End-user license agreement - threat to information security: a real life experiment

Žygimantas Kaupas
Faculty of Informatics
Kaunas University of Technology
Kaunas, Lithuania
e-mail: zygimantas.kaupas@ktu.edu

Jonas Čeponis
Faculty of Informatics
Kaunas University of Technology
Kaunas, Lithuania
e-mail: jonas.ceponis@ktu.lt

***Abstract*—This paper analyses end-user license agreement (EULA) and its impact on security of information and information technologies. Popular opinion suggests that people tend to accept EULA legal statements without good understanding of potential impact on their confidential data. To have a clear picture about current situation, real life experiment with specifically created license text was conducted. The results reveal serious information security flaws.**

Keywords—end-user license agreement; EULA; acceptance without reading

I. INTRODUCTION

As more and more data is stored online and the number of internet users is constantly increasing, creators of malicious software are persistently looking for some innovative ways to acquire valuable confidential information.

When recent malware, spyware, ransomware and other digital attacks were disclosed publicly and attracted a lot of attention [1], common trust in online information decreased notably. It is a commendable general practice to use an antivirus solution, do not open suspicious links or give your confidential data to an untrusted source. However, one attack vector is often forgotten.

Digital world is no longer imaginable without countless number of various software. Almost all of it asks the user to accept the end-user license agreement (EULA) before the start of an installation process. Following part is frequently overlooked by most of the users, even though real security threats might be hidden there.

This work analyses the concept of EULA and its drawbacks. Users trust in the information found online is tested with a software, which is made for this experiment and has a specifically designed EULA text. Obtained results enable identification of the problem scope and propose actions, which could help in closing this security gap.

II. END-USER LICENSE AGREEMENT ANALYSIS

End-user license agreement is a legal contract between a software application author or publisher and the user of that application [2]. This document should be used for protecting software creators from copyright infringements and liabilities when something goes wrong because of the mistakes in their products. However, it can become a tool against the final user too [3].

Multiple parameters have to be defined in order to evaluate if the user understood the license agreement before he consented to it. The most important objective variable R indicates the total amount of document readings. An additional subjective variable – understanding (U) – can be applied, however, it is too ambiguous to use without extensive questionnaires after the experiment. It is expected that for this EULA research the ratio between R and D (total amount of program downloads) will be at least 1/2. This would indicate that more than half of participants read the license agreement text.

A. *Ways of accepting the EULA and its drawbacks*

There are number of methods how a user might accept the EULA (sometimes even without knowledge of doing so) [2]:

- by clicking on “I agree” button during the software installation;
- by opening the shrink wrap on the package;
- by breaking the seal on the case;
- by sending a special card back to the software publisher;
- by executing a downloaded file (applicable more to UNIX systems);
- by using the software.

Users trust in the information found online will be tested with the first of the above-mentioned methods, since it is the most common one used in practice nowadays.

From the acceptance methods list it is already obvious that notifying the user about EULA terms is the least important objective for the software developers. Even more, this drawback is only the first one of many criticism objects related to this document.

One of the most criticized aspects of EULA is its length [4]. On average, it reaches 3000 words (11 pages with double spacing), but on some cases this number is more than 10 times bigger (in 2012 PayPal EULA contained 36 275 words [5]).

Unfortunately, there is no data available how many (if any) users read these documents at all.

In addition, difficult legal terminology is always used in EULA language. This significantly decreases documents' readability and contradicts the main idea, that all people should be able to read and understand it. Also, terms that may be harmful to user system or information confidentiality can be well hidden among those legal phrases.

B. Common Harmful EULA Terms

Even well-known companies use EULA for specific purposes. User monitoring is very often mentioned in this document. For example, in order to have a fully functional user assistant Cortana in Microsoft Windows 10 operating system, agreement on user data (installed programs, browsing history, etc.) collection by default is included into EULA. These settings can be disabled later, but that would cost some time, knowledge and effort for the end user [6].

Facebook on the other hand claims that it can use any digital content posted by its users for any companies' objectives as long as this media is not deleted from the website. Users' photos or videos could be included in an advertising material without any official notifications [7].

The restriction to criticize the software or compare it with similar products can be also found in EULA text. Even though in 2003 global computer security company McAfee was penalized for forbidding benchmark publications in such way, today well-known software products like Microsoft SQL Server or VMware Workstation still use similar restrictions in their EULA [8]. It is obvious that some terms are so desirable, that even financial punishment does not frighten software creators.

Finally, some IT giants like Microsoft or Google granted themselves the right to change users operating system state (uninstall programs, change settings, etc.) based on EULA. Officially this could be easily explained as a basic user protection; however, it does not exclude a possibility to delete some unwanted software or change required settings without any warning or justifying cause. Furthermore, Google allows itself to change EULA without a warning at any time. The underlying presumption is that user will check the latest version of this document from time to time [9]. Even though authors do not think that these well-known companies would risk their good name to exploit terms mentioned above, but there are number of those, who certainly would.

C. Legal EULA Analysis

There are many discussions online where EULA's legal obligations are debated. Usually people tend to think that this document is like an informational message or standard instruction, despite its usual start with the words "important legal agreement". Situation is even more complicated in Lithuania, since there are no judicial practices related to this question and even the EULA document itself most of the time is written in English language.

The Republic of Lithuania Law on Electronic Communications states that it is forbidden to gather any digital confidential information except when the user is informed and gave his agreement [10]. Similar principles are echoed in other legal documents about access to personal data. EULA perfectly fits the aforementioned principle – inform and receive a consent.

Situation in European Union is very similar to Lithuania's – there is still a shortage of court decisions related to the discussed document. According to E-commerce directive [11], each member state could exclude electronic agreement from binding documents list. However, as of 2011, none has selected this option and no information is present that it is chosen by anyone today [12].

Finally, even birthplace of EULA – USA – has no common verdict regarding legal obligations of this document. Related judgements are always made *ad hoc*. However, statistics are in favor of EULA and some widely-publicized trials ended in supporting this document and thus strengthened its legal power even more [13].

D. EULA's research and known solutions.

There is not a lot of academic attention to this document neither in Lithuania nor in the world. No published research could be found in Lithuanian language where EULA is the main analysis object. This document is seldom mentioned only in the context of intellectual property protection, but nowhere the potential threat of the software license agreement to confidential information or IT infrastructure is discussed.

Somewhat more research was done regarding the user familiarity with EULA text (before accepting it) worldwide. One of the most famous and extensive experiments was made in 2010 by Rainer Böhme and Stefan Köpsell [14]. They evaluated 80 000 respondents and concluded that less than 8% of them spent enough time to read the presented EULA text before clicking the accept button.

Other experiments gave similar results. In 2005 antivirus company PC Pitstop included information about the 1000$ prize in their EULA text. It was granted to the first responder who will write them a letter about it. The winner showed up only after 4 months and 3000 downloads [15]. Similar results occurred when cyber security solutions company F-Secure decided to do a Wi-Fi experiment and gave free public access to a specific hotspot only if the user agreed to give away his firstborn child [16]. In only 30 minutes 33 connections were made and there were no complaints about that tricky clause whatsoever.

On the other hand, there are just a few solutions to evaluate and automatically guard yourself against potential threats written in EULA. In the middle of 2012 the project called "Terms of Service; Didn't Read" started with a lot of public attention [17]. It rated and labeled websites terms & privacy policies into five groups and specified pros and cons from their agreements. Sadly, the last entry is dated July 2014 and it appears as the project is no longer active. Similar situation is with an application that automatically analyses EULA – "EULAlyzer" [18]. Though this program is still the best solution at the moment, it is also no longer developed and left with very limited functionality.

III. Experiment of Users Trust in the EULA

A. Research environment and collected data

This experiment was performed at the end of 2016. 653 first year students of Informatics faculty of Kaunas University of Technology were selected for this investigation. The defined scope helped in achieving several goals:

- to have a limited and known respondents number;
- to make sure that users do have greater than minimum computer literacy skills;
- to analyze the behavior of users which have a motivation to participate.

Experiment was carried out in the form of knowledge testing application for a specific university course. Students received a link to the downloadable application Quizza via an email from the course lecturer. It was specifically stated that this program is a personal project with potential programming errors. If the student answered more than 50% of test questions correctly, he received a link to the bonus material. No other information about the experiment was given in the email text. Even though the email sender in this case was not fake (in real-life phishing scenarios attacker tries to mimic the valid source), publication method and the fact that Quizza program was presented only once (no references were made during live lectures) should have raised at least some mistrust.

When user wanted to install the testing application on either Windows operating system machine or Android mobile device, it prompted the EULA to be accepted otherwise installation will be canceled. Every step of this experiment was made to replicate real world scenario as close as possible.

If users accepted the specifically modified EULA document, the installed software not only performed expected and visible functions, but also collected and sent some data from the machine it was running in. Actions with personal data are very restrictive and in most cases need various user approvals even for research purposes, therefore only a limited set of parameters for data collecting was chosen, which demonstrated access possibilities and security risks, but did not allow the exact person identification. This set included number of attached memory devices (hard drive, USB, CD/DVD), letter assigned to each drive (in Windows operating system) and the amount of free/occupied space. For the software to access these parameters it needs to have high privileges in the system. In comparison, it would be impossible to get this information by using a malicious web application.

B. Design of special EULA

Specific EULA text was developed for this experiment. Antivirus software Kaspersky license agreement was selected as a base model [19]. One difference from the standard EULA sample was that this time the document was written in Lithuanian language. Such modification made the EULA compliant to the country's law. It also helped evaluating whether the language does any difference to the readability of EULA and if that raises some questions for end users, why an unknown simple application would bother to use native language in its license agreement.

Other details were selected according to the standard license agreement: length of 3000 words, difficult legal language, liability limitations of software developer, etc. Several specific statements were created to trigger reader's attention and placed in the middle of EULA document text.

The first statement was labeled "Technical assistance" and had an active link to the application's support page. When visiting it, user could get an access to the desired bonus content without installing malicious application. Users who entered this page during the experiment and downloaded resources from it, were categorized as those who have read the EULA.

The next specific statement was a mixture of indications that this document is not a standard sample. One piece stated that "user data will be sent to the developer to have a better application security" (without any detailed explanation why or what exactly will be shared). Another part was a reference to the Republic of Lithuania Law On Legal Protection Of Personal Data [20] and data collection for scientific reasons. Finally, the last statement advised to cancel the installation and visit technical assistance page if the user does not agree with the license text.

C. Applications for Windows and Android operating systems

Two environment options were presented for the users in the experiment: Windows .msi or Android .apk installer files of a Quizza application. Both operating systems are the most popular in their domain with highest usage count [21].

In the Windows environment EULA usually has an additional dialog window where "Next" or "I agree" button has to be pressed in order to proceed. One common safeguard was added in our experiment to stop the user from automatically pressing the same button (usually "Next") throughout all installation process: additional agreement checkbox had to be selected before continuing to the next step.

The unsophisticated testing environment would be loaded afterwards, where users have to answer five out of ten questions correctly in order to get the desired extra content. Experimental application for devices running Windows was developed using Java programming language. It is a very lightweight solution where minimal code complexity is added only because of GUI (JavaFX package was used for its development). During the test user received a random question from a .txt file where the list of 30+ of them is present. Final score was counted after 10 questions. If minimal amount of 5 points is not reached, user can retry the attempt with another random set of questions.

In parallel to this activity, Quizza application used standard Java libraries to collect information about memory devices and third party email client Gmail to send data to the mailbox prepared for this experiment. None of the existing user's accounts were used for this process – the mail address of the sender was also created for this project and hardcoded into the application.

From the architectural point of view, two classes in application were separate and not connected to the quiz type functionality. *SpaceIO* class had 4 variables (*driveLetter*, *driveType*, *driveTotalSpace* and *driveFreeSpace*) and *calculateSpace()* method. If the method succeeded without any exceptions, all these 4 parameters were passed to *Email* class and *sendEmail()* method was invoked.

This class had several variables already hardcoded, like username, password, recipient, port, host, etc. Such solution enables keeping all code execution within an application. No calls to other programs or services are required. From this short description is obvious that experimental application is very simple and could be created by anyone having even limited programming skills. Still even this is enough to gather important data or invoke malicious code inside another user system.

Android application did not have any major differences neither with respect to functionality, nor related to hidden processes. Its Application Programming Interface (API) enables accessing many system parameters, however to do so it asks the user to grand rights in a special "App permissions" dialog before installing the application. During the testing stage it was noticed that Android version is more stable and reliable because mobile devices usually do not have any antivirus or other security software, which could block the outbound traffic.

Compared to Windows version, Android Quizza application is even less complicated, because GUI and part of system resources could be manipulated directly. In the Android environment it is easy to track whether the user has already accepted the EULA for a specific program version even after it is reinstalled many times in the same system. This enables the reduction of the amount of data being sent to the "attacker" and removes all possibilities of information duplication.

On one hand, there are almost no obstacles for malicious processes to perform hidden actions once the program is installed in the Android device. On the other hand, special permission window is displayed to the user before successful application installation. If the user pays attention to this dialog and has an idea how the program should work, any unnecessary privileges included in the list would certainly cause suspicion. This might result in user terminating the process before the attacker gathers any valuable data from that device.

D. Distribution environment of created programs

For the successful experiment, one needs to have not only prepared applications, but also the way to share them without causing any doubt about their legitimacy. Having this in mind, a bogus website quizza.tk was created. Only free services were used for its creation: .tk domain name and free Lithuanian hosting provider. Similar approach would allow an attacker to make a number of identical copies/alternatives of the distribution environment without spending a cent. In addition, during the registration for these services no real personal information was entered and no trackable financial payments were made thus allowing the real owner to stay hidden.

Main quizza.tk page during the whole experiment displayed notification "Site under maintenance. We'll be back soon". This fraud was applied in order to save time needed for a detailed website creation herewith creating a false expectation that such page really exists. In addition, it removed the possibility of navigation inside the page, which was needed to monitor how many students visited one or another link (prevented browsing through all the resources at once).

Furthermore, information about applications and website was sent from the mailbox of course instructor to all students. In our case, the sender was not falsified, but nowadays it is quite straightforward to alter this data and present it as coming from non-related legit source. Multiple links (separate for Windows and Android applications) were included in the email message. In practice, such method (well know source and some references to additional material) is commonly used for fraud purposes.

All links had a server side PHP script, which monitored how many times each of these references were clicked by the user. Three counters were set-up for each application to have versatile results of the experiment: how many times it was downloaded, how many people read the EULA and visited the "technical assistance page", how many students agreed with the license, solved the test and downloaded bonus content afterwards.

IV. RESULTS OF THE EXPERIMENT

From the initial email with details about these programs until the disclosure of the experiment two weeks were given for students.

As it is observable from Table I, more than a half of downloads ended up with application being installed and test passed. However, this statistic does not mean that similar number of students read the EULA and reached extra content via different link. Alternative route has not been visited at all, so EULA has not been read even once. What is more, almost 80% of those who passed the test shared their system data unknowingly.

TABLE I. WINDOWS PROGRAM STATISTICS

Times downloaded	Test passed	Data about devices received	EULA read
245	130	103	0

Biggest interest in experimental application was during the next day after the announcement – data about 62 devices (60% from total amount) was received. As it was expected, not only hard drives, but also USB devices and CDs/DVDs were monitored. Even though during the testing stage, some antivirus solutions proved that they would stop "malicious" traffic from leaving user computer, other ones did the opposite. For example, specific Avast versions even inserted additional text to the email that was sent without user awareness – "--- This email has been checked for viruses by Avast antivirus software. https://www.avast.com/antivirus". Even after experiment disclosure was made, 13 students used the

application and thus shared their data to the author (1 from those even after 1.5 month from that date). It looks like people still trust the program despite knowing that it did things with their machine without their awareness.

Result of Android application experiment are presented in Table II. In general they are very similar to Windows version, however even less students who downloaded the application bothered to finish the test with required result (probably they wanted just to see the application's appearance, expected to get different practice questions or just installed it on multiple various devices). Surprisingly that even though there are usually no security solutions in the mobile environment, 10% less (70% on Android compared to 80% on Windows) data was successfully gathered from this malicious application. Overall, none of the students bothered to read the EULA and check the link included in its text.

TABLE II. ANDROID APPLICATION STATISTICS

Times downloaded	Test passed	Data about devices received	EULA read
155	73	50	0

V. CONCLUSIONS

In conclusion, the conducted experiment confirmed that users tend to skip the EULA and agree with any text written in it. The expected R/D ratio of 1/2 was not reached as nobody accessed the alternative link in license agreement text thus setting this ratio to the lowest minimum - 0.

Since this agreement is a legal document, all included terms must meet strict law regulations. However, even official applications could collect considerable amount of confidential data or track user behavior without breaking any laws.

In addition, this experiment showed more alarming IT security trends. First of all, if the attacker manages to trick the user with the initial source validity, other steps to the complete control over his system might be very easy. More than 60% of data received came within the first 24 hours from the start of the experiment. This tendency favors zero-day exploits or new fraud schemas and as it was visible no home antivirus solutions provide sufficient protection against data theft.

Furthermore, received data disclosed that home users do not benefit by virtualization technology to increase their systems security. During the experiment malicious application has monitored hard drives with plenty of storage accessible. Also, in many instances connected external USB flash drives were detected when user installed this untrusted application. That could be easily used for further spread of the malware. Finally, data from 17 new devices was received after the disclosure of this experiment. It shows that either information does not reach all parties even in a relatively small group or some people still use digital resources after their malicious behavior (potentially only one of many) is known.

There are lots of security solutions from the simplest free versions to expensive professional programs, yet it seems that lessons from 5 thousand years' legend about Trojan Horse are still not learned. Why bother breaking down multiple security layers if the user himself will take you inside?

REFERENCES

[1] M. Ward, "'Alarming' rise in ransomware tracked". Available: http://www.bbc.com/news/technology-36459022 [Accessed: 22 February 2017].

[2] M. Rouse, "End User License Agreement (EULA)". Available: http://searchcio.techtarget.com/definition/End-User-License-Agreement [Accessed: 21 February 2017].

[3] J. Newman, "Top EULA Gotchas: Website Fine-Print Hall of Shame". Available: http://www.pcworld.com/article/249396/top_eula_gotchas_website_fine_print_hall_of_shame.html [Accessed: 22 February 2017].

[4] R. W. Gomulkiewicz, "Getting Serious about User-Friendly Mass Market Licensing for Software" George Mason Law Review, vol. 12, pp. 687-718, 2014.

[5] S. Jary, "Apple iTunes T&Cs 10% longer than Shakespeare's Macbeth". Available: http://www.pcadvisor.co.uk/feature/apple/apple-itunes-tcs-10-longer-than-shakespeares-macbeth-3346281/ [Accessed: 22 February 2017].

[6] D. Goldman, "Is Windows 10 really a privacy nightmare?" Available: http://money.cnn.com/2015/08/17/technology/windows-10-privacy/ [Accessed: 22 February 2017].

[7] Facebook Statement of Rights and Responsibilities. Available: https://www.facebook.com/legal/terms [Accessed: 22 February 2017].

[8] A. Newitz, "Dangerous Terms: A User's Guide to EULAs". Available: https://www.eff.org/wp/dangerous-terms-users-guide-eulas [Accessed: 22 February 2017].

[9] Google Chrome Terms of Service. Available: https://www.google.lt/intl/eng/chrome/browser/privacy/eula_text.html [Accessed: 22 February 2017].

[10] The Republic of Lithuania Law on Electronic Communications. Available: https://www.e-tar.lt/portal/en/legalAct/TAR.82D8168D3049 [Accessed: 21 February 2017].

[11] Directive on electronic commerce. Available: http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:32000L0031 [Accessed: 21 February 2017].

[12] M. Webber, L. Rubin, "Liability matters under end user licence agreements". E-Commerce Law and Policy, vol. 13(4), 2011.

[13] N. Anderson, "No, you don't own it: Court upholds EULAs, threatens digital resale". Available: https://arstechnica.com/tech-policy/2010/09/the-end-of-used-major-ruling-upholds-tough-software-licenses/ [Accessed: 22 February 2017].

[14] R. Böhme, S. Köpsell, "Trained to accept?: a field experiment on consent dialogs" CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2403-2406, 2010.

[15] "It Pays To Read License Agreements (7 Years Later)". Available: http://techtalk.pcpitstop.com/2012/06/12/it-pays-to-read-license-agreements-7-years-later/ [Accessed: 21 February 2017].

[16] "Tainted Love: How Wi-fi betrays us". Available: https://fsecureconsumer.files.wordpress.com/2014/09/wi-fi-experiment_uk_2014.pdf [Accessed: 21 February 2017].

[17] "Terms of Service; Didn't Read". Available: https://tosdr.org/ [Accessed: 22 February 2017].

[18] "EULAnalyzer". Available: https://www.brightfort.com/eulalyzer.html [Accessed: 21 February 2017].

[19] Kaspersky EULA. Available: http://www.kaspersky24.lt/kis/Licence%20agreement%20LT.pdf [Accessed: 22 February 2017].

[20] Republic of Lithuania Law on Legal Protection of Personal Data. Available: https://www.e-tar.lt/portal/lt/legalAct/TAR.5368B592234C [Accessed: 22 February 2017].

[21] "Operating System Market Share Worldwide". Available: http://gs.statcounter.com/os-market-share [Accessed: 22 February 2017].