# An approach for software protection in cloud computing environment based on watermarking technique

Anna Storozhenko

Information Security Department,
Lviv Polytechnic National University, UKRAINE, Lviv,
S. Bandery street 12, E-mail: anetso18@gmail.com

***Abstract – A descriptive survey for software security threats in the cloud computing environment is presented in this paper. As a solution for that applying watermarking technology for a generic copyright protection is proposed.***

Key words – software protection, software watermarking, cloud computing, service provider.

## I. Introduction

Today many companies and organizations are moving their systems to the cloud. This is a platform-as-a-service hosting (PaaS) of cloud computing. It is an alternative approach to use of powerful computing resources without having local servers to handle their applications. Windows Azure is an example of Microsoft's cloud PaaS service. Itself it is an operating system that runs on server computers located in Microsoft data centres and enables running Windows applications and storing data in the cloud [1].

Cloud computing also provides software-as-a-service (SaaS) hosting, which becomes more and more popular among software users. That is because they don't need to have programs locally installed on their computers, so there is no reason to update them every time. Applications are running on the cloud servers and could be accessed via the Internet. This will save a memory and CPU power as well as money, because user pays for "service on demand".

Cloud technology allows you to expand the capabilities while saving resources. But at the same time because of communications over the Internet there are more security threats to user data, databases and software applications.

## II. Software Security and Cloud computing

On Cloud providers side there are standard protocols and security features for software protection. Mostly they are related to user's authentication, secure runtime environment, access control and data security. This article discusses security feature for software copyright protection in the cloud, that cloud be supplied by the provider itself as well as by user application.

Applications running on cloud platforms have a different trust relationship between the development environment and the deployment environment from traditional enterprise applications. In a traditional enterprise application, all of the environments are contained within the enterprise. This trust is created by isolating secure hosts and secure networks, which are part of the enterprise's computing infrastructure. Cloud computing platforms change the trust boundary relationships between the development environment and the application's runtime environment [2].

In this article, we form a security model of the copyright-infringement problem by considering the security concerns for organization, using the cloud provider to store and execute its software on the cloud platform. This is PaaS cloud model.

We also have to consider the case when the company wants to sell software or provide it as a service on cloud platform. Other's companies and users could purchase this software and access it throw Internet. That's why, SaaS cloud, in which the software is running on cloud-based servers is also briefly considered in this scenario.

Fig. 1 describes communication scheme between user terminal and cloud platform.

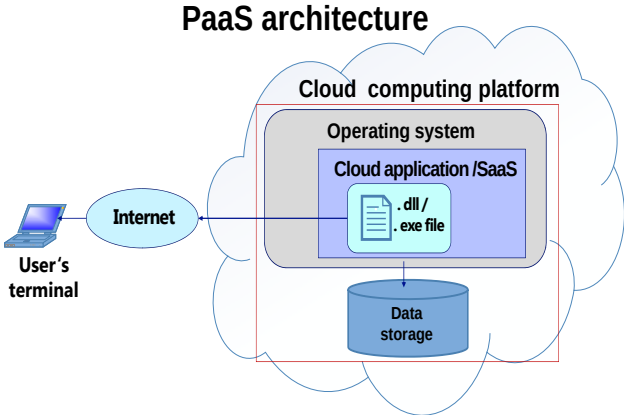


Fig. 1.Cloud computing schema

## III. Software Threats in Cloud Computing

It is assumed that three kinds of entities try to attack the cloud computing system. External attackers can eavesdrop or modify Internet communications between a user terminal and service program. Malicious users try to attack other users to steal secret information or using a service without the correct permission. Furthermore, we have to consider malicious platform providers. However, if the ability of the malicious platform provider is unlimited, we have to assume all possible attacks by the provider, which is a very difficult task to realize secure cloud computing. Thus, it was considered a reasonable adversary model.

The platform provider honestly executes user requests and cannot obtain any information from the execution environment such as physical memory. However, the platform provider may try to use the user's program maliciously or to obtain information from data storage. This model is a reasonable model where we consider the system manager of the platform as an attacker. We should consider the following threats for secure cloud computing.

1. Malicious users or malicious platform providers may access a service program and execute it on the platform.
2. Malicious users or malicious platform provider may steal user's information stored into the service program.
3. External attacker may modify a communication between a user terminal and the platform, or steal user's information from communication data [3].

4. External attacker, malicious users or malicious platform provider may simply download paid applications and reverse engineer them, make minor modifications and then claim copyright and release applications to make money.

## IV. Methods for Software Protection

There are some commonly used techniques of software protection, such as license checking by using an activation code, encryption and digital signature, software obfuscation and watermarking. These methods all have certain limitations in practical applications.

The serial numbers are only used to protect the program's entry point. In the case of evaluation software, a serial number is provided by the software vendor to activate the product. The software generally stores the activation date in registry and checks the evaluation period based on that. This protection is easily overcome by evaluators by clearing the registry entities or resetting the system clock. Moreover the same serial number is used on different machines to get access to multiple installations. In some scenarios, the evaluation period needs to be extended legitimately. For this purpose, the user needs to request a separate license. However, the vendor has no control on the license already issued, in the event of an evaluator violating the licensing terms. When users change roles or leave the organization, some user rights need to be revoked and new rights need to be granted. But, in most of the digital rights management solutions revocation of rights is cumbersome and requires a new license to enforce it. Password based protection techniques are not robust and often come up with shortcomings such as sharing of passwords or reuse of serial numbers [4].

The encrypted program must be decrypted before executing process. Code obfuscators can resist reverse engineering techniques to some extent but cannot offer a foolproof protection [5].

Online schemes use an external server to check the user license or to execute essential parts of a program. The drawbacks of these schemes are that the program manufacturer must deploy and manage a server.

There is the same security issue on mobile agent [6] how to protect execution code against a malicious execution environment. Sander and Tschudin proposed a solution [7] using a homomorphic encryption scheme. In their scheme, all operations are executed for encrypted data; thus, only limited operations can be implemented, and heavy computations are required for the operations [3].

Collberg and Thoborson proposed software watermark techniques to protect software modules [8]. In the past, watermarking was mainly used for digital copyright management. Software watermark is a unique identifier embedded in the protected software, which is hard to remove but easy to verify. However, most of commercial and open source software does not have software watermarks embedded.

## V.The Software Watermarking Model

Our interest is watermarking software, not media. But many of the principles are the same. Given a program **P**, a watermark **W**, and a key **K**, a software watermark embedder produces a new program **Pw**. We want Pw to be semantically equivalent to P (have the same input/output behavior), be only marginally larger and slower than P, and of course, contain the watermark W. The decoder takes Pw and the key Ke as input and returns W.

Let P denote the set of programs that are accepted by a watermarking system. W is the set of watermarks for this system and Ke - the set of embedding keys.

A watermark is a message of bits expressed by 0 and 1 with a finite length.

Embedding function is:

$$P \times Ke \times W = Pw \quad (1)$$

The watermark embedding function **E** is used to insert a watermark into a program. For the program **P** that belongs to the set of programs P, **Ke** that belongs to the set of keys Ke whereas the watermark **W** will be taken from the set of watermarks W is, so

$$\forall P \ni P,\ \forall Ke \ni Ke, \forall W \ni W:$$
$$Pw = E(P, Ke, W) \quad (2)$$

This combines to form a watermarked program. Watermark detection function is as follows:

$$Pw \times Kd \times W = W \quad (3)$$

The detection function retrieves the watermark from the watermarked program, this helps in verifying the ownership of the software.

$$\forall P \ni P,\ \forall Ke \ni Ke, \forall Kd \ni Kd, \forall W \ni W:$$
$$D(E(P, Ke, W), K) = W \quad (4)$$

## VI. Software Watermarking Schemes for PaaS and SaaS

As was assumed in section 3 of this paper, the cloud provider has great powers of observations as well as control to respect the services and infrastructure it provides to client. But also there are the threats of illegal copying and stealing software by both external and internal violators. So, additional requirements to platform are occurring in this context.

We studied the approach by which service provider could ensure more trusted software sharing, - the software watermarking. Software watermarking is a technology of copyright protection, which embeds the copyright information into digital production to avoid, being tampered, peculated, and illegally copied. There are many existing software watermarking schemes [9, 10], but main idea of technique is based on putting some unique byte sequence into the binary code. The byte sequence must be inserted carefully as not to make the binary unusable. It also depends on the type of binary - .NET, .jar or native .EXE etc. Also the byte sequence must be such that it is hard to identify for anyone not familiar with this specific watermarking system.

There is two possible ways to implement software protection in cloud environment using watermarking.

According to the first way, the mechanism for embedding watermarks may be included into the basic functionality of platform provider. In this way, cloud provider should offer an additional pair of services for embedding and detecting the watermark.

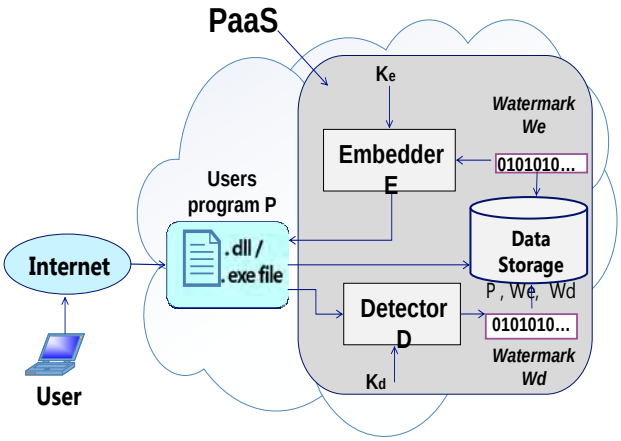


Fig. 2. Watermarking scheme in PaaS

Fig. 2 shows the watermarking scheme implemented to the architecture of the cloud platform.

The mechanism for embedding watermarks included into the basic functionality of platform provider. When users program P is uploading on the server, detector D checks it for watermarks. If is not found any tag, the embedder E will automatically label the program with watermark W. A special database can be created on server's data storage to store additional information about each program. As an example, it may be hash code of program P, which can be checked each time at every boot. If system not finds any watermarks in application, but the database contains a record for this program with information about watermark, which should be present inside, program P, in this case, will be considered as stolen. Every attempt to install it on the server will be rejected. Otherwise, if the system finds a watermark in the application, but it does not match the databases sample the program will be rejected too.

Also, as it was described in [11] the algorithm for embedding watermarks can be implemented by special web application, which provides a watermarking service in a SaaS layer.

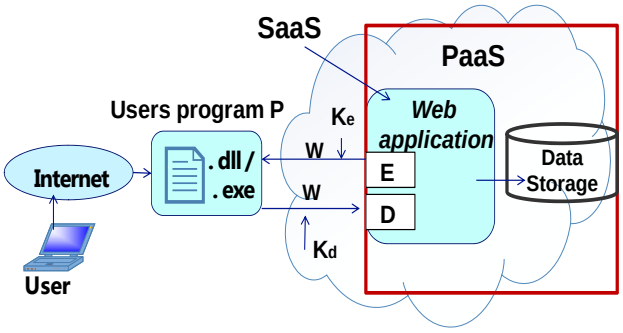


Fig. 3. Watermarking scheme in SaaS cloud

Fig. 3 shows the watermarking scheme implemented by the web application.

The application will use steganographic security labels W, for digital-rights management service, which is ensured by cloud platform. However it also may embed watermarks, upon request from the user, to mitigate the outside-users threat, which uses its software.

These watermarks must be robust enough to resist removal attempts. Furthermore, our application must be able to inspect all uploaded software P for watermarks.

In this approach, the watermarking key Ke must be known to user, but should remain a secret from the outside-users – otherwise the key provides no security advantage. The key Ke cannot be specific to the program being watermarked; otherwise user would not be able to retrieve a watermark from an arbitrary program of unknown or dubious provenance. Watermarking service must retain archival copies of unmarked programs in any event, for their investigations, and duplicated requests can be detected efficiently by comparing hashes of the application code.

We can also apply fingerprinting techniques. In this way different watermarks are embedding for different authorised recipient. Fingerprinting adds an identifying value in a program. As with the watermarking service, the fingerprint string should be encrypted to mitigate known-plaintext attacks on detection function.

## Conclusion

In this paper was presented the way in which software piracy in cloud computing environment could be reduced.

Main software threats were considered and existing protection schemes were reviewed in this context.

Also were proposed two security models for software copyright protection in the cloud, that cloud be supplied by the provider in PaaS layer, and by user application (SaaS), that are based on watermarking.

After a brief review of existing methods for software protection, you can say that each of them has its advantages and disadvantages. The proposed approach also has its drawbacks. Watermarking solutions can act as deterrents but cannot actively prevent misuse of software. In addition there is a risk removing the watermark from the program. Therefore, for the better protection it is possible to apply a combination of several of these methods.

## References

[1] Microsoft. "Windows Azure. Cloud services," itunes.apple.com [Online]. Available: http://www.windowsazure.com/en-us/services/cloud-services/. [Accessed: Oct. 01, 2013].

[2] Cloud Security Alliance. "Domain 10: Guidance for Application Security V2.1" 2010. [Online]. Available: https://cloudsecurityalliance.org/guidance/ csaguide-dom10-v2.10.pdf. [Accessed: Sep. 25, 2013].

[3] Kazuhide Fukushima, Shinsaku Kiyomoto, Yutaka Miyake, "Towards Secure Cloud Computing Architecture – A Solution Based on Software Protection Mechanism", Journal of Internet Services and Information Security, volume: 1, number: 1, pp. 4-17 [Online]. Available: http://isyou.info/jisis/vol1/no1/jisis-2011-vol1-no1-01.pdf. [Accessed: Oct. 01, 2013].

[4] Ravi Sankar Veerubhotla, Ashutosh Saxena. A DRM Framework Towards Preventing Digital Piracy. 7th International Conference on Information Assurance

and Security (IAS), 2011. [Online]. Available: http://www.academia.edu/4560551/A_DRM_Framework_Towards_Preventing_Digital_Piracy [Accessed: Oct. 01, 2013].

[5] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai,S. Vadhan, and K. Yang. "On the (Im)possibility of Obfuscating Programs," Advances in Cryptology (CRYPTO01), volume 2139 of Lecture Notes in Computer Science, pp.1–18. Springer-Verlag, August, 2001. [Online]. Available: http://www.boazbarak.org/Papers/obfuscate.pdf [Accessed: Sep. 25, 2013].

[6] A. Corradi, R. Montanari, and C. Stefanelli. Security issues in mobile agent technology. In Proc. of the 7th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS '99), Cape Town, South Africa, IEEE, pages 3–8, December 1999.

[7] Tomas Sander and Christian F. Tschudin. Protecting mobile agents against malicious hosts. In Proc. Of Mobile Agents and Security 1998, LNCS, Springer-Verlag, volume 1419, pages 44–60, 1998.

[8] Christian Collberg, Clark Thomborson. "Software watermarking: Models and dynamic embeddings." In: Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages. ACM, 1999. p. 311-324. [Online]. Available: http://users.rowan.edu/~tang/courses/ref/watermarking/collberg.pdf [Accessed: Sept. 05, 2013].

[9] Tharaud J, Wohlgemuth S, Echizen I, Sonehara N. Privacy by data provenance with digital watermarking - a proof-of-concept implementation for medical services with electronic health records. Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 510–513, 2010.

[10] Hwang K, Li D. Trusted cloud computing with secure resources and data coloring. IEEE Internet Computing Sept - Oct 2010; 14(5):14–22. [Online]. Available: http://gridsec.usc.edu/hwang/papers/ trusted-cloud-computing.pdf [Accessed: Oct. 01, 2013].

[11] Zhiwei Yu1, Chaokun Wang, Clark Thomborson, "A Novel Watermarking Method for Software Protection in the Cloud", Software Practice and Experience 2010; 00:1–23 [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/spe.1088/abstract [Accessed: Oct. 01, 2013].