DOI:10.33179/BSV.99.SVI.11.CMC.13.3.4

Maja Bolle

Strokovni članek

INFORMACIJSKA VARNOST IN ODPRTOKODNA PROGRAMSKA OPREMA

INFORMATION SECURITY AND OPEN SOURCE SOFTWARE

Professional article

Povzetek Informacijska varnost je ob vedno večji odvisnosti od računalniških sistemov pomembna tema tudi za javno upravo. Ob vedno večjem številu napadov in drugih posegov v integriteto operacijskih sistemov in drugega programja so se mnoge države odločile za prehod na odprtokodno programsko opremo, ki poleg varčevanja pri nakupu programskih licenc državam omogoča tudi večji nadzor nad to opremo. Nekatere raziskave govorijo v prid varnosti zaprtokodnih sistemov, spet druge priporočajo uporabo odprtokodnih. Podatki kažejo, da ima odprtokodna programska oprema na marsikaterem področju boljše varnostne mehanizme kot zaprtokodna, vendar pa ima tudi ta svoje omejitve. Zaradi teh mora biti prehod držav na odprtokodno programsko opremo dobro premišljen.

Ključne besede ***Odprta koda, odprtokodna programska oprema, zaprtokodna programska oprema, informacijska varnost.***

Abstract In times of increasing dependence on computer systems, information security emerges as an important issue for public administrations. Many governments have made a transition to open source software; the reason being not just financial, but connected to increasing numbers of operating systems security issues. Having more control over the systems is also key. Some researchers speak in favour of proprietary software, others in favour of open source software. Data shows, however, that open source software leads over proprietary software in security mechanisms, although it is not without its limitations. This is the reason that states transitioning to open source software must take precautions in doing so.

Key words ***Open source, open source software, proprietary software, information security.***

Introduction

We live at a time in which life and work is no longer possible without a computer. It is therefore essential to address the question of information security. Due to increasing interventions into the integrity of the most utilized operating systems and other software, numerous countries have decided to transit to open source software. Some countries, such as Slovenia, are still searching their way through the transition. Although most countries have transferred to open source software for economic reasons, security reasons should not be disregarded (Kimberly, 2005).

Open source software is a term used for describing software in which the source code is freely accessible and which can be used free-of charge. The operation of such software can be investigated and its original, supplemented and modified copies can be changed. The above is not possible to do with proprietary software. The terms of its use are written in various licenses that include guidance on the use of the Open Source Initiative. The most important criteria are free distribution, access to the source code and permission to modify and integrate the source (http://www.opensource.org/).

1 OPEN SOURCE HISTORY AND PHILOSOPHY

The beginnings of open source software date back to the 1950s. The first computers were large and, considering contemporary standards, not very capable. They also had very poor software. User interfaces were unfriendly and software capabilities were lower than what the hardware enabled. At that time, programmers were hard to find. Consequently users started to unite with the intent to exchange ideas and software that would make better use of the hardware capabilities and satisfy their needs. Thus the first associations uniting such enthusiasts were formed – the most well known association is the SHARE association, founded in 1955. However, at the end of the 1960s the system changed. The companies that were selling hardware also enclosed software which, at the time, was free of charge (http://tinyurl.com/6jgg3ut). Hence, the market offering software for payment was getting stronger every day.

1.1 Origin of the first open source software

1.1.1 Unix

In 1969, Ken Thompson produced an operating system UNICS (Uniplexed information and computing services) that was later renamed Unix. The AT&T company, at which Thompson was employed, had seen a business opportunity in Unix and requested its licensing. The first licence was free of charge and provided considerable freedom to users. However, the AT&T company did not offer support for this licence. All software is inevitably confronted with the so-called bug phenomena, which are faults in the programme code resulting in undesired and unexpected outcomes on use. All this had a direct influence on Unix users who had started to unify into groups, collectively eliminate bugs and to improve Unix – something that the AT&T company did not foresee with its license. However, the source was closed in such a way that the groups had to pay AT&T to access the source code.

This was also done by the BTL research group that copied the source code in the C Programming Language. By making modifications to the code it made it possible for Unix to operate on any hardware and on any computer. It was also made possible for the users to produce their own drivers for equipment they needed at work (Weber, 2004).

1.1.2 BSD

The University of California, Berkeley, at which Unix was used, played an important role in the development and modification of open source software. Researchers, Bill Joy and Chuck Haley, developed and supplemented Unix's core. In 1978, Joy produced a number of add-ins for Unix that, together with the core, formed a package named Berkeley Software Distribution (BSD). BSD was not an independent operating system, but a Unix distribution. BSD became extremely popular among students and researchers and Unix was quickly abandoned in favour of it (Weber, 2004).

In 1968, the predecessor of today's internet, ARPANET, started to operate. At first it provided connection between the American Department of Defence's agency DARPA (Defense Advanced Research Projects Agency) and other research institutions.

DARPA's aim was to communicate with other institutions via ARPNET. This, however, was made difficult due to the incompatibility of different computers and operating systems. DARPA therefore asked the BSD researchers to develop software that would work on all hardware equipment; version 4.2.

The BSD programme from 1983 included a new protocol called TCP/IP used for internet communication that is still used and serves as the base of today's internet. The internet and the TCP/IP protocol are the main reasons why BSD became widely distributed over the internet.

In the meantime, AT&T tightened the Unix license conditions thus increasing its price. In 1989, the price of the license was 250,000 USD. Since the universities could no longer afford to buy Unix, they started using BSD (Weber, 2004).

1.1.3 Free software foundation

Richard Stallman is a founder of the Free Software Foundation. The aim of this non-profit making institution was to create a free of charge operating system - the source code of which would be available to everyone and could be freely changed. The operating system was named GNU. In 1984, he wrote the GNU Manifest in which he explained the meaning of the term free software.

The term does not necessarily refer to a free of charge software, but only to free software: In this sense free refers to the accessibility of the source code and the possibility to modify it (Wynants, 2005). This means that Stallman is not opposed to software

having a price with which the programmer's work is paid, but wants to keep it free in its essence (Spanish libre). The manifesto contains four principles that still apply:

1) freedom to use the programme for any kind of purpose (freedom no. 1);
2) freedom to study the programme and modify it as desired. Requirement to access the source code (freedom no. 1);
3) free distribution of copies (freedom no. 2);
4) freedom to improve the programme and publish improvements (and adapted versions) to the benefit of the community (freedom no. 3); precondition for which is access to the source code.

Since Stallman was aware that these basic freedoms could be taken advantage of, he has additionally improved the license. Such a license is the opposite of copyright and is called a General Public Licence (GPL). The software licensed under GPL can never become proprietary. This applies also to the modified programme equipment that derives from free software. Only a combination of proprietary and free software can be issued and even that only under the condition that everything is licensed under GPL (Weber, 2004). The GPL license was supplemented throughout time (the last version is GPL v3) and used to serve as a basis for more specific licenses (http://tinyurl.com/hdpo9).

1.1.4 Linux

Linux is the world's most widespread open source operating system in the field of supercomputers and servers. It owns 91 percent of the market share among supercomputers, followed by Unix with three percent and Windows with one percent (http://tinyurl.com/27l5wvh). Linux also owns the largest market share among servers, which is as much as 70.71 percent (http://tinyurl.com/3hdqvgd). However, Linux still has the lowest share in desktop computers (5.1 percent), while Windows owns 85 percent and Mac OS X 8.3 percent of the market share (*http://tinyurl.com/28lpgq*).

Linux was created in 1991 under the management of the then 21 year old Linus Torvalds. On 25 August 1991, in a discussion group called comp.os.minix, he declared his intention to develop a kernel for a new operating system. On 17 September, he published his first version of the Linux operating system kernel on the internet. He invited people to test his system and improve it. Since Linux was able to obtain more and more supporters and developers every day, the first version 1.0.0 was published in 1994 (Weber, 2004). Nowadays, Linux can operate on almost any computer structure (desktop computers, supercomputers, servers, wrist watches, Playstation 3 game console) (http://www.Linuxfordevices.com/).

Since Linux is nothing more than a core of the operational system, developers from all across the world developed graphical interfaces, desktops, programmes, drivers etc. for it and combined everything into a package called distribution. The distributions are based on an individual kernel of the operating system (e.g. Linux, BSD) and differ from one another according to what type of operating system, desktop and

repositories they include. The most widespread and popular distribution of Linux is Ubuntu that over 12 million people use on their desktop computers (Jose, 2011).

2 SECURITY OF OPEN SOURCE SOFTWARE

Which system is more secure: the one with less kernels or a system which can quickly make corrections or perhaps a third one, the vulnerability of which can affect less people. There are a lot of researchers that are in favour of one or the other; most of them focus only on one of the mentioned aspects (Laurie, 2006). This paper is striving to present the issue in the broadest context possible and in different areas that are connected to information security in one way or another.

2.1 Large number of distributions

According to the Distro Watch data (http://distrowatch.com/), there are currently 320 distributions in the world all based on operating systems similar to Linux and Unix. The actual number of distributions is in fact much bigger, as anyone can make their own distribution at home. Due to a large number of distributions the possibility of malware [1] software being written for a specific distribution is much smaller than with proprietary software. In reference to two of the most widespread proprietary operating systems we are not even familiar with the term distribution, for every individual purchases an operating system that no longer has the form of distributions, but only versions (Apple and its Mac OS X with versions Snow Leopard, Lion and Microsoft's Windows with versions Windows XP, Vista, 7...). Each version of these proprietary operating systems is based on a different kernel, which reduces the transferability of malware among them. However, the malware written for proprietary systems has a much larger distribution value due to a small number of versions and the mono-culturality of Microsoft's operating systems. This is because these versions are modified less frequently than those for the open source operating systems. In addition, the probability of malware infecting a large number of distributions is less likely to occur due to differences in code. From the aspect of information security, a large number of distributions, which is characteristic of open source software, is therefore considered to be an advantage.

2.2 Malware

Malware is written for a specific operating system and its version, as the source code differs, is different for each version. To this day we have witnessed over two million cases of malware for the Windows operating system, 1989 cases of malware for the Linux operating system and 48 cases for the Apple Mac OS X (Kalkuhl, 2009). The number of malware cases has considerably increased during the last two years – both in open source and proprietary operating systems. The recent reports by the Kaspersky institute state that the reason for the increase in the number of malware cases lies in the fact that both operating systems are becoming more popular (http://

[1] *Malware software is software that wishes to be infiltrated into a computer system and damage it without the user consenting to such an action (Meintjes, 2011).*

tinyurl.com/44x9pxd). The report also states that Linux is the most affected operating system that resembles Unix. It is also the most widespread. Nonetheless, the statistics demonstrate that Linux was attacked mainly through servers and less through the desktop area (Sapronov, 2007; Germain, 2008).

Despite the fact that up to this day we have witnessed as many as 1989 cases of malware for open source software its lifespan is very short and it does not cause as much damage to Linux as to the Windows operating system. The reason for this lies in the administrator's access (superuser account – more widely known as ROOT) which is, for safety reasons, automatically deactivated in Linux, BSD and other Unix-like operating systems to prevent users that are unskilled in using the operating system from damaging it (http://tinyurl.com/o4foa). In practice this means that we set a password that enables us to modify all settings in the operating system. It is different with the Windows operating systems that had not known an actual blockade of administrator access with a password up until the versions Vista and 7. Nonetheless, this blockade is still not very severe as the users can modify a number of things without having administrator access. (Schneier, 2006). The Apple's operating system Mac OS X is based on Unix's kernel in which the administrator's access is disabled by default and requires the user to type in the password, which is a positive trait.

(http://support.apple.com/kb/ht1528). If an infection with malware occurs in Linux the damage will not be significant, since this equipment will not have administrator access for the entire system. Its effect will therefore be either local or there will be no effect at all (http://librenix.com/?inode=21; Koetzle, 2004). Similar findings were stated in the research Analysis of the Impact of Open Source Software from 2001 (http://tinyurl.com/6lyeod8) in which the impact of viruses on various operating systems was studied: At the time, Windows had over 60,000 viruses, while Mac OS X and Linux both had 40 respectively. Despite the fact that most viruses written for Windows did not cause great damage, some hundreds of viruses were much more harmful. Two thirds of all known viruses have caused considerable damage to the Apple's Mac OS X system, while not even one of the Linux viruses caused great damage or spread more widely across the system (Peeling, 2001). The safety of operating systems similar to Unix can be considerably threatened by the so-called rootkit that enables covert access to a computer system and the use of administrator privileges (Chuvakin, 2003).

As we can see, the amount of existing malware for a specific operating system is not that important. What is more important is the impact malware can have on a system in terms of level and scale.

2.3 Do many eyes really see more?

The many eyes system is a system of inspection, with which every user can have an overview of the source code of the open code software – in theory this minimizes

the possibility for open code software to contain a malicious code such as backdoor through which an unauthorized person could obtain access to the system.

The defenders of open source software often argue that the many eyes system enables a rapid detection of bugs in the code. However, this is not always the case in practice. Nowadays most users are not skilled in programming, cannot read the source code or recognize deficiencies or possible backdoors in them. Open source software is used for everyday chores, such as writing texts, management of tables and writing e-mails. Nonetheless it still provides insight for those who are interested in it and capable of it. This is an important difference, for proprietary systems do not enable such insight (Laurie, 2006).

There have been a number of cases in history when vulnerabilities were not discovered for several years even though the open source software was examined by a number of people. One of the most interesting cases is Ken Thompson's backdoor. He was a developer of the Unix system into which he incorporated a backdoor. Thompson revealed this only after fourteen years. With this experiment Thompson wanted to demonstrate that we should not rely on other people too much. He is convinced that only the code we write ourselves is safe (O'Dowd, 2004). At this point we should address the question of the human factor. Even though the code was examined by a number of people it does not mean that the examination was detailed enough or that the examiners were competent enough to detect all vulnerabilities.

2.4 Time for correction

Time for correction is a time between the moment in which the vulnerability in the code has been detected and the time at which the correction has been made. This time has to be as short as possible – the longer the vulnerability is left without a correction, the more endangered is the system's security. Research comparing the security of Windows operating systems and different Linux distributions (Debian, Red Hat, Mandark) conducted over one year has demonstrated a number of threats, the time necessary for the production of the correction and a number of corrected threats (Koetzle, 2004). On average Windows spent the least amount of time, that is 25 days on the production of corrections. It is followed by Linux's distributions Red Hat and Debian with 57 days and Mandark with 82 days.

However, this data alone does not suffice to make a comparison of the operating systems: It was discovered that the Windows operating system has the highest level of threats (67 percent of all threats), followed by Red hat with 56 percent of the same level threats. The research has also included the measuring of time that the providers require for insertion of corrections into the distribution. For the Windows operating system this time was the same as for the production of corrections, since the distributions of the Windows systems do not exist. On average the Debian would require only 32 days, which is a lot less than the time required for the production of corrections (57 days). The same applied for Red Hat with 47 days. Debian was so fast

because this was the only examined distribution that was being used without a new installation of the entire system being required (rolling release).

The time of production of these corrections is not the only important security element. The most important are the users, because they have to install the corrections. Microsoft users have been threatened by nine vulnerabilities of the highest level. Nonetheless, most examined users have not installed the corrections for over 305 days. This means that on average, they have been at threat for 305 days despite the fact that Microsoft produced the corrections after approximately 25 days (Koetzle, 2004).

If people discover vulnerabilities in the open source software through the system many eyes they immediately publish the news on internet pages, forums etc. (such as Ubuntu Bugs Launchpad – https://bugs.launchpad.net/ubuntu/). The developers of open source software eliminate weaknesses in the shortest time possible. However, there exist differences among the developers of various open source software. On average the Apache issues corrections on a daily basis, which means that a certain vulnerability seldomly exists more than a day. Ubuntu, the most widespread Linux distribution, issues corrections in reference to priority order determined through the Ubuntu Bug Launchpad.

The providers do not refresh their bases in accordance with the everyday open source software. This deficiency is eliminated by the repository that enables the users to install the latest version of the programme independent of the providers of the distribution and in the moment the developer publishes it. The repository was made for purposes of faster distribution of the newest version of software equipment to users and in order for the developers to obtain faster feedback about the quality of the equipment, which speeds up its development (Laurie, 2006). In 2007, the Ubuntu issued the **Personal Package Archive** (PPA) software with the purpose to additionally expedite and facilitate the distribution of software through repositories (Humbrey, 2011). However, not all repositories are safe and it is therefore recommendable to use only those repositories that are verified and in no way "suspicious".

The proprietary operating systems Windows and Mac OS do not have the many eyes system, therefore security is provided by the developers of each system individually. They do not publish all vulnerabilities, for which reason we are unaware of how long we are exposed to them or their actual number.

As we can see, a large number of published threats does not mean that the system is more vulnerable, but more transparent. Since we cannot see all vulnerabilities in proprietary software because of its intransparency, the open source software is an advantage.

2.5 Security through transparency or concealment

We have heard many times that hiding of the source code provides better security, however in practice this is not the case. One of the first cryptologists, Auguste Kerckhoffs, wrote six principles of good cartography in 1883. These principles are nowadays known as the Kerckhoffs' Pinciple, which states that a good coding system is safe even if we know everything except the encryption key about it. Kerckhoffs also rejects the principle that it is possible to provide security only by means of concealment. He does not demand that the encoding system is public, but stresses that secrecy does not ensure greater security; on the contrary, it can even threaten it (Kovačič, 2006). A covert system can threaten the security by containing faults that could be, if such a system was public, detected and repaired. Bruce Schneier, an expert on information security and a cryptologist says: "I cannot remember any cryptographic system developed secretly, in which, after being introduced to the public, the cryptographic community would not find a mistake." (Schneier, 2002).

Something similar occurred in the famous database case of the Borland InterBase, in which a backdoor was discovered in 2000. That was the year in which the company published the software source code that prior to this occasion had been proprietary or closed.

The programmers discovered that in 1994 a backdoor had been intentionally added to the database. The door enabled an individual to have full access to all information and even to insert information and contents with the user name "politically" and the password "correct". It is even more alarming that the database was used by the Boston stock market and large corporations such as Motorola, Nokia and Boeing. The programmers of open source software made rapid corrections that closed the backdoor (Poulsen, 2001).

In the spirit of open source today, even Microsoft provides countries with access to the source code; however, it does so only under conditions laid down in the Government Security Programme contract. The contract was signed by approximately 60 countries, including NATO states, China and the Russian secret service FSB (Espiner, 2010). Nonetheless, Microsoft is the one that decides if the source code will be revealed to a specific country or not. The countries to which Microsoft does not enable access to the source code are: Venezuela, Cuba and other countries whose public administrations transferred to the open code software. Richard Clayton from Cambridge University draws attention to the weaknesses of such a system: the countries can detect vulnerabilities in security much more easily and thus use them to attack other states, for they do not publish the information about the mentioned vulnerabilities. The latter are known only inside the system which has access to the source code. Another limitation of the Government Security Program is that it provides countries with an insight into the source code, but does not enable its modification (http://tinyurl.com/3fldnhr).

3 TRANSFER OF NATIONAL PUBLIC ADMINISTRATIONS TO OPEN SOURCE SOFTWARE

Recently, more states decided to transfer to open source software. Some of them are making only a partial transfer (e.g. in certain government agencies) and use only open source software (these are the USA, France, Germany, the Czech Republic, Macedonia, South Africa, the Philippines). Other countries have decided to make a full transfer to open source software (China, Russia, Brazil, Venezuela, Pakistan, Cuba, Turkey, Malaysia and Spain) meaning that they use the distributions of the Linux or BSD operating systems together with the relevant software. Most countries that have decided to do a full transfer have created their own national distributions of the operating systems that contain specific software (the one that is used in a specific public administration). In order to ensure better security these countries made their own repositories, which are updated by their public institutions. The software found in these repositories was developed specifically for the needs of state institutions. This contributes to greater security of the operating systems, since the software located in repositories is examined and developed by the states themselves. In addition to a reduction in costs, security is one of the main reasons why national public administrations decided to transfer to the open source software (Lewis, 2006). In 1999, when the first reports that the American National Security Agency (NSA) had entered a backdoor into every copy of the Windows 95 operating system (Campbell, 1999), states have started to question the security of and the control over the operating systems of the Microsoft company. They were concerned also because of Microsoft's mono-cultural tendencies, for the company owns over 80 percent of the desktop computers market share. In addition, Microsoft had started to supplement the code in such a way that it limited its operation on other systems thus "chaining" the states to it (vendor-lock in). This has encouraged countries to think about alternative programme solutions that would provide better control over computer systems, greater transparency, greater independence from Microsoft and a possibility to develop and adjust the system to their needs (Geer, 2003). Numerous countries have seen the solution to their problems in open source software.

A few examples: In Venezuela an independent operating system was developed. This system, based on the Debian Linux distribution, is called Canaima. National decree no. 3390 (http://tinyurl.com/3pdksvv) prescribes the use of Canaima in public administration and further states that all software specifically developed for public administration would have to be licensed under GPL (Cleto, 2004). For Hugo Chavez one of the reasons to transfer to open source software (in addition to security and the desire to be independent from the USA and Microsoft) was the information that 75 percent of licensed software is distributed to other countries, 20 percent for the support of foreign agencies, while only five percent is left for the Venezuelan programmers (Proffitt, 2002). Russia as well has decided to transfer to open source software. However, its transfer is still in progress and will be completed in 2012 or at latest by 2015. According to Putin, one of the main reasons for the transfer was the desire to be less dependant on other countries in the use of proprietary software (Morozov, 2011).

By 1990, China had also demonstrated interest in open source software. In 2005 it produced the first version of its national operating system of Linux distribution, Red Flag Linux that is used in the public administration. At the same time China has developed an Asianux distribution that was oriented towards Asian markets, for it supports Chinese characters (Blanchard, 2007). China, whose economy is growing extensively and with which also grows the requirements for a more localized software that satisfies the needs of local companies, is becoming more competitive in world markets through the development of its own software (Saxenian, 2003). Once, China had one of the highest levels of piracy in the world. This number has started to reduce due to the use of open source software. In this way China can provide greater information security and independence (Lock, 2006).

One of the largest supporters of open source software is the European Union. Some of the largest open source projects and solutions have originated within this organization (Gonzalez-Barahona, 2006). There has been an Open Source Observatory and Repository for European public administrations (OSOR) established in the European Union. The purpose of this institution is to develop special applications and the open source software to be used in the public administration within the EU. By means of this project the EU desires to reduce the costs in the public administration and standardize the formats and procedures across the union, reduce the expenses of e-government and help spread good practices (http://www.osor.eu/about).

The situation on the use of open source software in the public administration in Slovenia will be presented in a brief overview: In 2003, Slovenia adopted a document, The Policy of the Government of the Republic of Slovenia in the development, deployment and application of software and the solutions based on open source (Politika Vlade RS pri razvijanju, uvajanju in uporabi programske opreme in rešitev, temelječih na odprti kodi). Among other things the document states that the country will support the use of open source solutions and treat them equally together with the licensed solutions and support education on how to use them (*http://tinyurl.com/6gyjnou).* At the moment the document has not yet been put into practice. So far, based on a research Assessment of economic justification of the MS EA for the period between 2003 and 2005 (Ocena ekonomske upravičenosti MS EA za obdobje 2003-2005) (*http://tinyurl.com/68u2cm7)* in which it was established that the use of license software is more wise from a financial aspect than the use of the open source software, the state has been purchasing MS Office licensed equipment for public administration through public tenders. However, there are some bright exceptions in the public administration, such as the Supreme Court of the RS that completed the transfer between the years 2006 and 2007, thus changing the MS Office package with the Open Office package, the Microsoft Internet Explorer browser with the Mozilla Firefox and installing an open source application for e-mails called Thunderbird on 4600 work posts. The Supreme Court has established that this decision helps them save approximately 400,000 € per year (http://tinyurl.com/6hszcy5).

The year 2011 was a sort of a turning point in this domain for Slovenia, for the country published a study about its intention to gradually transfer to the use of open source software by 2015; at first merely by replacing the MS Office systems with the Open Office systems and in time by replacing all operating systems with the open source systems such as Linux distributions (*http://tinyurl.com/3f69g8u*). The mentioned study has initiated a number of critiques, mainly from the providers of licensed programme solutions. The Microsoft company has declared that with such a decision the Government would cause them to loose at least 2.5 million Euros per year (Mihajlovič, 2011).

In terms of open source software application Slovenia is behind in comparison to other EU member states. At this point we should nonetheless draw attention to a possible problem in the transfer of Slovenia's public administration to open source software. The problem lies in the applications that were made especially for the public administration – they are made solely for the Microsoft environment. Other countries have encountered similar problems; they all had to produce new programmes and applications or modify those already made that were adapted to the Microsoft environment, so they would also support other operating systems and be compatible with different formats, which only contributed to the increase of expenses (Souza, 2006).

There are some examples of transfers from the open source software back to the proprietary software. One such case occurred in Vienna when the users transferred back to Windows Vista and decided to develop their own distribution based on Debian Linux, called Wienux in 2005. In this case the major problem was a programme for computing education for children developed in 2003. It was developed only for the Internet Explorer environment and did not support the open source Firefox environment. The company that has developed the programme has foreseen the support for Firefox as late as in 2009. Therefore, in 2008, Vienna decided to go back to using Windows (Mobility, 2008).

The last such transfer was made by the German Ministry of Foreign Affairs that transferred to open source software in 2005. At the ministry they installed the Debian Linux distribution on their computers. The purpose of the mentioned transfer was to save the money that they would normally have spent on their licensed programmes. In their 2007 report they wrote that the transfer to open source software indeed helped them reduce their expenses. In 2011, they publicly announced that they are transferring back to MS Windows and MS Office. As a reason they stated that the programme did not support all hardware such as printers etc. In their opinion, expenses have not decreased, because they had to invest a lot of money in the development of their own printer drivers. The users have also complained regarding the lack of functions and poor interoperability. In their opinion, to transfer back to MS Windows will cost less, because they will not have to pay programmers to develop new drivers (http://tinyurl.com/5s4k3ry).

4 INFORMATION SECURITY AND THE ROLE OF OPEN SOURCE SOFTWARE

News about cybernetic attacks is becoming very frequent; most prominent are attacks between the USA and China. Nevertheless, cybernetic attacks are taking place also between numerous other states, since this asymmetric form of fighting allows them to reach targets with little effort and, what is even more important, without the use of force. However, these types of attack can have even greater consequences (Stuxnet example). Based on the McAfee report, over 120 countries have developed or possess "cybernetic weapons" for attacks on financial markets, state computers, military bases etc. These attacks are of various forms; from the DDOS attacks and hacking into systems to the theft of information. Due to an increased number of such attacks numerous countries have established special centres (in addition to CERT) to provide better communication between the affected individuals of an attack or to provide better response to such a situation. There is no uniform approach to the solving of this issue – each country has an approach of its own. The opinions on who should have the information about the attack and to what extent when such an attack occurs are also very diverse. Some believe that it is best to exclude the public, while others recommend as much transparency as possible (Baker, 2009).

Two approaches are predominant in the field of information security. The "top-down" approach derives from the concept of national security and is based on realistic security theory; it puts the state and its role in the writing and adopting of legislation, guidelines and strategies in the field of information security in the forefront. The security of individuals in the field of information and communication technology (ICT) must be taken care of by the state. However the state can also be directly or indirectly threatened by individuals (Svete, 2005). The weakness of such a system is that the legislation is falling behind the practice and that it is difficult for countries to protect individuals in the filed of ICT security in practice.

The "bottom-up" approach derives from the concept of human security, liberalistic and constructivist theory. This approach focuses on the individual, his values and interests. An individual in this field is not merely a victim that must be protected by the state, but an extremely important factor within the framework of information security who can have a strong influence on it through his work. An individual unskilled in the ICT domain can considerably threaten security. On the other hand, a very skilful individual can greatly contribute to security (Svete, 2005).

In the field of information security open source software can present a "bottom-up" approach, for it gives every individual an opportunity to control his own system. Even though its security is each individual's responsibility, the role of the open source community, which publicly warns about the threats, is likewise very important.

According to Diver (2007) an ideal system would be a combination of the "top-down" and "bottom-down" approaches. The state requires strategic guidelines and

legislation in the field of information security. Nonetheless, it would also be beneficial if they had individuals that would be more skilled in information science, who would take care of the system's security and thus prevent the spread of malware.

Conclusion

Throughout the years open source software has become serious competition for proprietary systems that hold a primary position both among supercomputers and servers. This increase has been noticed also by states, which have decided to transfer to open source software during the economic crisis with the desire to reduce costs, while other states decided for the transfer merely for security reasons and due to a desire for greater independence.

Security is relative. Each system can be hacked; therefore we cannot claim that one system is better than the other. Open source software has good security mechanisms, but in practice its safety depends greatly on its users. Open source software provides greater transparency and the possibility of insight into the source code, which interests only a small number of people. The majority of people use their computer for simple matters, such as writing documents. As far as security is concerned, people trust the developers of open source software and the many eyes system. But, in practice, it is evident that both systems are too much trust-based. There are a lot of cases where vulnerabilities in the systems have not been discovered for several years. Proprietary systems, on the other hand, do not allow an ordinary user to access the source code and are lacking the many eyes system. As with the open source software, here as well, security is based on the trust put into the developers. The only difference between open source software and proprietary systems is transparency. However, both systems are, unfortunately, based on the trust of people. Numerous countries have decided for open source software, because it allows them to inspect the source code themselves and, in addition, they can develop custom-made open source software. The main advantage of open source software in reference to proprietary systems is the public information about its vulnerabilities and their rapid elimination.

Open source software has good security mechanisms and provides greater transparency. Regardless, the endangerment of the system remains in the hands of the users. Most users are not skilled in using computers. These users endanger the security of their own and foreign systems, because they do not use antivirus programmes and do not update their computers regularly.

Hypothetically, in information security, open source software could be useful in the "bottom-up" approach only if all users were skilled in using computers, able to read the source code and detect vulnerabilities they could later publish publically. Since such expectations are utopian it is difficult to claim that open source software is better in terms of security because, at the end of the day, it is the user who presents a threat to the system.

Due to the current economic crisis in the world many countries are deciding to transfer to open source software to reduce costs. Slovenia is one of them. At this point it is necessary to draw attention to the possible challenges that might appear in case of a hasty and thoughtless transfer to open source software. We can learn a great deal from the German and Viennese examples (transfer back to proprietary systems because open source software did not support all hardware equipment). The compatibility of open source software and the current (or planned) hardware would have to be ensured. If established that such equipment is not supported, the costs pertaining to the development of appropriate drivers would have to be examined. The largest challenge probably lies in the software that is written only for the Windows environment. Therefore the costs for the development of corrections that would provide open source software support and the time for their production would also have to be examined. It would be reasonable to examine the current distributions and find the most appropriate distribution for the Slovenian environment (development of a local distribution is also possible).

If Slovenia does not start a possible transfer to open source software wisely it could happen that such a transfer will not be beneficial. In Slovenia, literature about this topic is scarce and so are articles and studies. In the future, it would be useful to carry out more independent cost-benefit analyses that would establish the advisability of Slovenia's transfer to open source software.

Bibliography

1. *Baker, S., 2009. In the Crossfire: Critical Infrastructure in the Age of Cyber War. Santa Clara: McAfee, Inc. http://tinyurl.com/3l6k4bm (3 August 2011).*
2. *Blanchard, J. F., 2007. China, multinational corporations, and globalization: Beijing and Microsoft battle over the opening of China's gates. Seoul: Asian perspective. Institute for Far Eastern Studies. http://tinyurl.com/3vb97m4 (2 April 2011).*
3. *Campbell, D., 1999. NSA Backdoor Into Windows. http://tinyurl.com/3k3e4d (12 April 2011).*
4. *Chuvakin, A., 2003. An Overview of Unix Rootkit. Chantilly: iDEFENSE Inc. http://tinyurl.com/42gbmjd (10 September 2011).*
5. *Cleto, S., 2004. Venezuela Embraces Linux and Open Source Software, but Faces Challenges. http://tinyurl.com/3dqh9ns (3 May 2011).*
6. *Diver, S., 2006. Information Security Policy - A Development Guide for Large and Small Companies. Washington: SANS Institute. http://tinyurl.com/3suc5tc (17 September 2011).*
7. *Espiner, T., 2006. Trend Micro: Open source is more secure. http://tinyurl.com/3c6u6cz (20 May 2011).*
8. *Espiner, T., 2010. Microsoft opens source code to Russian secret service. http://tinyurl.com/2w8moaq (3 August 2011).*
9. *Geer, D., in drugi, 2003. CyberInsecurity: The Cost of Monopoly How the Dominance of Microsoft's Products Poses a Risk to Security. http://tinyurl.com/63bhse8 (18 September 2011).*
10. *Germain, J. M., 2008. Linux: A Tempting Target for Malware?. http://tinyurl.com/65jn5vu (1 May 2011).*
11. *Gonzalez-Barahona, J., Robles, G., 2006. Libre Software in Europe. V C. DiBona, ur. Open Sources 2.0: The Continuing Evolution. O Reilly Media, p. 161–188.*
12. *Humbery, B., 2011. The Evolution of the Personal Package Archive system. http://tinyurl.com/3tqw8rq (20 May 2011).*

13. *Jose, M., 2011. The Goal is 200 Million Ubuntu Users in 4 Years - Mark Shuttleworth at UDS. http://tinyurl.com/3cd4p67 (23 May 2011).*
14. *Kalkuhl, M., 2009. Malware beyond Vista and XP. http://tinyurl.com/3qemfbs (20 May 2011).*
15. *Kimberly, S., 2005. The value of open standards and open-source software in government environments. Austin: IBM SYSTEMS JOURNAL. Volume 44 Issue 2, January 2005. http://tinyurl.com/3qsatqt (12 May 2011).*
16. *Koetzle, L., 2004. Is Linux More Secure Than Windows? http://tinyurl.com/3p9uue8 (20 May 2011).*
17. *Kovačič, M., 2006. Kriptografija, anonimizacija in odprta koda kot boji za svobodo na internetu. Javnost- the public. Vol. 13. (2006). Fakulteta za družbene vede, Univerza v Ljubljani, p. 93–110.*
18. *Laurie, B., 2006. Open Sources and Security. V C. DiBona, ur. Open Sources 2.0: The Continuing Evolution. O Reilly Media, p. 57–71.*
19. *Lewis, J., 2006. Government Open Source Policies – August 2007. Washington: Center for Strategic and International Studies. http://tinyurl.com/3mhva3y (12 May 2011).*
20. *Lock, B. Y., Liu L., Saxena S., 2006. When China Dances with OSS. V C. DiBona, ed. Open Sources 2.0: The Continuing Evolution. O Reilly Media, p. 197–210.*
21. *Meintjes, T., 2011. Is a virus or malware infection the cause of your slow computer? http://tinyurl.com/442u5se (26 May 2011).*
22. *Mihajlovič, N., 2011. Microsoft gre nad Pahorja, zdaj hoče pošteno konkurenco. http://tinyurl.com/3lxz4va (24 May 2011).*
23. *Mobility, T., 2008. Vienna failed to migrate to GNU/Linux: why?. http://tinyurl.com/6mktzl (9 September 2011).*
24. *Morozov, E., 2011. A Walled Wide Web for Nervous Autocrats. http://tinyurl.com/2vflb3c (29 May 2011).*
25. *O'Dowd, D., 2004. Linux Security Controversy. http://www.ghs.com/linux/security.html (18 September 2011).*
26. *Peeling, N., Satchell, J., 2001. Analysis of the Impact of Open Source Software. http://tinyurl.com/6lyeod8 (19 May 2011).*
27. *Poulsen, K., 2001. Borland Interbase backdoor exposed. Open source reveals foolishly hardcoded password. (12 May 2011).*
28. *Proffitt, B., 2002. Venezuela's Government Shifts to Open Source Software. http://tinyurl.com/3j9kqzo (15 May 2011).*
29. *Sapronov, K., 2007. Kaspersky Security Bulletin 2006: Malware for Unix-type systems. http://tinyurl.com/3vuu2k3 (23 May 2011).*
30. *Saxenian, A., 2003. Government and Guanxi: The Chinese Software Industry in Transiton. Berkeley: University of California at Berkeley. http://tinyurl.com/3u37nwl (12 May 2011).*
31. *Schneier, B., 2002. Secrecy, Security, and Obscurity. Crypto-Gram. http://tinyurl.com/5rk6jaw (17 May 2011).*
32. *Schneier, B., 2006. Microsoft Vista's Endless Security Warnings. http://tinyurl.com/ges4k (23 May 2011).*
33. *Souza, B., 2006. How Much Freedom Do You Want. V C. DiBona, ur. Open Sources 2.0: The Continuing Evolution. O Reilly Media, p. 211–229.*
34. *Svete, U. 2005. Varnost v informacijski družbi. Ljubljana: Fakulteta za družbene vede.*
35. *Weber, S., 2004. The success of open source. Cambridge: Harward University Press.*
36. *Wynants, M., 2005. Free as in Freedom, not Gratis!. V Wynants, M., Cornelis J., ed. How Open is the Future? Economic, Social & Cultural Scenarios inspired by Free & Open-Source Software. Brussels: Brussels University Press, p. 69–85.*

Sources

1. *http://distrowatch.com/ (26 May 2011).*
2. *http://librenix.com/?inode=21 (23 April 2011).*
3. *http://support.apple.com/kb/ht1528 (26 May 2011).*
4. *http://tinyurl.com/27l5wvh (26 May 2011).*
5. *http://tinyurl.com/28lpgq (28 May 2011).*
6. *http://tinyurl.com/3f69g8u (19 May 2011).*
7. *http://tinyurl.com/3fldnhr (26 May 2011)*
8. *http://tinyurl.com/3hdqvgd (20 May 2011).*
9. *http://tinyurl.com/3pdksvv (20 May 2011).*
10. *http://tinyurl.com/44x9pxd (26 May 2011).*
11. *http://tinyurl.com/5s4k3ry (10 September 2011).*
12. *http://tinyurl.com/68u2cm7 (12 April 2011).*
13. *http://tinyurl.com/6gyjnou (20 May 2011).*
14. *http://tinyurl.com/6hszcy5 (20 May 2011).*
15. *http://tinyurl.com/6jgg3ut (20 April 2011).*
16. *http://tinyurl.com/hdpo9 (18 September 2011).*
17. *http://tinyurl.com/o4foa (3 May 2011).*
18. *http://www.Linuxfordevices.com/ (20 April 2011).*
19. *http://www.osor.eu/about (20 May 2011).*
20. *https://bugs.launchpad.net/ubuntu/ (18 September 2011).*