large organizations will avoid adopting and using OSSS.

Keywords: information security, dual-use technology, open source security software, FLOSS, trust, hacker, Nmap, ohloh.net.

1 Introduction

"Why would thousands of top-notch software developers contribute for free to the creation of a public good?" was a question asked by Lerner and Tirole (2002).

Ten years later, interest for open source software (OSS) development and community collaboration has been taken to another level. Some popular community collaboration web sites, such as SourceForge[2], have over three hundred thousand available projects[3] with over three million developers participating and contributing to this open source community. Moreover, we saw an uptick in the number of companies, creators of some popular closed source software, being involved in OSS. One example is IBM's contribution to Eclipse with over twelve million lines of code[4].

As we saw the phenomenon behind OSS growing, in parallel, the corresponding research was giving more attention to producing important studies. With the rise of the OSS 'movement' (DiBona et al., 1999; Markus et al., 2000; O'Reilly, 2000), different studies saw OSS teams as virtual organizations (Crowston and Scozzi 2002;

[1] https://www.ohloh.net/p/nmap
[2] http://sourceforge.net
[3] http://sourceforge.net/about
[4] http://dash.eclipse.org/dash/commits/web-app/commit-count-loc.php?sortBy=loc&show=

E. Petrinja et al. (Eds.): OSS 2013, IFIP AICT 404, pp. 194–206, 2013.

Gallivan 2001; Malone and Laubacher 1998; Markus et al. 2000). As such, participants of these organizations are providing their work without financial remuneration or any special interest. Question that many researchers asked is why? Why do developers contribute (Hars and Ou 2002; Lakhani et al. 2003).

Top programmers associated to open source movement are often coming from hacking milieu, and one of the first well-known examples is Linux. Open source software project created by Linus Torvalds, software engineer and hacker, who started the project in 1991 being single contributor. Few years later, Linux project attracted several thousand programmers (Moon and Sproull, 2002) to produce today's one of the most known open source software.

Number of open source projects (e.g. Nmap security scanner) are dealing with information security where their dual use raises trust and security concerns.

On the one side, Black hat hackers can use it to conduct illegal activities against target networks, while on the other side, information security professionals can use it to safeguard their network by discovering security issues and potential threats. In this dual-use context, question of trust is raised. Current research did not explore the link between trust and dual-use context of the OSSS, and we aim to close the existing research gap by exploring the following questions:

Who are the programmers behind open source security software?

Can we trust programmers developing open source dual use security software?

To answer these questions, we will conduct qualitative interviews with fourteen information security professionals who use Nmap OSSS.

We proceed as follows. Firstly, we explore the literature on open source software, dual-use technology and trust theory. Secondly, we present the research methodology outlining data collection and analysis. Thirdly, we discuss the findings of the study. Finally, we conclude by providing some directions for future research, implications for practice, and highlighting limitations of this study.

2 Literature Review

In 1998, Bruce Perens and Eric Raymond, together with other hackers created the "open source" movement (Perens, 1998). The open source movement creations that followed produced a number of open source software with an ever increasing number of contributors. For Linus Torvalds, creator of infamous Linux kernel, the motivation behind it was that it was just "natural within the community" (Torvalds, 1998) or it was "fun to program" (Torvalds and Diamond, 2001). Today's evolution of open source software brought a number of applications in different areas and there is almost no closed software that does not have an equivalent in the open source milieu. One such area relates to information security where different open source security tools appeared. Question of trust is raised as these open source security tools having

dual-use behaviour could have potentially undesired consequences on the security of the information system. In the next sections we will discuss open source and information security literature to finish with trust question.

It is also important to define what "hacker" term means. Unfortunately, there is still no consensus among scholars on the final definition as two opposite definitions are today widely used. One definition speaks of an unorthodox, highly talented professional programmer having deep knowledge and understanding of computers. Another definition refers to one who obtains unauthorized, illegal, access to computers. Recently, we saw clearer separation among different hacker subcultures (White hat, Black hat, Grey hat, Blue hat, Neophyte, Script kiddie, Hacktivist, Elite hacker), but in this study, we keep generic "hacker" definition provided above where we see a hacker as a highly talented programmer that can, eventually, conduct some illegal activities.

2.1 Open Source Dual Use and Information Security

Open source software (OSS) phenomena were widely studied in past years. Research mostly focused on incentives and motivations related to career concerns and peer recognition (Lerner et al. 2002, Hann et al. 2004), learning and enjoyment (Lakhani et al., 2002), and private reputations playing a significant role (Lerner and Tirole, 2000). Virtual organizations and teams where members are working in different countries and using collaboration tools to contribute to the open source appeared as an important research area (Hertel, 2002; Hertel et al., 2002).

Intrinsic and extrinsic motivations were studied in a number of empirical studies showing why OSS developers contribute to OSS development (Hars and Ou 2002; Lakhani and Wolf 2005; Roberts et al. 2006; Wu et al. 2007).

From the extrinsic motivation factors, one is related to the "own-use" value, which refers to "internalized extrinsic motives to create OSS for contributors' personal use" (von Krogh et al., 2012). This own-use value is guiding developers to fix bugs and add features they need. This personal use can lead to possible modifications of the source code which could be difficult to identify in millions of lines of code. An open door for possible malicious programming code inside of the OSSS could be opened. Dual-use question and trust in this coding paradigm can be raised.

Dual-use technology (Evans and Hays, 2006) is a term referring to a technology that has two faces: one that can be used for peaceful aims and one that can be used for military aims. One example is Global Positioning System (GPS), originally used for military use, while today it is widely spread in different end user applications in end user aims. According to Eriksson (1999), any action that will try to prohibit the means of information warfare altogether or restricting their availability - are largely impossible due to the ubiquity and dual-use nature of information technology. Reppy (2012) argues that the dual-use nature of cyber technology along with its status as a quasi-public good defines both the source of the benefits of the technology and the limits to government control.

OSSS represents a good example of the dual-use where a number of unanswered and controversial questions arise. Controversies have two sides: on one side, the question can be asked who are the end users of this potentially dangerous security software? Hackers? Information security professionals? On the other side: can you trust these tools if you are using them for peaceful aims?

OSSS can be used by attackers to perform illegal activities by exploiting vulnerabilities identified on the remote systems (Cavusoglu et al. 2007; Cavusoglu et al. 2008) representing the negative side of the use. The positive side would be used to proactively monitor and safeguard network.

This opens a question of trust. Can information security professionals trust software developed mostly by hackers, which at the end are using these same tools to add features they need. This "add features they need" represents an important own-use value factor where past studies showed that developers develop open source software they find useful by adding features they need (Lakhani and von Hippel (2003), Osterloh and Rota (2007), and Raymond (1999)).

2.2 Trust and OSS

McAllister (1995) defines trust as "the extent to which a person is confident in and willing to act on the basis of, the words, actions, and decisions of another". Later, Jarvenpaa generalized this definition to the team level (Jarvenpaa et al. 1998) and this is very valid for OSS teams as what one OSS team member provides as an output from his participation will depend on the efforts and contributions of other team members. Some previous studies showed strong link between trust and project participation decrease (Jarvenpaa and Leidner 1999) followed by important number of changes in project members structure (von Krogh et al. 2003).

For McAllister (1995), affective and cognitive parts of trust are very important. Affective trust is about emotional attachment between two sides: trustor and trust target. Stewart and Gosain (2006) pointed that affective trust can be "relevant to potential developers' psychological and emotional reasons for joining, staying with and contributing to OSS teams". On the other side, cognitive trust specifies rational assessment of the target by the trustor (McAllister, 1995) where utility and reputation motivations appear as key components which lead programmers' project participation.

Gallivan (2001) explored OSS project participants (core developers, peripheral developers and message posters) identifying different types of trust occurring in the relationships between them. Figure 1. shows these three major groups and trust relationships assumed to be reciprocal. Gallivan (2001) also argues that "the OSS movement appears to rely on explicit forms of control to a much greater degree than on trust" which can be explained either by the fact that trust is probably implicit and unacknowledged in the OSS project lifecycle or it can also be that OSS projects avoid to rely on trust as it could make them vulnerable to members' illegal activities.

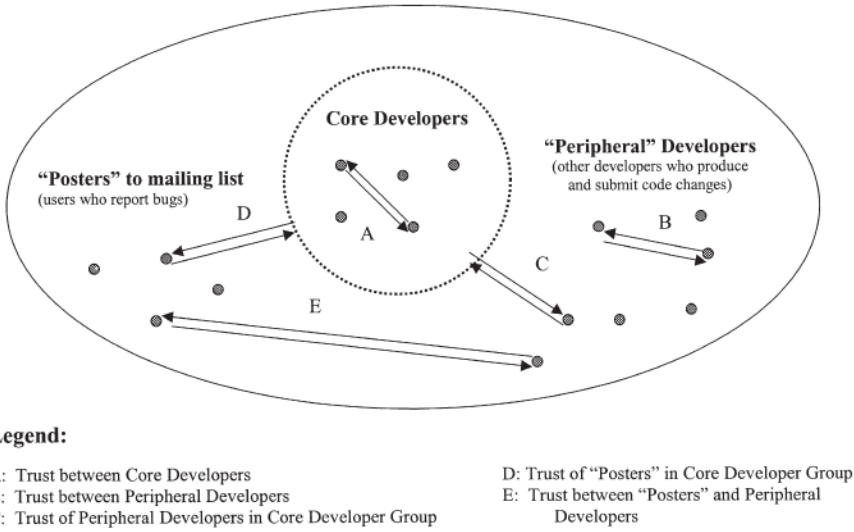

Fig. 1. Structural patterns of trust in open source software projects (Gallivan, 2001)

3 Research Methodology

Our research used ethnographic approach and observations to answer the first research question related to programmers' profiles behind OSSS combined with qualitative interviews and to answer the question related to trust paradox qualitative method was used. In the next sections, we introduce the research setting and explain qualitative and observation approaches used in the paper.

3.1 Research Setting

To investigate our research question we will use open source security tool Nmap, which is a free network security open source scanner. Nmap was created in 1998 by Gordon Lyon who is network security expert, open source programmer, writer, and a hacker. It is used for network scanning and security audits. It is one of the most popular open source security tools. It was also used in popular "The Matrix Reloaded" movie. Reason for Nmap choice is because it is very popular among information security professionals, but it has also wide usage within computer underground community. It also appeared as a good research tool already used in academia (see e.g. Haines et al. 2003). Nmap, being one of the most popular OSSS, is used to set the stage for all interviews and guide all discussions.

To answer the question related to developers behind OSSS, we used observations and qualitative interviews. We analyzed ohloh.net database, large online OSS community with important number of open source projects, of Nmap programmers and randomly identified twenty project contributors. Using internet and Google search tools, we explored profiles of each of the twenty contributors.

3.2 Nmap Security Scanner

Nmap was published in 1997 in famous Phrack (www.phrack.com), an underground magazine with full source code included. Since then, with the help of the computer security community and security enthusiasts, its development continued at an ever increasing pace. Its main purpose is to discover computers and services on the target network. Nmap is an example of the dual-use tool which can be used by hackers in gaining access to unauthorized systems and performing illegal actions. Its typical usage is to discover vulnerabilities on the remote system so they can be exploited by the attacker. On the other side, information security professionals use Nmap to better understand their network, discover potential security issues, and check for eventual vulnerabilities in order to protect their network.

3.3 ohloh.net OSS Community

With its Web 2.0 based application programming interface (API) (Allen et al. 2009) it represents a large OSS community indexing open source software. With over 500,000 OSS, it connects 1.8 million contributors. Ohloh is not a forge (e.g. sourceforge.net). It does not host projects and code, but it connects to project source code repositories and provides analytics insights about project contributors, activity, demographics and code composition.

As explained in the research setting, we use ohloh.net database to identify Nmap contributors in order to explore their profiles and see if programmers belong to hacking milieu.

3.4 Data Collection and Analysis

We gathered data through two different methods. First, we conducted fourteen semi structured interviews between November and December 2012. Second, we used ohloh.net website to complete our first insights on programmers' profiles behind Nmap OSSS. Interviews lasted between 25 to 35 minutes long (on average 28 minutes), totaling 38 pages of transcribed text. Interviewees were chosen from two different categories: information security professional (six interviewees) and programmers (eight interviewees) from different organizations: security companies (36%), large firms (36%), medium size firms (14%), independent experts (14%). Some of the questions we used to conduct interviews were: "In your opinion who are programmers behind open source software?", "How do you see the relationship between trust and open source security tools?", "Did you ever use Nmap to conduct illegal activities?" and "Can programmers inject some malicious code into open source security tools?"

Extracted data from ohloh.net was used to confirm insights from interviews, and after identifying randomly selected twenty contributors from Nmap (https://www.ohloh.net/p/nmap/contributors), we explored their profiles to understand if contributors were hackers. We used observations to identify which contributors described them as being hackers.

We used NVivo software program (version 10) to code the interviews and used exploratory analysis as suggested by Creswell (2002). Data was analyzed and we identified and highlighted different ideas to get some first insights from interviews. Next, we coded different patterns, data, phrases and words and grouped them into defined categories and themes. In this preliminary analysis, two main themes emerged that we further analyzed and discussed in the next sections.

4 Findings

In this section we will explore the findings related to our research questions about programmers' profiles behind OSSS and if we can trust these programmers developing OSSS.

Overall, our findings supported by interviews and data observed from ohloh.net website showed that: 1) Programmers developing open source security tools have important hacking background; 2) Open source security tools have important dual use; 3) Information security professionals generally trust open source security tools and, 4) Large organizations do not trust open source security tools.

To preserve interviewee's anonymity we will proceed as follows: to each interview we will add "INT" with corresponding number between 1 and 14 identifying the interview. For instance, "INT2" will correspond to Interview 2.

4.1 Contributors to Open Source Security Tools Are Hackers

Project members have usually great programming skills and are code addicted. They usually create and code for fun. Interviews and observations of ohloh.net Nmap participants revealed that contributors to open source security tools are hackers. Out of eight programmers we interviewed, seven confirmed that majority of contributors consider themselves as hackers. For one interviewee (INT 3) there is a natural link between being hacker and contributor: "*…well, I think you cannot dissociate hacker subculture and the fact you want to create…hacker by its definition is someone who needs to be seen and heard…in that context, we contribute…*". For another one (INT2) hackers are at the origin of the OSS movement: "*..an example is Linux kernel, it all started as fun and Linux was born…from hacker's coding…this was the origin and now, I see myself as a true hacker, not like some script kiddies…I feel my contribution is visible…*". Other interviewees (e.g. INT 4, INT 7, INT 6) also spoke about hacker's heavy involvement in the open source community projects as for them you cannot be programmer without being hacker. When asked about ethical hacking and more precisely if they ever conducted any black hat activities, almost all interviewees confirmed positively highlighting that it is a normal learning process, especially when you are younger and eager to learn faster. One interviewee (INT 4) said "*I used to be black hat hacker but its definition is always challenging…I never stolen anything…I did not rob anyone…I wanted to learn and learn faster…today, I see things differently…*".

After these first insights from interviewees which clearly pointed to a strong link between contributor and hacker, we analyzed data from ohloh database by exploring randomly selected contributors of Nmap open source security scanner.

By using observations, we explored their profiles and could confirm our initial insights from interviewees, showing that out of 20 contributors representing our sample, 15 declared themselves as hacker.

One example is from Nmap contributor that declared on his website: "I'm a FOSS hacker" or another one stating: "Free Software hacker and Security Enthusiast".

These facts answer our first research question by confirming that programmers behind open source security tools are hackers. However, it is also important to highlight that all interviewees clearly stated that they consider themselves as white hat hackers (hackers that do not conduct illegal activities). This leads to our second research question where knowing that programmers equal hackers we can naturally ask – can we trust them knowing that OSSS have the dual-use side? The answer to this question will be analyzed in the next sections.

4.2 Open Source Security Tools Dual Use

As seen in the previous section, major contributors to OSSS are hackers. Taking into account the fact that dual-use means that hackers can use the tool to conduct illegal activities but also tools can be used for peaceful means by information security professionals, we explored this dual-use paradox by interviewing information security professionals. For one interviewee (INT 9) dual-use can be important threat when using tools like Nmap as you don't have any guarantee behind: "*If I know that my 'enemy' is using the tool to attack my system – the same tool I'm using…there is a problem there…*" Other interviewees (e.g. INT 10 and 12) agreed with this but also added that distinction should be made between very mature projects such as Nmap and some new ones that appeared recently and are still at early development stage: "*…you can't be sure…who knows? Maybe they can modify a piece of code that will impact my network without that I see it immediately…still, I know for mature projects [Nmap] there is very clear code release process…I guess it would be difficult to change something…*" (INT 10).

4.3 Information Security Professionals Generally Trust Open Source Security Tools

Despite dual-use challenge and the fact that open source security tools are developed by hackers, there is a strong direction in trusting open source security tools by information security professionals. Several interviewees highlighted that trust is somehow inevitable and that for well known OSS tools it is part of its fundaments. For one interviewee (INT 14) trust is "*mandatory…without trust any open source project could be challenged*". For another one (INT 13) "*trust is like a marriage – it is that invisible*

link you don't check proactively but you also have mechanisms to do it – at the end, we trust but can also check and understand if anything is potentially harming you".

When questioned about ways and methods to check eventually the source code of the open source security tool that they are using, most information security professionals agreed that it would be very difficult as "*you need to have the knowledge and also, there are millions of lines of code…it would take you an eternity to check it*".

Overall, information security professionals generally trust open source security tools but they also recognized that trust cannot be easily checked due to projects' complexity and time constraints.

4.4 Large Organizations Do Not Trust Open Source Security Tools

Our finding from previous section showed that information security professionals generally trust open source security tools despite the facts that there is no easy method to check the trust relationships. However, several information security professionals confirmed that large organizations, per their policy and procedures, do not trust open source security tools and usually outlaw them by forbidding any use of these tools. One interviewee (INT 12) explained that fear is justified as in large organizations it is not easy to control "*if you have 10,000 employees and you allow all of them to use some open source security software, where source code can be literally modified by any of these employees, you can potentially have important security flaw…so what they do – they simply 'outlaw' it by forbidding access to the software*". Another interviewee (INT 11) challenged this by explaining that a number of governmental organizations implemented open source software and no major issues have been observed: *"take the example of French government where they implemented Linux at large scale and open source kicked out closed source software…there was no big impact on information system security*". Most of other interviewees had an agreement on the fact that large organizations tend not to trust open source security tools but they also highlighted that more we understand underlying mechanisms behind OSS projects and the way they are controlled, more trust will be there. One interviewee spoke of companies such as RedHat that started as open source companies but are now offering OSS software combined with some trust relationships (INT 9): "*RedHat is the perfect example…today they sell 'trust' combined with free open source system – Linux…large organizations have much higher trust when they know there is a commercial organization that brings trust and guarantee…*"

5 Discussion

In this section we will discuss our findings presented in the previous section. From different interviewees held with information security professionals and hackers, together with data from ohloh database, we found that main contributors to OSSS are hackers. Also, despite dual use side, information security professionals generally have trust in OSSS. However, large organizations avoid OSSS use in their network.

Previous studies that explored questions like 'Who are the developers' provided demographics statistics on programmers of OSS (e.g. Lakhani et al 2002, Hertel et al. 2033) without revealing programmers' profiles such as their hacking background. With our study we provide the answer to this question from a more holistic perspective where we confirm hackers are top contributors to OSSS.

Also, our study confirms previous study (Gallivan, 2001) that speaks of implicit and unacknowledged trust that does not necessarily appear in OSS project descriptions. Harrison and St John (1996) argued that too excessive controls could lead to 'conflict and distrust,' and this is what we saw also in our study, where for contributors, too much of control would be harmful as there seems to be a general agreement on trust level between different process stakeholders. Gallivan (2001) also argued that there is no need for high level trust for collaborative software development and our study further extends this direction by adding relationships between organizational size component and trust. In this relationship, large organizations will refuse OSSS use and adoption unless there is a trusted provider of open source technology (e.g. Redhat) where source code remains free and open but trust component is guaranteed by open source leader.

While some past research on trust confirmed that 'trust building' and 'control mechanisms' are two factors that should ensure confidence in another party's behaviour (Das and Teng, 1998), our research shows that another factor we call 'leader link' should be part of the triangle. This 'leader link' would have compliance and police role by guaranteeing trust to its partners.

6 Conclusion

This study tried to answer two important research questions: who are the programmers behind OSSS; and can we trust programmers developing open source dual use security software? Our study revealed some important findings: contributors behind OSSS are hackers, open source security tools have important dual-use dimension, information security professionals generally trust open source security tools and large organizations will avoid adopting an OSSS.

With the present research we close existing research gap as dual-use technology related to OSSS usage and adoption within organizations, in the context where these same tools developed by hackers did not get significant research focus in literature.

Our findings contain some practical implications for organizations to take into account, especially for large organizations, where we found a need to have 'leader link' involved in 'control' and 'trust' triangle. This leader would bring more trust to open source software and would reduce risk level.

Limitations of this study relate to the fact that we focused only on one (Nmap) OSSS which could have some influence on some insights from interviewees. Moreover, to understand programmers' profiles we relied on what programmers were saying

about their background and if they considered themselves as hackers or not. More quantitative approach would be more accurate in this context.

For future research directions, it would be interesting to approach the trust topic more from black hat perspective trying to understand underlying motivations they could have when coding. Also, more in-depth study is welcomed to understand contributors' profiles, their background and trust relationships.

References

1. Allen, J., Collison, S., and Luckey, R.: Ohloh Web Site Api (2009), http://www.ohloh.net
2. Boehm, B.W.: Software Engineering Economics. Prentice Hall (1981)
3. Cavusoglu, H., Cavusoglu, H., Raghunathan, S.: Efficiency of Vulnerability Disclosure Mechanisms to Disseminate Vulnerability Knowledge. IEEE Transactions on Software Engineering 33(3), 171–185 (2007)
4. Cavusoglu, H., Cavusoglu, H., Zhang, J.: Security Patch Management: Share the Burden or Share the Damage? Management Science 54(4), 657–670 (2008)
5. Creswell, J.W.: Educational research: Planning, conducting and evaluating quantitative and qualitative Research. Pearson Education, Inc., Upper Saddle River (2002)
6. Crowston, K., Scozzi, B.: Open Source Software Projects as Virtual Organizations: Competency Rallying for Software Development. lEE Proceedings Software 149(1), 3–17 (2002)
7. Das, T.K., Teng, B.: Between trust and control: developing confidence in partner cooperation in alliances. Academy of Management Review 23, 491–512 (1998)
8. DiBona, C., Ockman, S., Stone, M.: Open Sources. Voices from the Open Source Revolution. O'Reilly & Associates, Sebastapol (1999)
9. Eriksson Anders, E.: Information Warfare: Hype or Reality. The Nonproliferation Review (Spring-Summer 1999)
10. Williams, E.M., Hays Bret, B.: Dual-Use Technology In the Context of the Non-Proliferation Regime, History and Technology (March 2006), doi: 10.1080/07341510500517850
11. Gallivan, M.: Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies. Inf. Syst. J. 11(4), 277–304 (2001)
12. Gallivan, M.J.: Striking a Balance Between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies. Information Systems Joumal 11(4), 277–304 (2001)
13. Haines, J., Ryder, D.K., Tinnel, L., Taylor, S.: Validation of Sensor Alert Correlators. IEEE Security and Privacy 1(1), 46–56 (2003), http://dx.doi.org/10.1109/MSECP.2003.1176995, doi:10.1109/MSECP.2003.1176995
14. Harrison, J.S., St John, C.H.: Managing and partnering with external stakeholders. Academy of Management Executive 10, 46–61 (1996)
15. Hars, A., Ou, S.: Working for Free? Motivations for Participating in Open-Source Projects. International Journal of Electronic Commerce (6), 25–39 (2002)
16. Hars, A., Ou, S.: Working for Free? Motivations for Participating in Open Source Projeets. International Journal of Electronic Commerce 6(3), 25–39 (2002)

17. Hertel, G.: Management virtueller teams auf der basis sozialpsychologischer modelle. In: Witte, E.H. (ed.) Sozialpsychologie Wirtschaflicher Prozesse, pp. 172–202. Pabst Publishers, Lengerich (2002)
18. Hertel, G., Konradt, U., Orlikowski, B.: Managing distance by interdependence: goal setting, task interdependence, and team-based rewards in virtual teams, submitted for publication. Jargon File (2002), The On-Line Hacker Jargon File, Version
19. Hertel, G., Niedner, S., Herrmann, S.: Motivation of software developers in Open Source projects: An internet-based survey of contributors to the Linux kernel. Research Policy 32(7), 1159–1177 (2003)
20. Lakhani, K.R., von Hippel, E.: How Open Source Software Works: 'Free' User-to-User Assistance. Research Policy 32(6), 923–943 (2003)
21. Lakhani, K., Wolf, B., Bates, J., DiBona, C.: Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. The Boston Consulting Group Hacker Survey (2002), http://www.osdn.com/bcg
22. Lakhani, K.R., Wolf, R.G.: Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects (September 2003). MIT Sloan Working Paper No. 4425-3. Available at SSRN http://ssrn.com/abstract=443040 or http://dx.doi.org/10.2139/ssrn.443040
23. Lerner, J., Tirole, J.: Some Simple Economics of Open Source. Journal of Industrial Economics 50(2), 197–234 (2002)
24. Lerner, J., Tirole, J.: The Simple Economics of Open Source, NBER Working Paper Series, WP 7600. Harvard University, Cambridge, MA (2000)
25. Malone, T.W., Laubacher, R.J.: The Dawn of the E-Lance Economy. Harvard Business Review 76(5), 144–152 (1998)
26. Markus, M.L., Manville, B., Agres, C.E.: What Makes a Virtual Organization Work? Shan Management Review 42(1), 13–26 (2000)
27. Markus, M.L., Manville, B., Agres, C.E.: What makes a virtual organization work? Sloan Management Review 42, 13–26 (2000)
28. Moon, J.Y., Sproull, L.: Essence of distributed Trust and control in a virtual organization 303 © 2001 Blackwell Science Ltd. Information Systems Journal 11, 277–304 (2000), work: the case of the Linux kernel. First Monday: Peer-Reviewed Journal on the Internet, http://www.firstmonday.org/issues/issue5_11/moon/index.html
29. Moon, J.Y., Sproull, L.: Essence of distributed work: the case of the Linux kernel. In: Hinds, P., Kiesler, S. (eds.) Distributed Work, pp. 381–404. MIT Press, Cambridge (2002), Also available on the World Wide Web: http://www.firstmonday.dk/issues/issue511/moon/index.html (retrieved October 28, 2002)
30. O'Reilly, T.: Open source: the model for collaboration in the age of the Internet. Computers, Freedom and Privacy (keynote address), Toronto,Canada. O'Reilly Network (2000), http://www.wideopen.com/reprint/740.html
31. Osterloh, M., Rota, S.G.: Open Source Software Development—Just Another Case of Collective Invention? Research Policy 36(2), 157–171 (2007)
32. Perens, B.: The Open Source Definition (1998), http://perens.com/articles/osd.html
33. Raymond, E.S.: The Cathedral & the Bazaar, pp. 19–64. O'Reilly & Associates, Inc., Sebastapol (1999)
34. Reppy, J.: International School on Disarmament and Research on Conflicts, http://www.isodarco.it/courses/andalo12/paper/ISO12_ReppyCyber.pdf

35. Roberts, J.A., Hann, I., Slaughter, S.A.: Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. Management Science 52(7), 984–999 (2006)
36. Stewart, K.J., Gosain, S.: The Impact of Ideology on Effectiveness in Open Source Software Development Teams. MIS Quarterly 30(2) (2006)
37. Torvalds, L.: Interview with Linus Torvalds: what motivates free software developers? First Monday 3 (1998), http://www.firstmonday.dk/issues/33/torvalds (retrieved from the World Wide Web, December 14, 2001)
38. Torvalds, L., Diamond, D.: Just for Fun: the Story of an Accidental Revolutionary. Harper Business, New York (2001)
39. von Krogh, G., Haefliger, S., Spaeth, S., Wallin, M.W.: Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development. MIS Quarterly 36(2), 649–676 (2012)
40. Wu, C., Gerlach, J.H., Young, C.E.: An Empirical Analysis of Open Source Software Developers' Motivations and Continuance Intentions. Information & Management 44(3), 253–262 (2007)